Desenvolvimento de um Sistema de Reconhecimento Automático de Voz Contínua com Grande Vocabulário para o Português Brasileiro

Ênio Silva, Luiz Baptista, Helane Fernandes e Aldebaro Klautau

[1]Laboratório de Processamento de Sinais – LaPS
Departamento de Engenharia Elétrica e de Computação
Universidade Federal do Pará – UFPA
Rua Augusto Correa, 1 – 660750-110 – Belém, PA, Brasil
http://www.laps.ufpa.br

{enio,kidbit,helane,aldebaro}@deec.ufpa.br

Abstract. *This work presents some steps towards the development of a large vocabulary continuous speech recognition system for Brazilian Portuguese, with a vocabulary of more than 60 thousand words. The paper discusses the creation of a phonetic dictionary, with pronunciation(s) of each word, the improvement of a N-gram language model through the adoption of a class-based scheme, and acoustic modeling using triphones clustered through decision trees. Some preliminary results for the Spoltech corpus are also presented.*

Resumo. *Este trabalho apresenta os avanços na implementação de um reconhecedor de voz para o português brasileiro. O sistema é apto a processar voz contínua e suporta um vocabulário de 60 mil palavras. Discute-se a geração de um dicionário fonético, contendo a(s) pronúncia(s) de cada palavra, o aperfeiçoamento do modelo N-grama da linguagem através da utilização de classes de palavras, e o treinamento do modelo acústico usando trifones agrupados através de árvores de decisão. São também apresentados resultados preliminares para o corpus Spoltech.*

1. Introdução

Não existe atualmente em domínio público um sistema de reconhecimento automático de voz para o português brasileiro (PB), com suporte a grandes vocabulários, ou seja, com mais de 30 mil palavras. Esses sistemas são identificados na literatura como *large vocabulary continuous speech recognition* (LVCSR), e estão disponibilizados para a língua inglesa, por exemplo, através do projeto Sphinx 4 [Sphinx, 2005].

No Brasil, existem diversos grupos de pesquisa atuando em reconhecimento de voz, mas a grande maioria dos trabalhos publicados restringe-se ao uso de palavras isoladas ou vocabulário reduzido [Ynoguti and Violaro, 1999, Pessoa et al., 1999b, Santos and Alcaim, 2002, Fagundes and Sanches, 2003]. As dificuldades para o desenvolvimento de sistemas LVCSR para o PB podem ser aglutinadas ao redor de duas lacunas: a de um corpus de voz digitalizada e transcrita, grande o suficiente para o treinamento de modelos acústicos, e a de recursos específicos ao PB, tal como um dicionário fonético. O presente trabalho concentra-se prioritariamente na descrição das ações visando ao desenvolvimento dos recursos necessários.

Espera-se com esse trabalho, disseminar os recursos junto à academia, facilitando as atividades dos que desejam atuar em reconhecimento de voz, mas que atualmente devem vencer diversas barreiras para a configuração de um sistema básico. A estratégia adotada assume que existem bons toolkits em domínio público para o desenvolvimento de sistemas LVCSR, tais como o HTK (linguagem C), Sphinx 4 (Java) e ISIP (C++). Dispõe-se também do Spoltech [SPOLTECH, 2001], um corpus para o PB de tamanho relativamente reduzido desenvolvido pela UFRGS e OGI (EUA), mas que serve como um ótimo ponto de partida por conter transcrições fonéticas úteis para o "bootstrap" dos modelos acústicos. Assim, a contribuição desse trabalho é o desenvolvimento de recursos que são específicos ao PB, com os quais torna-se possível testar e disponibilizar um framework para LVCSR em PB. Devido à atual inexistência de um corpus adequado para o treinamento dos modelos acústicos, os resultados são bem aquém dos encontrados para outras línguas, mas o framework reúne todos os ingredientes para o melhoramento dos vários estágios que compõem um sistema LVCSR.

Dentre os recursos aqui discutidos, encontra-se o modelo de linguagem N-grama, o dicionário fonético e tabelas para o mapeamento entre alfabetos fonéticos, as quais são essenciais uma vez que as fontes de informação adotam diferentes alfabetos. Por exemplo, a primeira versão do dicionário fonético utilizava um alfabeto diferente do utilizado nas transcrições do corpus Spoltech.

