UM SISTEMA DE RECONHECIMENTO DE VOZ CONTÍNUA DEPENDENTE DA TAREFA EM LÍNGUA PORTUGUESA

Sidney Cerqueira Bispo dos Santos e Abraham Alcaim

Resumo - Este artigo propõe um sistema de reconhecimento de voz contínua para ligações telefônicas automáticas, em língua portuguesa, que permite ao usuário uma forma mais natural de conversação, evitando que se limite a apenas repetir dígitos. O sistema utiliza as unidades fonéticas de um dos inventários reduzidos de unidades recentemente propostos para o português. Propõe-se uma configuração de máquina de estados para o processo de reconhecimento que é baseada na estrutura gramatical da língua portuguesa. A partir dessa configuração, propõe-se uma nova estrutura com a incorporação dos conhecimentos dependentes da tarefa. Em seguida, são analisados e comparados dois algoritmos, amplamente utilizados na fase de reconhecimento por vários sistemas: o "level-building" e o "one pass". Mostra-se, também, como incorporar nos algoritmos os conhecimentos traduzidos na máquina de estados.

Palavras-chave: Reconhecimento de Voz Contínua, Algoritmos de Reconhecimento de Voz.

Abstract - In this paper, we propose a continuous speech recognition system for automatic dial-up telephone calls in the Portuguese language. The system allows a more friendly operation, avoiding that the user be constrained to digit repetitions. The phonetic units are obtained from a reduced inventory recently proposed for the Portuguese language. A finite state machine configuration is proposed for the recognition process, which is based upon the structure of the Portuguese grammar. Using this configuration a new structure is proposed that incorporates task-dependent knowledge sources. Two algorithms widely used in the recognition step (the "level-building" and the "one-pass") are then analyzed and compared. Moreover, it is shown how to incorporate into the algorithms the knowledge from the state machine.

Keywords: Continuous Speech Recognition, Speech Recognition Algorithms.

1. INTRODUÇÃO

Atualmente, os sistemas de reconhecimento de voz estão saindo dos laboratórios para a utilização comercial plena, principalmente nas áreas de reserva de passagens, editores de voz (*speech-to-text translation*) e de telefonia. Entretanto, os maiores e mais poderosos sistemas ainda estão em fase de pesquisa. A quase totalidade desses reconhecedores (por exemplo, SPHINX, JANUS, ABBOT, LINCOLN, etc) foi desenvolvida baseada na língua inglesa, impulsionados principalmente pela ARPA - *Advanced Research Projects Agency* e em menor escala (p. ex: TANGORA, HTK, TI, etc) por grandes corporações das áreas de informática e telecomunicações. Essas pesquisas produziram bancos de dados que se tornaram padrões tais como o *997-Word RM* [1], o TIMIT [2], o WSJ [3] etc, o que impulsionou enormemente o desenvolvimento do reconhecimento de voz. Entretanto, esses bancos não podem ser utilizados nos sistemas em português.

Ao se desenvolver um reconhecedor de voz em um país que possua língua diferente da inglesa é importante observar a taxa de recobrimento fonético entre as duas [4]. No caso desse recobrimento ser alto, é suficiente um grande banco de dados para treinamento e modelagem de alguns aspectos específicos da linguagem. Sistemas de Reconhecimento de Palavras Isoladas (RPI) e Reconhecimento de Palavras Conectadas (RPC) podem ser adaptados dessa maneira. Contudo, sistemas de Reconhecimento de Voz Contínua (RVC) são muito mais complexos e exigentes e, mesmo dentro de um país onde haja unidade lingüística, diferenças regionais, sotaques e idiossincrasias podem acarretar perdas no desempenho.

Fazendo-se uma rápida comparação entre as línguas portuguesa e inglesa, verifica-se que existem muitos aspectos divergentes. Alguns destes aspectos são os seguintes:

- Vários fonemas da língua portuguesa não existem na língua inglesa e vice-versa;
- As duas divergem em diversos aspectos na construção e ordem de algumas classes gramaticais;
- Na língua inglesa existe uma nasalização intrínseca na prosódia e não existe distinção fonética entre vogais orais e nasais [5], como por exemplo, no português para se distinguir *massa* de *maçã*;
- A língua portuguesa possui, ainda, combinações fonéticas próprias, como por exemplo, as combinações *ão* e *ões*, entre outras.

Com base nessas observações, conclui-se que, em aplicações com um grau razoável de exigência, a criação de um sistema de reconhecimento de voz contínua em português é mais do que a simples adaptação de um sistema estrangeiro ou o treinamento desse sistema com bancos de dados apropriados. O conhecimento e a utilização de características próprias da língua portuguesa, sem dúvida alguma, tornará o processo de reconhecimento mais preciso e mais natural - requisitos essenciais de várias aplicações. Nesse aspecto, a

Sidney Cerqueira Bispo dos Santos é do Sistema de Proteção da Amazônia - SIPAM, Brasília - DF. Abraham Alcaim é do CETUC - PUC - Rio de Janeiro, RJ. E-mails: sidney@sipam.gov.br, alcaim@cetuc.puc-rio.br. Editor Ad Hoc responsável: Rui Seara. Artigo submetido em 11/Jun/2001, revisado em 09/Jan/2002, aceito em 23/Abr/2002.

pesquisa na área de voz no Brasil, embora com resultados de grande importância (vide, por exemplo, [7]-[32]), ainda certamente necessita de um maior desenvolvimento. Além disso, não existe banco de dados em português que seja considerado padrão para fins de comparação.

Neste artigo é proposto um sistema de reconhecimento de voz contínua para ligações telefônicas automáticas em língua portuguesa. O sistema utiliza as unidades do inventário 1 descritas em [6], que emprega 149 unidades fonéticas, os quais ficam numa posição intermediária entre fonemas independentes do contexto e fonemas dependentes do contexto. O objetivo é permitir ao usuário uma forma mais natural de comunicação. O enfoque é direcionado principalmente para os algoritmos de decodificação e para a utilização de gramáticas e de conhecimentos dependentes da tarefa. O sistema é descrito na Seção 2, o treinamento dos modelos na Seção 3 e os algoritmos de reconhecimento na Seção 4. A Seção 5 apresenta os resultados de simulação. Em seguida, na Seção 6, é considerada a incorporação de conhecimentos dependentes da tarefa. Finalmente, a Seção 7 resume as principais conclusões do trabalho.

2. DESCRIÇÃO DO SISTEMA

A Figura 1 apresenta o diagrama em blocos do modelo utilizado. O objetivo do sistema é permitir que o usuário, ao ouvir o tom de discar ou uma voz sintetizada lhe informando para pronunciar seu pedido, o faça de forma natural - sem se preocupar em falar os números na forma de dígitos ou de cardinais.

Até recentemente, o reconhecimento de dígitos isolados ou conectados eram consideradas as únicas abordagens viáveis. Entretanto, principalmente no Brasil, as cadeias de dígitos são memorizadas e pronunciadas como números *naturais* (zero vinte e um, zero oitocentos, quinhentos e doze etc) e não somente como uma seqüência de dígitos. Portanto, será muito benéfico o desenvolvimento de sistemas que reconheçam os números falados da maneira como eles são pronunciados. Por outro lado, é muito mais difícil reconhecer esses tipos de números, pois a quantidade de palavras e de segmentos acusticamente semelhantes aumenta consideravelmente (dezenove → dez e nove, dezoito → dez e oito, oitenta e cinco → oitenta cinco etc) trazendo grandes problemas de ambigüidade, além do que, desses números dependerão diretamente diversas ações do sistema. Isso torna a tarefa de reconhecimento dos algarismos ainda mais crítica devido à necessidade de precisão.

Esse ambiente foi escolhido neste trabalho por diversas razões. Primeiro, porque a tarefa de reconhecer números é uma das mais exigentes em termos de processamento. Segundo, porque é suficientemente complexo exigindo a utilização de conhecimentos sintáticos, pragmáticos e de algoritmos de decodificação (*parsing*) bastante complexos. Terceiro, porque não exige um número excessivo de palavras e finalmente, porque é um problema real e de interesse prático.

Devido ao fato de que os estudos a serem efetuados estão direcionados principalmente para o treinamento, para a aplicação de conhecimentos sintáticos e pragmáticos e desenvolvimento de algoritmos de decodificação, a análise de problemas de cápsulas telefônicas, de canal e de ruídos oriundos da linha telefônica ficarão como proposta para uma etapa posterior. Assim, foram utilizados como arquivos de voz, gravações efetuadas com microfones comuns, entretanto, em ambientes normais de conversação.

Após uma estatística entre aproximadamente 150 pessoas, quanto ao seu modo de pedir uma ligação telefônica, definiram-se as palavras do dicionário do sistema. Essas palavras estão apresentadas na Tabela 1.