Este artigo encontra-se organizado da seguinte forma. Na Seção 2 faz-se uma breve revisão dos modelos de linguagem adotados em reconhecimento de voz e discute-se o desenvolvido para este trabalho. Na Seção 3 apresentam-se a criação de um dicionário manipulação de alfabetos fonéticos. A Seção 4 apresenta resultados preliminares para o PB usando o corpus Spoltech, e os recursos discutidos. Na Seção 5 são apresentadas as conclusões do trabalho.

2. Desenvolvimento do Modelo de Linguagem

A modelagem da linguagem é ingrediente essencial de muitos sistemas computacionais, tais como reconhecimento de voz. Geralmente, os sistemas de reconhecimento de voz (SRV) são baseados em cadeias escondidas de Markov (HMMs, de hidden Markov models) [Huang et al., 2001]. Esses sistemas convertem o sinal de voz digitalizado em uma matriz X de parâmetros, e buscam a seqüência de palavras W que maximiza a probabilidade condicional.

$$\hat{W} = \arg\max_{W} P(W|X)$$

Na prática, usa-se a regra de Bayes para implementar a busca através de:

$$\hat{W} = \arg\max_{W} \frac{P(X|W)P(W)}{P(X)} = \arg\max_{W} P(X|W)P(W)$$

com $P(X)$ sendo desprezado pois não depende de W. Para cada W, os valores de $P(X|W)$ e $P(W)$ são fornecidos pelos modelos acústico e de linguagem (ou língua [Pessoa et al., 1999a]), respectivamente.

Modelos estatísticos de linguagem fornecem a probabilidade de uma seqüência de palavras $w_0^l = w_0 \ldots w_l$, a qual também será representada por w_0^1 e chamada genericamente de sentença. Assume-se que w_0 é um símbolo para o início da sentença consistindo

de $l-1$ palavras, e w_l é um símbolo para o final da sentença. O modelo de linguagem (ML) mais utilizado para aplicações em reconhecimento de voz usa a aproximação *N*-grama , a qual assume que a distribuição de probabilidade para a palavra atual depende somente das $n-1$ palavras precedentes:

$$p(w_1^l|w_0) = \prod_{i=1}^{l} p(w_i|w_0^{i-1}) \approx \prod_{i=1}^{l} p(w_i|w_{i-n+1}^{i-1}).$$

Ressalta-se que a probabilidade para o símbolo final da sentença será avaliada no fim da sentença como se fosse uma outra palavra, enquanto que o começo da sentença é tratado apenas como uma informação do histórico (ou contexto).

Na criação do modelo de linguagem é desejável então encontrar estimativas ótimas para probabilidades condicionadas a cada contexto. A principal dificuldade em encontrar essas estimativas provém da esparsidade dos dados do treinamento. Uma vez que muitas palavras são nunca ou raramente observadas, suas estimativas não são confiáveis. Para um reconhecedor de voz, palavras que possuem probabilidade zero nunca serão reconhecidas nem que elas sejam acusticamente plausíveis. Isso é chamado de *problema da freqüência zero*. Existem muitas técnicas de suavização que buscam assegurar que todas as palavras, mesmo as que não apareçam no conjunto de treino, possuam probabilidade positiva.

Para melhor estabelecer os objetivos do presente trabalho, alguns conceitos importantes são descritos a seguir. Um texto T é uma coleção de sentenças e sua probabilidade $p(T)$ é o produto da probabilidade de sentenças individuais (assume-se independência estatística entre as sentenças). Para avaliar a qualidade de um modelo de linguagem em T, pode-se usar a entropia cruzada (também chamada *per-word coding length* ou *cross-entropy*),

$$H_p(T) \stackrel{\text{def}}{=} -\frac{1}{W_T} \log p(T).$$

onde W_T denota o número de palavras em T. Note que se uma probabilidade zero é atribuída a uma palavra que aparece no texto, $H_p(T)$ é infinita. A partir de $H_p(T)$, pode-se definir a perplexidade como

$$PP \stackrel{\text{def}}{=} 2^{H_p(T)}.$$

A perplexidade pode ser entendida como o número médio de diferentes (e equiprováveis) palavras que podem seguir uma dada palavra, de acordo com o modelo de linguagem adotado. Por exemplo, $PP = 10$ em um SRV para dez dígitos (0 a 9). Para SRV da língua inglesa, com vocabulários de tamanho superior a 20.000 palavras, PP costuma variar entre 100 e 250. Para uma dada tarefa de reconhecimento de voz, objetiva-se encontrar modelos de linguagem que conduzam a baixas perplexidades e custo computacional reduzido.