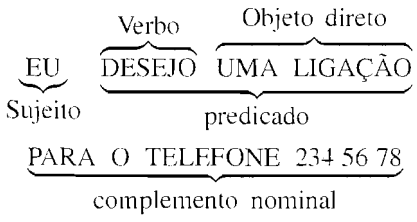
Durante o desenvolvimento do reconhecedor verificou-se a necessidade de se considerar as palavras "*a cobrar*" como uma só para que fossem capturados os fortes efeitos de coarticulação existentes entre elas. O mesmo fato aconteceu com "*gostaria de*".

As palavras da Tabela 1 podem ser consideradas terminais de uma linguagem formal. Quando concatenadas segundo regras de uma gramática regular, extraídas da língua portuguesa, formam as sentenças permitidas pela linguagem. De forma semelhante, as unidades constantes do inventário 1, descrito em [6], podem ser consideradas terminais e concatenadas para formar as palavras da Tabela 1. A linguagem formal resultante pode ser representada por uma máquina de estados finita que permite a geração de todas as sentenças possíveis. A idéia é a incorporação da sintaxe da língua portuguesa na máquina de estados de modo que a linguagem formal gerada produza combinações de terminais que sejam sentenças legais do português.

Levando em consideração a estrutura da língua portuguesa, podemos verificar que ao se utilizar as palavras do dicionário para pedidos de ligações telefônicas pode-se construir estruturas, entre outras, do tipo:

- <sentença> = <sujeito> + <predicado> + <complemento nominal>
- <predicado> = <verbo significativo> + <objeto direto>
- <complemento nominal> = <preposição> + <artigo definido> + <substantivo> + <numerais>
- <sujeito> = <pronome>
- <objeto direto> = <artigo indefinido> + <substantivo>.

Por exemplo, a seguinte sentença pode ser formada com as palavras do dicionário e essa estrutura:



O reconhecedor para ligações telefônicas automáticas exige um número limitado de estruturas desse tipo. A partir da análise das combinações das classes gramaticais da língua portuguesa que podem ser aplicáveis à tarefa em questão, pode-se construir a máquina de estados apresentada

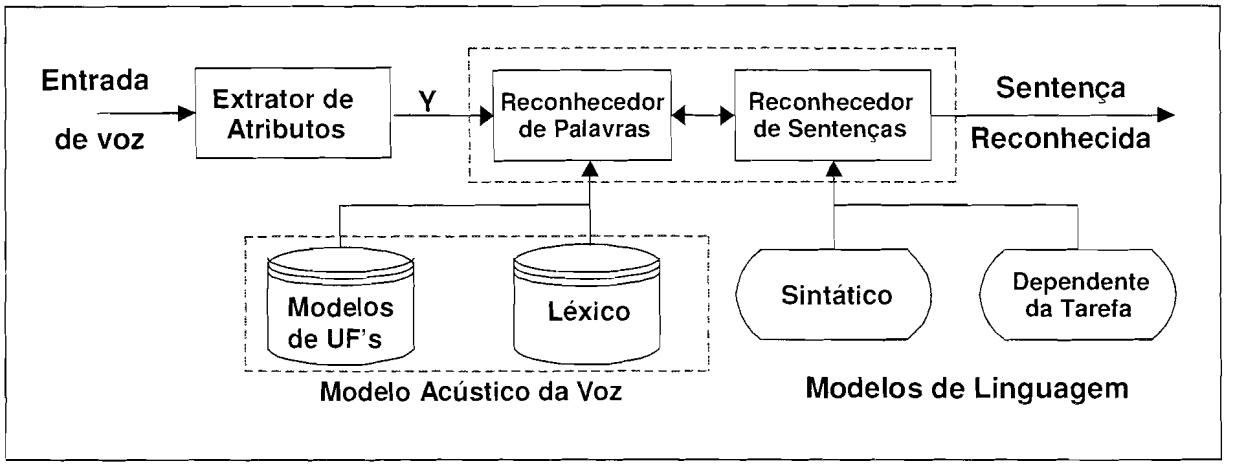


Figura 1. Modelo empregado no sistema.

na Figura 2. Essa estrutura leva em consideração a ordem das classes gramaticais, e como conseqüência, a ordem das palavras e suas interligações.

Associando-se cada estado da estrutura a um símbolo não-terminal e as transições entre estados aos símbolos terminais, cada conjunto de dois estados com uma transição representa uma regra de produção da gramática. Por exemplo,

$$A_i \xrightarrow{p} xA_j$$

significa transitar do estado i para o estado j emitindo o símbolo x com probabilidade p, onde o x pode representar qualquer terminal como por exemplo, classes gramaticais, palavras, fones etc dependendo do nível considerado. As probabilidades podem ser estimadas durante a fase de treinamento utilizando-se o algoritmo *Forward-Backward*.

Na Figura 2, os nós das classes gramaticais dos níveis superiores, como por exemplo, predicado, complemento nominal etc, foram fundidos, para fins de clareza, com os nós iniciais das classes inferiores, como por exemplo, artigo, verbo, pronome etc. Entretanto, suas probabilidades são levadas em consideração na decodificação. Os traços contínuos significam transição com emissão de uma palavra da classe e os traços pontilhados significam transições nulas, ou seja, sem emissão de palavras. Para facilidade de implementação, denominaram-se as classes genericamente de C1, C2, C3 e assim por diante. As probabilidades de transições entre classes e entre palavras (quando se utiliza bigramas) são calculadas durante o treinamento, a partir do conjunto de sentenças de treinamento.

A partir da Figura 2 nota-se que é possível construir sentenças do tipo:

- Eu necessito de uma ligação, a cobrar, para quinhentos e onze, vinte, trinta e um;
- Faça-me uma discagem pro telefone número quatro sete três, oitenta, oito dois;
- Me liga, a cobrar, zero vinte um, dois nove cinco, três dois, três dois;
- Eu desejo dar um telefonema para o número zero oitocentos mil;
- Ligação, a cobrar. O número é quinhentos e doze, treze, treze;
- etc.

Pode-se notar também, pela Figura 2, que dada uma seqüência de classes válida, por exemplo, *pronome* (C1) - *verbo* (C2), poderiam ocorrer algumas combinações de palavras que seriam válidas para a linguagem definida mas inválidas para a língua portuguesa, como por exemplo, *Me desejaria, Eu ligue*. Entretanto, essas combinações seriam descartadas durante o processamento, quando da análise no nível de palavras. Por exemplo, se fossem utilizadas bigramas as probabilidades $P(desejaria/me)$ e $P(ligue/eu)$ seriam iguais a zero.

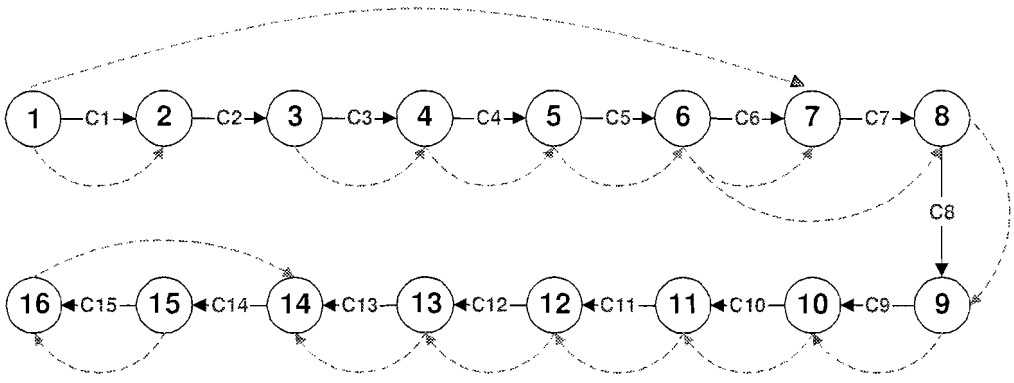
A determinação da máquina de estados a ser utilizada na representação da linguagem é uma parte muito sensível que pode ser diretamente responsável pelo sucesso ou insucesso do sistema. Nos reconhecedores muito grandes, com dicionários maiores que 1000 palavras, haverá a necessidade da utilização de um compilador de linguagem [33], para a sua transcrição sistemática em máquina de estados, e de programas específicos que possam analisar o resultado e fazer minimizações eliminando ramos e produções redundantes, isomorfismos e homeomorfismos.

A máquina de estados da Figura 2 é uma estrutura bastante flexível e desde que o dicionário seja mudado pode ser utilizada em outras tarefas sem perda de eficiência.