Considerando-se o SRV como um todo, a medida mais comum de avaliação é a taxa de palavras erradas (WER, de word error rate). Pode-se avaliar modelos de linguagem mantendo-se o modelo acústico fixo, e observando-se como as diferentes técnicas impactam a WER. Contudo, essa estratégia possui um custo computacional alto, sendo comum a utilização da perplexidade nos estágios iniciais do desenvolvimento de modelos de linguagem para SRV. Isso é justificado pela forte correlação entre WER e PP ou,

equivalentemente, $H_p(T)$, como indica a expressão[1]

$$WER \approx -12.37 + 6.48 log_2(PP) = -12.37 + 6.48 H_p(T).$$

2.1. Estimação dos Modelos de Linguagem N-grama

O modelo de linguagem precisa fornecer a probabilidade $P(W)$ sobre a seqüência de palavras W, tal que a mesma reflita a freqüência com que W ocorre. Usando-se o conceito de probabilidade condicional, a probabilidade conjunta das palavras $w_1, w_2, ..., w_n$ em W pode ser decomposta em:

$$P(W) = P(w_1, w_2, ..., w_n)$$

$$P(W) = P(w_i)P(w_1|w_2)P(w_3|w_1, w_2)...P(w_n|w_1, w_2, ..., w_{n-1})$$

$$P(W) = \prod_{i=1}^{n} P(w_1|w_2, ...w_{n-1})$$

Onde $P(w_i|w_2, ...w_{n-1})$ é a probabilidade que w_i terá, dada que a sequência de palavras $w_1, w_2, ..., w_{i-1}$ ocorreu previamente (note que se $i = 1$, obtém-se $P(w_1)$). Em $P(w_i|w_2, ...w_{n-1})$ temos que a escolha de w_i, depende da história passada da entrada. Para um vocabulário de tamanho v, há v^{i-1} histórias diferentes e então, para especificar $P(w_i|w_2, ...w_{n-1})$ completamente, v^i valores devem ser estimados. Na prática então, é inviável estimar as probabilidades $P(w_i|w_2, ...w_{n-1})$ mesmo para valores moderados de i. Uma solução prática para esse problema é assumir que $P(w_i|w_2, ...w_{n-1})$ depende somente de algumas classes equivalentes. As classes equivalentes podem ser simplesmente baseadas em diversas palavras prévias $w_{i-N+1}, w_{i-N+2}, ..., w_{i-1}$. Isto conduz ao Modelo de Linguagem N-grama. Se a palavra depende previamente de apenas duas palavras, nós temos um trigrama: $P(w_i|w_{i-2}, w_{i-1})$. Similarmente, nós podemos ter unigrama: $P(w_i)$, ou bigrama: $P(w_i|w_{i-1})$.

O modelo trigrama é particularmente poderoso, pois a maioria das palavras possui uma forte dependência das duas palavras anteriores, e isso pode ser estimado razoavelmente bem com corpus de porte considerável. No modelo de bigrama, nós fazemos a aproximação de que a probabilidade da palavra depende somente da identificação da palavra anterior. Para fazer $P(w_i|w_{i-1})$ significativo para i = 1, nós inserimos no inicio da seqüência, um token de distinção $< s >$; isto é, nós fazemos $w_0 =< s >$. Além disso, para fazer a soma da probabilidade de todas as strings igual a 1, é necessário inserir o token $< /s >$ para identificar o final da sentença. Para estimar $P(w_i|w_{i-1})$, a frequência com que a palavra w_i ocorre dado que a palavra anterior é w_{i-1}, nós simplesmente contamos com que frequência a sentença (w_{i-1}, w_i) ocorre em algum texto, e normalizamos o contador pelo número de vezes que w_{i-1} ocorreu.