3. TREINAMENTO DOS MODELOS

O treinamento dos modelos, em última análise, se traduz na estimativa da matriz de transição A, dos parâmetros da Função Densidade de Probabilidade de Saída (FDS) e do cálculo das probabilidades dos elementos das gramáticas especificadas nos diversos níveis.

Utilizaram-se como modelos acústicos das unidades HMMs contínuos com matrizes covariância diagonal, esquerda-direita (modelos Bakis), com três estados e 5 Gaussianas nas misturas. Concatenando-se esses modelos, de acordo com um dicionário de pronúncias (o léxico e um compilador léxico) obteve-se os modelos das palavras.



C1 = {Eu, Me}
C2 = {ligue, disque, disca, liga, chame, faz, faça, desejo, necessito, quero, preciso, desejaria, queria, gostaria}
C3 = {me}
C4 = {de, do}
C5 = {ligar, falar, discar, chamar, telefonar, fazer, dar, realizar}
C6 = {uma, um}
C7 = {chamada, telefonema, ligação, discagem}
C8 = {a cobrar}
C9 = {para, pra, pro, com}
C10 = {o}
C11 = {telefone}
C12 = {número}
C13 = {é}
C14 = {meia, zero, um dois, três, quatro, cinco, seis sete, oito, nove, dez, onze, doze, treze, quatorze, quinze, dezesseis, dezessete, dezoito, dezenove, vinte, trinta, quarenta, cinqüenta, sessenta, setenta, oitenta, noventa, cem, cento, duzentos, trezentos, quatrocentos, quinhentos, seiscentos, setecentos, oitocentos, novecentos, mil}
C15 = {e}

Figura 2. Máquina de estados que representa a linguagem do sistema proposto.

O algoritmo utilizado na estimativa da matriz A e dos parâmetros da FDS foi o proposto em [34]. Entretanto, o treinamento do sistema que está sendo tratado neste trabalho é bem mais complexo do que o utilizado em [34], onde o dicionário de palavras era composto de 11 unidades e podia-se cobrir todos os contextos possíveis na combinação dos três dígitos.

Uma forma de se obter o número de caminhos viáveis do estado i até outro estado j, em um grafo ou uma máquina de estados, é calculando-se a matriz de transitividade fechada. Esta pode ser obtida pelo produto repetido da matriz de conectividade C, definida da seguinte maneira: para cada par de estados (i, j) no grafo, conte o número de ramos conectando o estado i ao estado j. Esse será o valor de c_{ij}. Desse modo, C indicará o número de caminhos que podem chegar a j, em uma transição, partindo de i.

De forma similar, pode-se definir C^2 cujos elementos c_{ij} serão o número de caminhos partindo de i que podem chegar a j em duas transições e, generalizando-se, pode-se definir C^n cujos elementos serão o número de caminhos que chegam a j, partindo de i, em n transições.

Mostra-se [35] que, utilizando-se os conceitos anteriores, a matriz de transitividade fechada pode ser definida pela seguinte expressão:

$$B = \sum_{i=0}^{\infty} C^i = (I - C)^{-1} \qquad (1)$$

onde I representa a matriz identidade.

É possível verificar que até o estado 14 existem mais de 10^5 caminhos possíveis. Admitindo-se números de 7 dígitos, chega-se a números superiores a 10^{25} caminhos viáveis (sentenças diferentes possíveis). Esse é o número de frases que podem ser geradas pela máquina de estados da Figura 2 e, teoricamente, seria o número de sentenças necessárias para se levar em consideração todos os contextos em que as palavras podem aparecer. Conclui-se, portanto, que o treinamento, por melhor que seja, será sempre limitado e nunca poderá ser suficiente o bastante para se evitar erros na decodificação acústica.

Partindo-se do princípio de que, na prática, é impossível conseguir representações de todos os contextos fonéticos em que uma palavra pode aparecer e, também, que o treinamento apropriado é fundamental para que a mais alta taxa de reconhecimento possível seja obtida para um determinado banco de dados, a escolha das frases para o treinamento deve ser

1) para	2) pra	3) com
4) pro	5) o	6) a
7) e	8) de	9) do
10) eu	11) uma	12) ligar
13) falar	14) discar	15) chamar
16) ligue	17) disque	18) disca
19) liga	20) chame	21) faz
22) faça	23) telefonar	24) fazer
25) desejo	26) necessito	27) quero
28) preciso	29) desejaria	30) queria
31) gostaria de	32) dar	33) chamada
34) telefonema	35) ligação	36) discagem
37) telefone	38) número	39) cento
40) me	41) a cobrar	42) meia
43) zero	44) um	45) dois
46) três	47) quatro	48) cinco
49) seis	50) sete	51) oito
52) nove	53) dez	54) onze
55) doze	56) treze	57) quatorze
58) quinze	59) dezesseis	60) dezessete
61) dezoito	62) dezenove	63) vinte
64) trinta	65) quarenta	66) cinqüenta
67) sessenta	68) setenta	69) oitenta
70) noventa	71) cem	72) duzentos
73) trezentos	74) quatrocentos	75) quinhentos
76) seiscentos	77) setecentos	78) oitocentos
79) novecentos	80) mil	81) gostaria
82) realizar	83) é	

Tabela 1. Palavras do dicionário do sistema para ligações telefônicas automáticas.

feita de forma criteriosa.

Uma maneira de se selecionar frases para o treinamento é utilizar o algoritmo baseado nos bigramas[1] [35]. Esse algoritmo permite construir conjuntos de treinamento em duas fases. Na primeira, seleciona-se uma palavra de um determinado bigrama como ponto de partida. Na segunda, constrói-se sentenças partindo-se da primeira palavra do bigrama em direção ao estado inicial da máquina de estados e depois em direção ao estado final, seguindo-se determinados critérios.

A seleção do par de palavras (bigrama) de partida é crítica para que se obtenha o máximo número de ocorrências de bigramas e o mínimo número de sentenças de treinamento. O processo de seleção é baseado em critérios que consideram a verossimilhança de se gerar sentenças que contenham o maior número de novos bigramas. Esses critérios, em ordem de importância, são os seguintes:

1. O menor número de vezes em que o bigrama aparece;
2. O menor número de vezes em que as palavras do bigrama aparecem;
3. O menor número de sentenças que podem ser construídas a partir do bigrama de início;
4. A distância em que esta bigrama está do estado inicial (preferência para a que estiver mais longe).

Inicialmente a máquina de estados é processada para determinação dos bigramas presentes e da matriz de transitividade fechada. A seguir, aplica-se o primeiro critério, escolhendo-se como bigrama de partida aquele que apareceu o menor número de vezes. Com isto garante-se que os bigramas mais raros apareçam pelo menos uma vez no conjunto de treinamento. Em caso de empates, utiliza-se o segundo critério e assim sucessivamente.

Após a escolha do ponto de partida, realiza-se a segunda fase. Constroem-se sentenças, inicialmente, na direção do estado inicial caminhando-se pela máquina de estado com a ajuda da tabela de bigramas e dos seguintes critérios, em ordem de importância:

1. O menor número de vezes em que os bigramas aparecem;
2. O menor número de bigramas possíveis em direção aos estados antecessores;
3. O menor número de estados (palavras) até o estado inicial;
4. O menor número de sentenças permitidas a partir do estado atual até o estado antecessor.

Esse algoritmo permite que se consiga um conjunto de sentenças para treinamento onde existirá, no mínimo, uma ocorrência de cada bigrama. Entretanto, para gramáticas com um número grande de estados, não se pode garantir que seja um conjunto mínimo (o menor conjunto possível com pelo menos uma ocorrência de cada bigrama).

Durante o treinamento existem algumas ações que devem ser tomadas que são muito importantes e ligadas diretamente com o sucesso ou fracasso do treinamento. Elas são as seguintes:

a) Verificação do resultado da extração dos pontos terminais (*endpoints*);

b) Transcrição atenta das sentenças de treinamento em unidades fonéticas.

O trabalho de transcrição das sentenças em unidades fonéticas além de ser importantíssimo é um trabalho bastante estafante, demorado e que deve ser efetuado com o máximo de atenção, porque, se, por exemplo, um locutor falar /dêzêju/ e um outro /dézêjô/ e essas pronúncias forem transcritas como */dézêjô/*, os modelos das unidades /dé/ e /jô/ terão incorporadas características acústicas de /dê/ e /ju/. Se essas unidades só fossem utilizadas nesses contextos, não haveria problema, entretanto, elas entram na formação de outras palavras que possuem contexto diverso.

[1] A gramática bigrama refere-se às probabilidades $P(k/k')$ onde k é palavra atual e k' a palavra anterior. No presente contexto a palavra bigrama significa um par de palavras.