[1]Obtida por W. Fisher a partir do estudo de diversos SRV, e divulgada em reuni˜ao organizada pelo NIST / EUA em 2000. Veja *http://www.isip.msstate.edu/publications/courses/ece_8463/*.

Em geral, para o modelo trigrama a probabilidade da palavra, depende das duas anteriores. O trigrama pode ser estimado pela observação das freqüência ou contadores do par $C(w_{i-2}, w_{i-1})$ e o trio $C(w_{i-2}, w_{i-1}, w_i)$ como se segue

$$P(w_i|w_{i-2}, w_{i-1}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-2}, w_{i-1})}$$

O texto disponível para a construção do modelo é chamado de corpus de treino. Para o modelo N-grama a quantidade de dados de treino é geralmente de milhões de palavras. A estimação da equação acima é baseada no princípio da máxima verossimilhança [Huang et al., 2001].

O uso de classes equivalentes consiste em reduzir o número de parâmetros a ser estimado, agrupando-se as palavras que desempenhem a mesma função em uma linguagem. Existem diferentes maneiras para agrupar palavras, uma delas usar informações semânticas e sintáticas que existem na linguagem. É quase sempre vantajoso agrupar palavras que possuam uma função semântica similar. Por exemplo, se fossemos construir um modelo de linguagem de um sistema conversacional para aeroportos, poderíamos agrupar os diferentes nomes das empresas aéreas como, GOL, TAM e VASP, em uma classe chamada "linhas aéreas". Podemos fazer o mesmo para diferentes nomes de aeroportos, cidades, e assim por diante. Para esse sistema notamos uma grande redução no espaço de busca, visto que tal sistema é limitado a um determinado assunto, no caso, reserva de passagens aéreas. O modelo n-grama pode ser generalizado pela definição da função de classes equivalentes que atuam no histórico $\varepsilon(.)$ das palavras:

$$P(w_i|w_1, w_2, ..., w_{i-1}) = P(w_i|\varepsilon(w_1, w_2, ..., w_{i-1}))$$

A definição de ε para um modelo de palavras n-grama esta representado na equação abaixo:

$$\varepsilon_{palavra-n-grama}(w_1, w_2, ..., w_i) = \varepsilon(w_{i-n+1}, ..., w_i)$$

Em um bom modelo de linguagem a escolha de ε deve ser tal que providencie uma predição confiável da próxima palavra.

Isto pode ser feito mapeando um conjunto de palavras para classes de palavras $g \in$ G usando a função de classificação $G(w) = g$. Se uma classe possuir mais que uma palavra então o resultado desse mapeamento irá resultar num menor numero de classes de palavras do que palavras G $<$ W, reduzindo assim os parâmetros à predizer. A classe equivalente pode ser descrita como a seqüência das classes:

$$\varepsilon_{classes-n-grama}(w_1, w_2, ..., w_i) = \varepsilon(G(w_{i-n+1}), ..., G(w_i))$$

2.2. Resultados do Modelo de Linguagem

Nesta seção apresentamos resultados de simulações. Utilizamos um corpus extraído majoritariamente de um jornal e formatado usando XML. O corpus tem aproximadamente

30 milhões de sentenças, das quais extraímos dois conjuntos disjuntos para treino e teste, após retirarmos os tags XML.

Na Figura 1 encontram-se os resultados de um experimento preliminar usando bigramas com dicionários de tamanhos variados, a Figura 2 apresenta curvas semelhantes a 1, porém utilizando modelos trigramas. Estas curvas foram estimadas através do método "default" do software HTK (*http://htk.eng.cam.ac.uk/*). Nota-se pelo gráfico que a perplexidade apresenta valores relativamente altos, exigindo um aperfeiçoamento das técnicas de treino do modelo.