4. ALGORITMOS DE RECONHECIMENTO

O problema de decodificação no reconhecimento da voz pode ser considerado como o de encontrar a seqüência de palavras mais provável utilizando todas as fontes de conhecimento disponíveis. Para sistemas pequenos, a aplicação direta do algoritmo de Viterbi fornece bons resultados. Porém, para sistemas maiores, existe a necessidade de redução do espaço de busca para aumentar a eficiência computacional além das restrições necessárias para eliminação de caminhos inviáveis.

Existem basicamente duas filosofias em termos de algoritmos de decodificação: os algoritmos baseados no algoritmo de Viterbi e os baseados no A* *Search* [36]-[40]. Os mais utilizados são os baseados no algoritmo de Viterbi.

Os decodificadores baseados no algoritmo de Viterbi podem ser classificados naqueles que procuram a melhor hipótese (*one best search*) e naqueles que procuram as N melhores hipóteses (*N-best search*) e fazem múltiplas passagens, utilizando fontes de conhecimento diferentes. Os algoritmos de múltiplas passagens são utilizados normalmente com dicionários que contêm mais de 1000 palavras [39].

Neste trabalho utilizaremos decodificadores que procuram a melhor hipótese em uma única passagem.

A busca exaustiva utilizando o algoritmo de Viterbi produz a melhor seqüência que se pode obter para os dados apresentados. Possui ainda a vantagem de ser síncrona no tempo, evitando-se assim, a necessidade de normalização para hipóteses de comprimentos diferentes - uma das desvantagens dos algoritmos baseados no A* *Search*. Entretanto, computacionalmente, essa busca é muito onerosa. A solução é utilizar um algoritmo sub-ótimo chamado de *Viterbi beam search* [38]-[40], idêntico ao *Viterbi search*, exceto pelo fato de que as hipóteses cujas verossimilhanças fiquem abaixo de um determinado limiar são excluídas de futuras considerações. A utilização do *beam search* evita a necessidade de se calcular a verossimilhança para todos os estados e pode levar a uma substancial economia no esforço computacional com pouca ou quase nenhuma perda no desempenho.

Um algoritmo capaz de realizar o *Viterbi search* de forma eficiente é o *Level Building* [2]. Contudo, para dicionários médios e grandes, o esforço computacional se torna muito grande pois o número de cálculos é proporcional a $T \times L$ para cada palavra, onde T é o comprimento da sentença (número de quadros da locução de entrada) e L o número de níveis.

Um algoritmo mais eficiente e que produz os mesmos resultados do *Level Building*, para a busca exaustiva, é o *One Pass* [39]. Nele, o número de quadros é proporcional a $T \times K$ onde K é o número de palavras. O algoritmo *One Pass* é básico para dezenas de modificações que visam diminuir o espaço de busca, sendo usado inclusive, pelos algoritmos que efetuam múltiplas passagens.

A combinação do *One Pass* (OP) com a estratégia do *Beam search* é chamada de *Time-Synchronous Viterbi Beam Search* [38],[39]. Esse algoritmo utiliza basicamente duas equações da programação dinâmica:

$$D(t,j,k) = -\log[b_j(O_t)] + \min_i \{D(t-1,i,k) + -\log(a_{ij})\}, \quad i = 0,\ldots,j \tag{2}$$

e

$$B(t,j,k) = B(t-1, I_{\min}(t,j,k), k), \tag{3}$$

onde

$$I_{\min}(t,j,k) = \arg\min_i \{D(t-1,i,k) - \log(a_{ij})\}, \quad i = 0,\ldots,j. \tag{4}$$

A Eq. (2) fornece o escore (verossimilhança) do melhor caminho que termina no estado j da palavra k no instante t, onde $b_j(O_t)$ é a probabilidade da observação O no instante t estar no estado j, a_{ij} é a probabilidade de transição do estado i para o estado j, e a Eq. (3) fornece o instante de início do melhor caminho que termina no estado j da palavra k no instante t.

O OP pode ser visto como a busca do melhor caminho no conjunto de pontos (t,j,k), mostrado na Figura 3, que proporciona o melhor casamento entre a seqüência de vetores de entrada O_1^T e a seqüência de estados das palavras de referência. Esse caminho deve satisfazer determinadas restrições de continuidade e regras de transições, intra-palavras e entre-palavras. Para isso, cria-se um ponto artificial na grade de pontos, $i = 0$, chamado de nó de linguagem, que servirá de verificador para o início de novas palavras. Após o processamento de todos os pontos (t,j,k) para um dado instante t, o escore do melhor caminho levando ao nó de linguagem deve ser computado para a verificação do início de uma nova palavra. Para isso, o escore $D(t,0,k)$ é definido como:

$$D(t,0,k) = \min_{k'} \{D(t, J(k'), k') + C(k)\}, \tag{5}$$

$$k' = 1,\ldots,K.$$

onde $C(k)$ é o negativo do logaritmo da probabilidade fornecida pelo modelo de linguagem e $J(k)$ é o número de estados da palavra k. No caso da gramática bigrama $C(k) = P(k|k')$.

Para evitar a armazenagem das decisões para cada ponto da grade (t,j,k), define-se um ponteiro $B(t,j,k)$ que mantém o índice do último quadro da melhor palavra antecessora de referência. Nos nós de linguagem, $B(t,0,k)$ é iniciado com o índice do quadro anterior:

$$B(t,0,k) = t-1. \tag{6}$$

No interior das palavras, essa informação é propagada pelos pontos da grade através das equações (3) e (4).

O algoritmo processa todos os padrões de referência sincronizados com o tempo, movendo-se ao longo do eixo do vetor de observações realizando uma decodificação estritamente da esquerda para a direita, estendendo todas as hipóteses das seqüências de palavras em paralelo. A implementação requer três ciclos, um para os quadros de entrada, um para as palavras de referência e um para os estados de cada palavra de referência. Após o processamento do último quadro, a seqüência de palavras decodificada é obtida, utilizando-se os vetores de retorno, $B(\bullet)$, (*traceback arrays*).

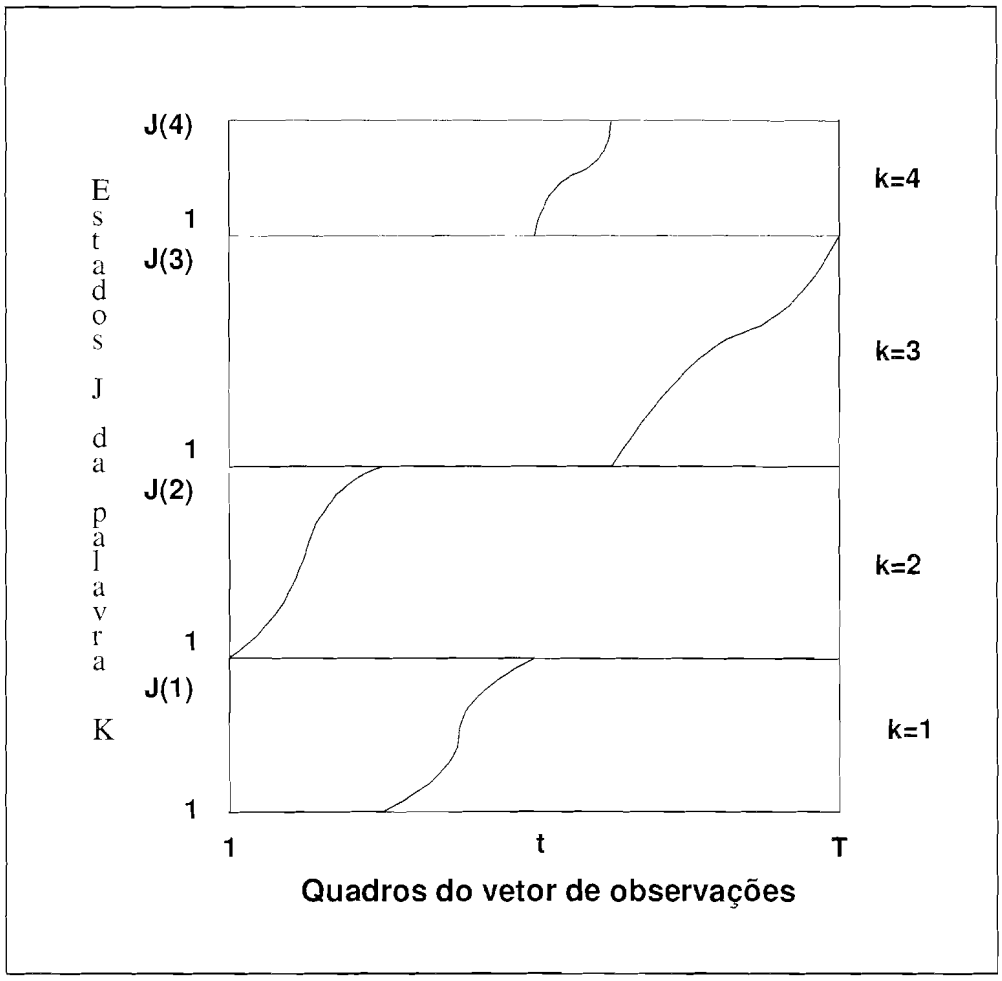


Figura 3. Grade de busca do algoritmo *One Pass*.

A exclusão dos caminhos inviáveis é efetuada nos nós de linguagem excluindo-se de futuras considerações os caminhos que não satisfizerem a seguinte condição:

$$D(t,0,k) < D(t,0,k^*) + \delta. \qquad (7)$$

onde $D(t,0,k^*)$ é a distância com menor valor entre os nós de linguagem no instante t e δ um limiar apropriado [36]-[40]. A exclusão também pode ser efetuada em cada estado (*Frame-Synchronous Viterbi Beam Search*) [40] diminuindo-se ainda mais o espaço de decodificação. Entretanto, a escolha do limiar "correto" é bem problemática, uma vez que dependendo dessa escolha, pode-se cortar prematuramente a seqüência de palavras verdadeira.

Existem diversas modificações que podem ser efetuadas no algoritmo OP com os objetivos, normalmente conflitantes entre si, de diminuir o esforço computacional e melhorar o desempenho. As principais são:

- a que leva em consideração o fato de que muitas palavras possuem os mesmos fonemas iniciais e utiliza o dicionário de pronúncias em árvores na decodificação, chamado de *Predecessor Conditioned One-Pass Algorithm* [39]. Normalmente é empregado para sistemas com dicionários maiores que 20.000 palavras; e
- a que utiliza pares de palavras em grafos. Esse algoritmo, chamado de *Word Graph Algorithm* [39], utiliza propriedades dos grafos em conjunto com o *Predecessor Conditioned One-Pass Algorithm*.

Essas modificações não foram consideradas no presente trabalho, tendo em vista que tornam a implementação mais complexa e, para o tamanho do dicionário utilizado, não adicionam nenhuma vantagem na decodificação pois muito poucas palavras possuem os mesmos fonemas iniciais.

4.1 RESULTADOS DE SIMULAÇÕES

Nesta Seção, por motivos operacionais, são apresentados apenas os resultados no modo dependente do locutor. O sinal de voz foi amostrado a uma taxa de 11025 Hz, pré-enfatizado com filtro $H(z) = 1 - 0{,}95z^{-1}$, dividido em janelas de 20 ms e quadros de 10 ms. Após o produto das amostras de cada janela com a janela de Hamming foram extraídos 12 coeficientes Mel-Cepestro, 12 ΔMel-Cepestros, 12 Δ^2Mel-Cepestros, $\log E$, $\Delta \log E$ e $\Delta^2 \log E$, totalizando 39 atributos.

Para realização dos experimentos, um determinado locutor gravou 30 repetições de cada unidade fonética constante do inventário 1 descrito em [6], 20 repetições de cada palavra constante da Tabela 1 e 154 locuções para treinamento geradas pelo algoritmo descrito na Seção 3.

Na fase de treinamento, as unidades foram modeladas por HMMs contínuos esquerda-direita, com 3 estados e 5 Gaussianas por mistura. Para estimação dos seus parâmetros foram utilizados o procedimento de Viterbi e o algoritmo descrito em [35].

Para teste do sistema utilizaram-se 30 sentenças geradas pelo mesmo algoritmo que gerou as sentenças de treinamento, mas com bigrama inicial diferente e 11 sentenças escolhidas para levar em consideração contextos ambíguos e repetições de dígitos, totalizando 41 sentenças com média de 12,3 palavras por sentença.

A Tabela 2 mostra os resultados das simulações realizadas utilizando-se a máquina de estados apresentada na Figura 2 e o Beam Search em todos os algoritmos de decodificação. Pode-se notar que o pior resultado do algoritmo *One Pass* ocorreu devido ao grande número de inserções ocorridas, principalmente da palavra UM que foi inserida 31 vezes. Essas inserções foram provocadas basicamente por dois motivos:

1. O OP não possui um limite para o número máximo de palavras por sentença;
2. Algumas palavras podem ocorrer várias vezes dentro da mesma sentença podendo proporcionar um caminho errado como aconteceu para a frase: *Eu desejaria fazer uma discagem para o número nove nove oito dois um três quatro* que foi decodificada como:
   Eu desejaria fazer uma discagem para o número UM TELEFONEMA PRA nove nove oito UM dois um três quatro.

	I	D	S	TOTAL	Taxa de Acerto
Level Building	23	10	16	49	90,25%
One Pass	65	0	15	80	84,10%
One Pass *	33	0	15	48	90,45%

*** com pós-processador sintático**
I = Inserções **D** = Deleções **S** = Substituições

Tabela 2. Resultado dos testes com a estrutura da Figura 2.

A utilização de um pós-processador sintático simples permitiu a redução dos erros de inserção em aproximadamente 50%. O processador utilizou as seguintes regras:

a) Eliminar palavras repetidas, exceto números;

b) verificar se a palavra UM é artigo ou número pela sua posição na frase e eliminar as repetições em caso de artigo;

c) eliminar palavras repetidas por classe - cada classe só pode ter uma palavra vencedora. Por exemplo: *Eu desejo* (nível 2) *quero* (nível 2) *ligar para o telefone* ...

d) eliminar palavras fora do contexto. Por exemplo: *Preciso DO telefonar pro número*...

Essas regras são simples de aplicar e computacionalmente o custo foi mínimo. Contudo, o algoritmo *Level Building* foi mais eficiente no aproveitamento das restrições impostas pela gramática, não necessitando de um pós-processamento.

5. INCORPORAÇÃO DE CONHECIMENTOS DEPENDENTES DA TAREFA

Os erros ocorridos nas simulações cujos resultados estão apresentados na Tabela 2 foram ocasionados, em grande parte, pela decodificação acústica incorreta ou por ambigüidades no nível acústico. Existem vários níveis de conhecimentos (sintático, semântico e pragmático) que podem ajudar a reduzir esse número de erros.

Experimentos psicológicos mostram que mesmo os seres humanos utilizam essas fontes de conhecimento e quando algum trecho de voz a ser reconhecido viola alguns desses níveis ou quando o ouvinte não possui o conhecimento apropriado, a taxa de reconhecimento decresce significantemente. Por exemplo, o reconhecimento pelos humanos cai quando as sentenças são ouvidas fora do contexto (pragmaticamente inconsistente) ou são sem sentido (semanticamente inconsistente) ou são numa linguagem desconhecida (sintaticamente inconsistente) [42]-[44]. Portanto, a utilização desses níveis é fundamental para a melhoria da taxa de acerto dos reconhecedores de voz contínua.

O processamento sintático normalmente é efetuado durante a decodificação de acordo com a gramática representada pela máquina de estados. O processamento semântico pode ser efetuado durante a decodificação, se o referido conhecimento for incorporado à máquina de estados ou após a decodificação. Esse tipo de processador elimina seqüências de palavras sintaticamente válidas, mas que carecem de significado.

Analisando-se a tarefa e o contexto em que o sistema vai ser utilizado pode-se impor restrições, baseadas nos níveis de conhecimentos citados, sem necessariamente construir-se processadores separados, guiando-se pelo conhecimento dependente da tarefa. Basicamente essas regras podem ser incorporadas às restrições sintáticas já existentes.

Esse conhecimento foi incorporado à máquina de estados, mostrada na Figura 2, acrescentando-se estados extras e modificando-se algumas transições com o objetivo de limitar os caminhos viáveis àqueles mais utilizados na prática.

Por exemplo, suponha que durante a decodificação, na transição do estado 2 para o estado 3 da Figura 2, a palavra FAÇA tivesse sido a vencedora (palavra associada ao modelo que produziu a maior verossimilhança). Todas as palavras pertencentes às classes existentes até o estado 15 teriam que ser colocadas como hipóteses, embora os bigramas, $P(W_i/\text{FAÇA})$, da maioria delas sendo zero, provocassem suas exclusões na fase de exclusão (*pruning*). Uma rápida análise permite verificar que apenas as palavras das classes 3 e 6 necessitariam ser colocadas como hipóteses nesse caso.

Assim, a incorporação de conhecimentos dependentes da tarefa permitirá a economia de recursos computacionais e guiará a decodificação de forma que somente palavras que produzam sentenças semanticamente válidas sejam possíveis. Desse modo pode-se evitar que seja apresentada sentença do tipo

Faça de uma chamada a cobrar, o telefone número ...

que embora sintaticamente correta é semanticamente incorreta. Ou ainda, que sejam apresentadas sentenças corretas semântica e sintaticamente, mas fora do contexto da aplicação como, por exemplo,

Ligue de um telefone: cinco quatro ....