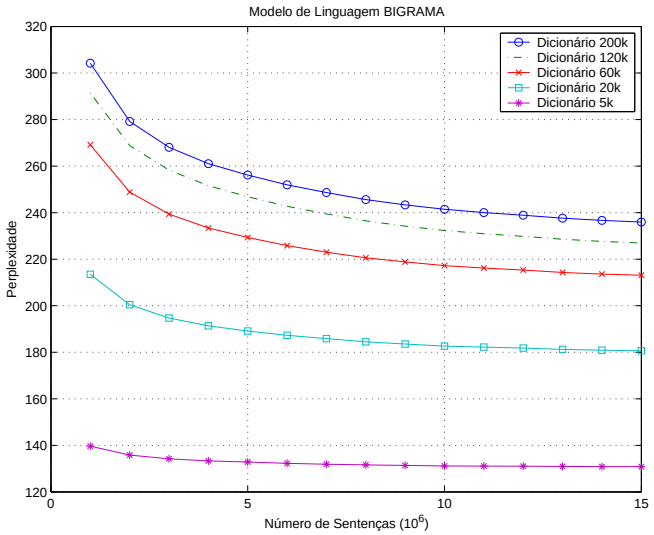

Figura 1. Evolução da perplexidade com o aumento dos dados para treino, para bigramas.

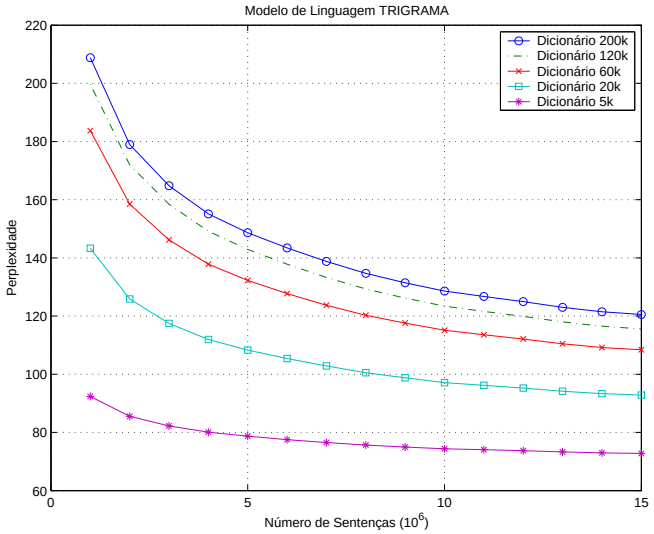

Figura 2. Evolução da perplexidade com o aumento dos dados para treino, para trigramas.

Na Figura 3 comparamos os diferentes modelos de linguagem e aplicamos a técnica de mapeamento, utilizando 300 classes, e o método de interpolação linear [Huang et al., 2001]. A Figura 4 mostra o resultado do modelo de trigrama mapeado

por classes em dicionários de tamanhos variados como em [Huang et al., 2001]. Resultados de simulações permitiram concluir que o mapeamento por classes de palavras conduz a uma melhoria do desempenho do sistema.

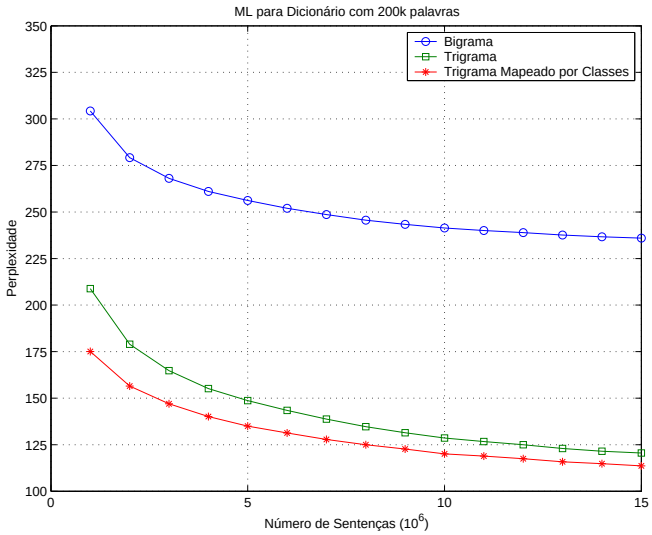

Figura 3. Comparação da evolução da perplexidade de diferentes modelos de linguagem com aumento dos dados para treino.