Para a incorporação dessas restrições, as sentenças parciais permitidas pela linguagem até o estado 14 da Figura 2 foram divididas em grupos de frases típicas. Com base nesses grupos, determinadas classes foram substituídas por outras, da seguinte forma:
C1 → {Eu}, {Me}
C2 → {desejo, necessito, quero, preciso, desejaria, queria, gostaria de}, {necessito, preciso}, {necessito, preciso, gostaria}, {ligue, disque, disca liga, chame}, {faz, faça}
C4 → {de}, {do}
C5 → {ligar, falar, discar, chamar, telefonar}, {ligar, discar, chamar}, {fazer, realizar}, {dar}
C6 — {uma}, {um}
C7 — {chamada, ligação, discagem}, {telefonema}
C9 — {para, pra, com}, {pro}.
As outras classes permaneceram inalteradas.

A Figura 4 mostra a máquina de estados após a incorporação das novas classes onde as suas denominações foram modificadas para se obter maior clareza. A parte correspondente aos cardinais, entre os estados 20 e 34, serão detalhadas a seguir.

Cada região desenvolve sua maneira de falar os números no momento de pedir uma ligação telefônica de acordo com as características da língua, do sistema telefônico e das tradições. Nos EUA, o mais comum é falar o número na forma de dígitos [44], na Dinamarca, por ser um sistema com oito números para as chamadas locais, eles são falados como grupos de 4 dezenas [45] e assim por diante.

No Brasil, costuma-se falar os números telefônicos impondo-se um ritmo à pronúncia - agrupando-se os números em conjuntos de três dígitos para o número de interurbanos e da central telefônica e em conjuntos de dois dígitos para o restante. Dependendo do número, ele pode ser dito na forma de dígitos ou de números naturais. Por exemplo, o número 542 47 85 normalmente é pronunciado na forma de dígitos, enquanto que o número 213 15 11 é mais provável que seja pronunciado como *duzentos e treze, quinze, onze* e raramente como *vinte e um, trinta e um, cinqüenta e um um*. De forma semelhante, o número 0800 22 1000 certamente será pronunciado como *zero oitocentos vinte e dois mil*.

Esses costumes regionais podem ser levados em consideração na construção da máquina de estados dos números com o objetivo de diminuir o espaço de decodificação bem como melhorar a taxa de acerto. Como o sistema ficará mais restritivo quanto às formas de pronúncias, é óbvio que as pessoas que falarem de uma maneira que fuja à regra geral terão que pedir suas ligações novamente.

A Figura 5 apresenta uma proposta de máquina de estados que incorpora esses conhecimentos dependentes da tarefa. Essa máquina considera apenas ligações locais com no máximo 7 dígitos. Essa limitação foi colocada simplesmente por uma questão de economia de recursos computacionais e de tempo para os testes. Entretanto, facilmente pode-se expandí-la para ligações locais com oito dígitos ou incorporação de ligações interurbanas, internacionais e de serviço.

A Tabela 3 apresenta os resultados dos testes realizados utilizando-se as estruturas apresentadas nas Figuras 4 e 5. O processamento acústico e as elocuções de treinamento e teste foram idênticas às utilizadas nos testes da seção anterior.

ALGORITMO	I	D	S	TOTAL	Taxa de Acerto
Level Building	20	0	12	32	93,64%
One Pass	9	3	13	25	95,03%

I = Inserções **D** = Deleções **S** = Substituições

Tabela 3. Resultado dos testes utilizando-se as máquinas de estado das Figuras 4 e 5.

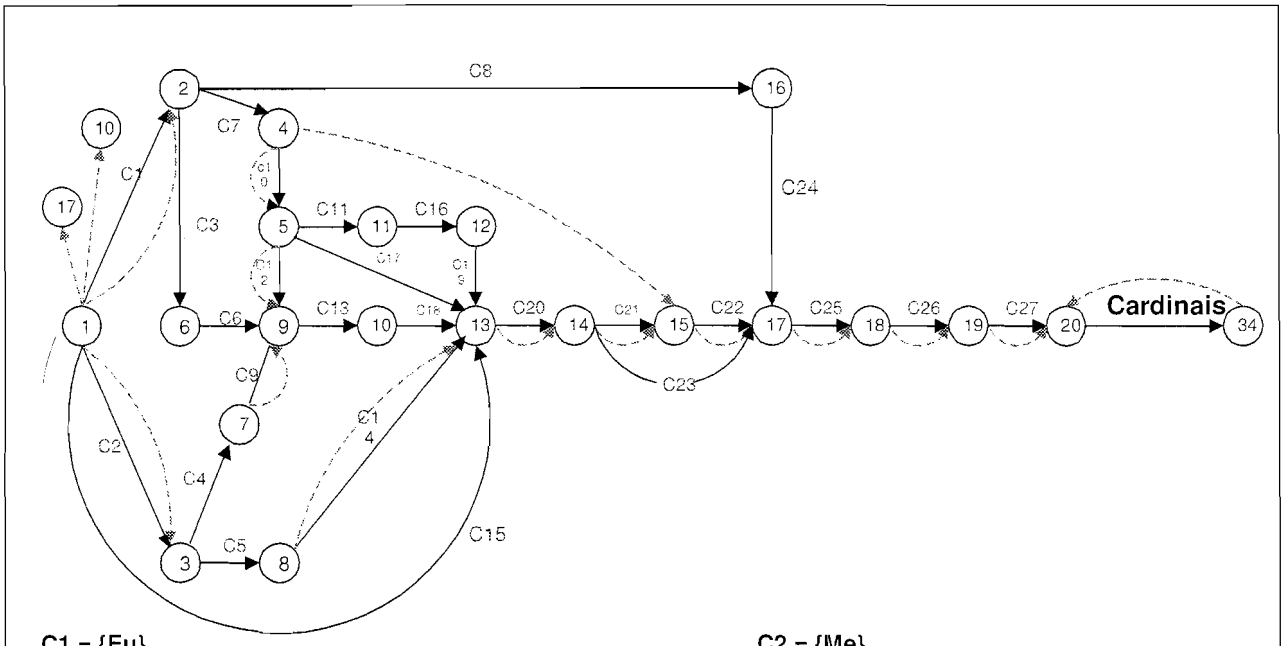
Pode-se verificar que a incorporação das restrições proporcionou uma redução da taxa de erro de aproximadamente 34,8% para o LB e de 47,96% para o OP. Os tipos de erros verificados não permitiram a utilização do processador sintático utilizado na Seção anterior.

A grande melhoria apresentada pelo algoritmo OP em relação ao LB está relacionada com a forma de utilização dos conhecimentos incorporados à máquina de estados.

O algoritmo LB pela sua própria característica, pode aproveitar de modo direto as restrições apresentadas pela gramática, ou seja, pode associar os níveis do algoritmo com as transições entre os estados da máquina, estabelecendo-se assim, uma relação biunívoca entre esses níveis de decodificação e as classes sintáticas. O algoritmo OP, entretanto, não possui essa relação direta pois cada nível é composto de apenas uma palavra e, durante a decodificação, esses níveis são processados a cada instante de tempo t. Sendo assim, as únicas restrições que o algoritmo pôde levar em consideração nos testes da Seção anterior foram as bigramas.

Para que o OP incorpore as restrições impostas pela máquina de estado de modo mais eficiente ou faz-se múltiplas decodificações utilizando-se fontes de conhecimento diferentes a cada passagem (abordagem que não foi utilizada neste trabalho) ou criam-se níveis com palavras repetidas um para cada palavra de cada classe e criam-se conjuntos de palavras antecessoras para cada repetição. Por exemplo, para o sistema da Figura 4 foram feitas duas cópias da palavra DISCAR, uma para a classe 15 (cópia 1) e uma para a classe 17 (cópia 2). Para a cópia 1 o conjunto de palavras antecessoras foi o conjunto vazio, ou seja, é uma palavra que pode iniciar uma sentença. Para a cópia 2 o conjunto antecessor foi composto das palavras {me, necessito, quero, preciso desejaria, queria, gostaria de}. Dessa forma, de maneira simples mas trabalhosa, consegue-se durante a decodificação, levar em consideração as restrições impostas pela máquina ou mesmo restringir o número de palavras por sentença.

Essa maneira de incorporar os conhecimentos no OP possui duas desvantagens:



C1 = {Eu}
C2 = {Me}
C3 = {necessito, preciso}
C4 = {faz, faça}
C5 = {ligue, disque, disca, liga, chame}
C6 = {de}
C7 = {desejo, necessito, quero, preciso, desejaria, queria, gostaria de}
C8 = {necessito, preciso, gostaria}
C9 = {me}
C10 = {me}
C11 = {dar}
C12 = {fazer, realizar}
C13 = {uma}
C14 = [me}
C15 = {ligar, discar, chamar}
C16 = {um}
C17 = {ligar, falar, discar, chamar, telefonar}
C18 = {chamada, ligação, discagem}
C19 = {telefonema}
C20 = {à cobrar}
C21 = {para, pra, com}
C22 = {o}
C23 = {pro}
C24 = {do}
C25 = {telefone}
C26 = {número}
C27 = {é}

Figura 4. Máquina de estados do sistema proposto após a incorporação dos conhecimentos dependentes da tarefa. Cada arco contínuo corresponde a uma transição com emissão de uma palavra da classe. Os traços pontilhados correspondem a transições nulas.

1. O número de palavras no dicionário aumenta porque cada cópia é considerada uma palavra diferente;

2. a máquina de estados tem que ser analisada em busca de classes que possuam palavras iguais de modo a criar cópias de palavras que compartilhem mais de uma classe e um conjunto de palavras antecessoras. Para dicionários muito grandes isso se torna inviável.

Entretanto, a incorporação de conhecimentos no OP proporciona uma melhoria significativa na taxa de reconhecimento.

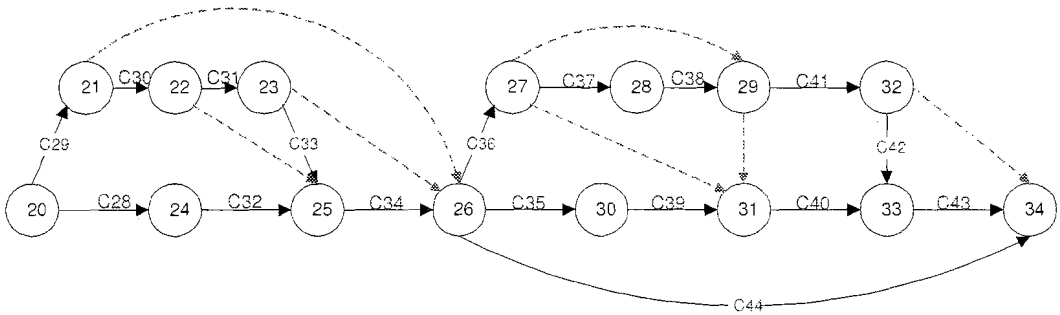
6. CONCLUSÕES

Foi proposto um sistema de reconhecimento de voz contínua para ligações telefônicas automáticas em língua portuguesa com um vocabulário composto de 83 palavras, o que permite ao sistema um reconhecimento de mais de 10^{25} sentenças diferentes.

Foi apresentado um método para treinamento, bem como a consolidação da utilização das unidades fonéticas propostas para o português apresentadas no inventário 1 descrito em [6].

Neste artigo, também foram propostas duas estruturas para modelagem do sistema que levaram em consideração particularidades do idioma português. O modelamento utilizando classes gramaticais em conjunto com a utilização do algoritmo OP, com pós-processador sintático, permitiu uma taxa de reconhecimento de 90,45%. A utilização de uma estrutura dependente da tarefa que levou em consideração particularidades da formação de sentenças e a incorporação de conhecimentos a respeito do costume na pronúncia dos números telefônicos, permitiu atingir uma taxa de reconhecimento de 95,03%, utilizando-se o algoritmo OP.

O algoritmo OP com palavras repetidas para levar em consideração as restrições impostas pelas máquinas de estado das Figuras 4 e 5 proporcionou resultados melhores que os obtidos pelo algoritmo LB. Entretanto, a utilização do LB é mais intuitiva (associação direta nível → transição) e menos trabalhosa.



C28 = {dois, três, quatro, cinco, seis, sete, oito, nove}
C29 = {cem, cento, duzentos, trezentos, quatrocentos, quinhentos, seiscentos, setecentos, oitocentos, novecentos}
C30 = {e}
C31 = {dez, onze, doze, treze, quatorze, quinze, dezesseis, dezessete, dezoito, dezenove, vinte, trinta, quarenta, cinquenta, sessenta, setenta, oitenta, noventa}
C32 = {zero, um, dois, três, quatro, cinco, seis, sete, oito, nove}
C33 = {e}
C34 = {zero, um, dois, três, quatro, cinco, seis, sete, oito, nove}
C35 = {zero, um, dois, três, quatro, cinco, seis, sete, oito, nove}
C36 = {dez, onze, doze, treze, quatorze, quinze, dezesseis, dezessete, dezoito, dezenove, vinte, trinta, quarenta, cinquenta, sessenta, setenta, oitenta, noventa}
C37 = {e}
C38 = {zero, um, dois, três, quatro, cinco, seis, sete, oito, nove}
C39 = {zero, um, dois, três, quatro, cinco, seis, sete, oito, nove}
C40 = {zero, um, dois, três, quatro, cinco, seis, sete, oito, nove}
C41 = {dez, onze, doze, treze, quatorze, quinze, dezesseis, dezessete, dezoito, dezenove, vinte, trinta, quarenta, cinquenta, sessenta, setenta, oitenta, noventa}
C42 = {e}
C43 = {zero, um, dois, três, quatro, cinco, seis, sete, oito, nove}
C44 = {mil}

Figura 5. Máquina de estados para decodificação de números telefônicos de 7 dígitos após a incorporação dos conhecimentos da tarefa.

REFERÊNCIAS

[1] P. Price et al.,"The DARPA 1000-Word Resource Management Database for Continuous Speech Recognition", Proc. ICASSP'88, pp. 651-654, Setembro 1988.

[2] J. R. Deller, J. G. Proakis e J. H. L. Hansen, "Discrete-Time Processing of Speech Signals", Ed. Macmillan Publishing Company, New York, 1993.

[3] J. L. Gauvain, L. Lamel e M. Adda-Decker, "Developments in Continuous Speech Dictation Using the ARPA WSJ Task", Proc. ICASSP'95, pp. 65-68, Setembro 1995.

[4] M. Adda-Decker, G. Adda et al.,"Developments in Large Vocabulary Continuous Speech Recognition of German", Proc. ICASSP'96, Setembro 1996.

[5] D. O'Shaughnessy, "Speech Communication - Human and Machine", Addison-Wesley, 1987.

[6] S.C.B. Santos and A. Alcaim, "Reduced Sets of Subword Units for Continuous Speech Recognition of Portuguese", Electronics Letters, 36(6), pp. 586-588.

[7] A. Alcaim, J. A. Solewicz e J. A. Moraes, "Freqüência de Ocorrência dos Fones e Listas de Frases Foneticamente Balanceadas no Português Falado no Rio de Janeiro", Revista da Sociedade Brasileira de Telecomunicações, 7(1), 23-41.

[8] I. Sanches e N. Alens, "A Speaker-Independent Digit Recognizer", ITS 90, Rio de Janeiro, pg 9.6.1 -9.6.5.

[9] F. J. F. Silva, "Implementação em Tempo Real de um Reconhecedor de Dígitos Isolados Independentemente do Locutor", 9º SBT, São Paulo-SP, Setembro de 1991.

[10] C. A. Sanbuichi e B. G. Aguiar Neto, "Reconhecimento de Palavras Isoladas Independente de Locutor Para Língua Portuguesa do Brasil", 9º SBT, São Paulo-SP, Setembro de 1991.

[11] S. C. B. Santos e R. R. Goldschmidt, "Reconhecimento de Palavras Isoladas Utilizando Redes Neurais", TELEMO92, Brasília-DF.

[12] G. A. Carrijo e M. G. S. Figueiredo, "Reconhecimento de Palavra Isolada Utilizando Quantização de Vetores e Redes Neurais Artificiais", TELEMO92, Brasília-DF, Julho de 1992.

[13] A. A. Carvalho, "Reconhecimento Automático de Fonemas

Explosivos Surdos em Língua Portuguesa", 11º SBT, Natal-RN, Setembro de 1993.

[14] R. D. R. Fagundes e N. Alens, "Reconhecimento de Voz, Linguagem Contínua, Usando Modelos de Markov", 11º SBT, Natal-RN, Setembro de 1993.

[15] M. Minami e N. Alens, "Um Sistema de Reconhecimento de Palavras Isoladas por HMM Discreto, Para a Linha Telefônica", 13º SBT, Águas de Lindóia-SP, Setembro de 1995.

[16] H. F. Nunes, F. Violaro e F. O. Runstein, "Reconhecimento de Dígitos Conectados Através da Fala", 13º SBT, Águas de Lindóia-SP, Setembro de 1995.

[17] J. A. Solewicz, J. A. Moraes e A. Alcaim, "Text-to-Speech System for Brazilian Portuguese Using a Reduced Set of Synthesis Units", Proc. of International Symp. on Speech, Image Process. and Neural Networks, Hong Kong, abril 1994.

[18] H. F. Nunes e F. Violaro, "Reconhecimento de Fala Com Vocabulário Flexível Para o Português Brasileiro Utilizando HMM", TELEMO-96, Setembro, 1996, Curitiba.

[19] I. Sanches, "Aplicação da Relação Sinal-Ruído Intra-Palavra no Reconhecimento Automático de Fala em Ruído", TELEMO-96, Setembro, 1996, Curitiba.

[20] R. A. B. Sória e E. F. Cabral Jr.,"Comparison of Different Neural Paradigms in a Speaker Recognition Task Using Mel-Frequency Cepstral Coefficients Correlations", TELEMO-96, Setembro, 1996, Curitiba.

[21] F. Violaro, B. Kaspar e J. A. Martins, "Isolated Word Recognition Using Hidden Markov Models", TELEMO-96, Setembro, 1996, Curitiba.

[22] C. T. Ishi, R. A. S. Passos e O. Saotome, "Análise e Implementação de um sistema Reconhecedor de Voz", XV SBT, Setembro 1997, Recife-PE.

[23] F. J. F. da Silva e O. Saotome, "Reconhecimento de Fala com Vocabulário Ilimitado Para o Português", XV SBT, Setembro 1997, Recife-PE.

[24] S. C. B. dos Santos e A. Alcaim, "Inventários Reduzidos de Unidades Fonéticas do Português Brasileiro", XV SBT, Setembro 1997, Recife-PE.

[25] J. V. Filho, P. Scalart e J. G. Chiquito, "Redução de Ruído em Sinais de Voz Captados em Veículos", XV SBT, Setembro 1997, Recife-PE.

[26] E. S. Morais e F. Violaro, "Sistema Híbrido ANN-HMM para Reconhecimento de Fala Contínua", XV SBT, Setembro 1997, Recife-PE.

[27] C. A. Ynoguti e F. Violaro, "Explorando Redundâncias do Sinal de Voz para Reduzir o Tempo de Reconhecimento em Sistemas Utilizando Redes Neurais", XV SBT, Setembro 1997, Recife-PE.

[28] F. Runstein, F. Violaro e C. H. Silva, "Desempenho de um Sistema de Reconhecimento de Fala Baseado em Redes neurais", XV SBT, Setembro 1997, Recife-PE.

[29] I. Sanches, "Ajuste da Matriz de Covariância em HMM Contínuo sob Ruído", XV SBT, Setembro 1997, Recife-PE.

[30] H. F. Nunes, E. J. Nagle e C. H. da Silva, "Segmentação Fonêmica Automática para Frases do Português Brasileiro Utilizando-se HMM", XV SBT, Setembro 1997, Recife-PE.

[31] M. Minami, N. Alens e I. Sanches, "Sistema Reconhecedor de Palavras Isoladas, com HMM-VQ, Múltiplos Livros de Códigos e Quantização Vetorial de Energia para a Linha Telefônica", XV SBT, Setembro 1997, Recife-PE.

[32] J. G. Guimarães, E. C. Negreiros, L. M. Silva e A. R. S. Romariz, "Comparação de Técnicas de Ajuste Temporal Para Reconhecimento de Palavras Isoladas com Redes Neurais", XV SBT, Setembro 1997, Recife-PE.

[33] M. K. Brown e J. G. Wilpon, "A Grammar Compiler for Connected Speech Recognition", IEEE Trans. ASSP, Vol 39, No 1, pp 17-28, Janeiro 1991.

[34] S.C.B. Santos e A. Alcaim, "Treinamento de Modelos de Unidades Fonéticas com Variabilidade Acústica em Reconhecedores de Voz Contínua Baseados em CDHMM", Revista da Sociedade Brasileira de Telecomunicações, 15(1), Junho 2000, pp. 34-43.

[35] M. K. Brown, M. A. McGee e L. R. Rabiner, "Trainning Set Design for Connected Speech Recognition", IEEE Trans. on Signal Processing, Vol 39, No 6, Junho 1991.

[36] Kai-Fu Lee, "Automatic Speech Recognition: The Development of the SPHINX System", Kluwer Publishers, Boston 1989.

[37] N. Deshmukh e J. Picone, "Methodologies for Language Modeling and Search in Continuous Speech Recognition", Technical Report, Institute for Signal Processing & Information Processing, Mississipi State University, MS State, 1996.

[38] N. Deshmukh, "Efficient search Algorithms for Large Vocabulary Continuous Speech recognition", Technical Report, Institute for Signal and Information Processing, Department of Electrical and Computer Engineering, Mississipi State University, Mississipi State, 1996.

[39] C. H. Lee, F. K. Soong e K. K. Paliwal, "Automatic Speech and Speaker Recognition - Advanced Topics", Kluwer Academic Publishers, 1996.

[40] N. Deshmukh, J. Picone e Y. Kao, "Efficient Search in Hierarchical Pattern Recognition Systems", Institute for Signal and Information Processing, Department of Electrical and Computer Engineering, Mississipi State University, Mississipi State

[41] W. A. Lea, "Trends in Speech Recognition", Ed. Prentice-Hall, New Jersey, 1980.

[42] W. J. Ebel e J. Picone, "Human Speech Recognition Performance on the 1994 CSR Spoke 10 Corpus", Technical Report, Institute for Signal and Information Processing, Department of Electrical and Computer Engineering, Mississipi State University, Mississipi State, 1994.

[43] N. Deshmukh, A. Ganapathiraju, R. J. Duncan e J. Picone, "Human Speech Recognition Performance on the 1995 CSR HUB-3 Corpus", Technical Report, Institute for Signal and Information Processing, Department of Electrical and Computer Engineering, Mississipi State University, Mississipi State, 1995.

[44] L. Rabiner e B. Juang, "Fundamentals of Speech Recognition", Ed. Prentice-Hall, New Jersey, 1993.

Sidney Cerqueira Bispo dos Santos graduou-se em engenharia elétrica com ênfase em Telecomunicações em 1985 no Instituto Militar de Engenharia (IME). Foi declarado Mestre em Ciências em 1989 também pelo IME e Doutor na área de Sistemas de Telecomunicações pela PUC-Rio, em 1997. Suas pesquisas estão voltadas para a área de Reconhecimento de Padrões e Processamento Digital da Voz principalmente no que se refere às interfaces homem-máquina. Até Setembro de 2000 foi Chefe (Diretor) do Departamento de Engenharia Elétrica do IME e em outubro de 2000 assumiu a Gerência de Telecomunicações da Secretaria Executiva do Conselho Deliberativo do SIPAM. Atualmente, é gerente Técnico do Centro Gestor e Operacional do SIPAM - CENSIPAM, responsável por operacionalizar o sistema de Redes e Telecomunicações que dará suporte ao Sistema de Proteção da Amazônia.

Abraham Alcaim recebeu o diploma de Engenheiro Eletricista e o título de Mestre em Ciências em Engenharia Elétrica pela Pontifícia Universidade Católica do Rio de Janeiro (PUC/Rio) em 1975 e 1977, respectivamente, e os títulos de D.I.C. e Ph.D. pelo Im-

perial College of Science and Technology, University of London, em 1981. Desde 1976 ele é professor do Centro de Estudos em Telecomunicações da Universidade Católica (CETUC), tendo atualmente o cargo de Professor Associado. O Dr. Alcaim trabalha há 25 anos nas áreas de processamento digital de voz e imagem. Ele é autor de diversos artigos publicados em congressos e revistas nacionais e internacionais. Em 1984 ele esteve por um período curto com o Centre National d'Etudes des Télécommunications (CNET), em Lannion, França, onde trabalhou em medidas de qualidade objetivas e subjetivas para codificadores de voz. De dezembro de 1991 a setembro de 1993 ele foi Cientista Visitante no Centro Científico Rio da IBM Brasil, onde trabalhou no projeto de novos codificadores de imagem, com aplicação especial para imagens obtidas por satélites de sensoriamento remoto. O Dr. Alcaim foi o Technical Program Chairman dos simpósios internacionais SBT/IEEE International Telecommunications Symposium de 1990 e 1994, e o Executive Chairman da IEEE Global Telecommunications Conference de 1999. Ele foi membro do Conselho Deliberativo da SBrT no período de 1996 a 2001 e membro do Comitê de Assessoramento de Engenharia Elétrica, Biomédica e Microeletrônica (CA-EE) do CNPq no período de 1998 a 2001. O Dr. Alcaim é correspondente regional do IEEE Global Communications Newsletter e Editor da área de Processamento de Sinais da Revista da SBrT.