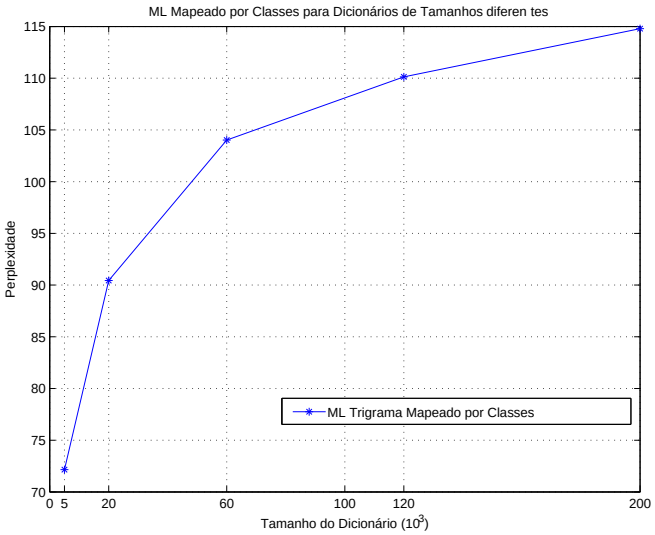

Figura 4. Evolução da perplexidade de um modelo trigrama mapeado por classes e utilizando método de interpolação linear com aumento do tamanho do vocabulário.

3. Alfabetos, Transcrições e Dicionário Fonético

Um sistema de LVCSR tipicamente adota um alfabeto fonético para representar os sons, ou mais especificamente, os fones modelados por HMMs em sistemas típicos. Para o atual projeto, optou-se pela adoção do alfabeto fonético SAMPA (Speech Assessment Methods Phonetic Alphabet). O SAMPA é uma notação derivada do IPA (International Phonetic Alphabet), a qual pode ser representada através do conjunto de caracteres ASCII,

disponíveis em um teclado normal. Nota-se que alguns dos símbolos do SAMPA foram adaptados para melhor representarem pronúncias no PB.

Tendo-se escolhido o alfabeto, pôde-se então partir para a construção do dicionário fonético, o qual consiste em uma a tabela que provê, para cada palavra do vocabulário, uma ou mais pronúncias descritas através dos símbolos do alfabeto fonético adotado pelo sistema.

Nesse trabalho, o dicionário vem sendo construído em três etapas. Na primeira, foram utilizados recursos de programação para extrair os verbetes de um dicionário eletrônico e assim obter uma pronúncia canônica para cada palavra. Posteriormente, com base em estatísticas de textos, as palavras com o menor número de ocorrência foram eliminadas. Por fim, pronúncias alternativas estão sendo geradas através de técnica semelhante à discutida em [Seara et al, 2003]. Como o dicionário eletrônico adota um alfabeto próprio, o mesmo teve que ser convertido para o SAMPA, em um processo semelhante ao descrito a seguir.

O Spoltech apresenta, para cada sentença (arquivo de voz digitalizada), uma transcrição ortográfica e, para um subconjunto dessas sentenças, também a transcrição fonética. Entretanto, o alfabeto fonético utilizado pelo Spoltech é o alfabeto do OGI [SPOLTECH, 2001]. Assim, foi cuidadosamente desenvolvido um mapeamento dos símbolos do alfabeto OGI para o SAMPA, uma vez que as transcrições são deveras importantes [Ynoguti and Violaro, 1999]. Parte desse mapa encontra-se descrito na Figura 5.

Parte do Mapeamento do Alfabeto OGI para SAMPA		
Fonemas do Português Brasileiro	**OGI (usado no Spoltech)**	**SAMPA**
A	Oral = a Nasal = ax~ Final de palavra = a	Oral = a Nasal = a~ Final de palavra = 6
E	Oral = fechado e, aberto E Nasal = e~ Átono final de palavra = i	Oral = fechado e, aberto E Nasal = e~ Átono final de palavra = i
Ei	Ej	eI
Em	ej~	e~I~
K	Kck	k
L	l e L (quando seguido de i como em litro)	l e L (quando seguido de i como em litro)
M	M	m
N	N	n
T	tct (como em *tala*), tctS (diante de i como em *tia*)	t (como em *tala*), tS (diante de i como em *tia*)

Figura 5. Parte do Mapeamento do Alfabeto OGI para SAMPA.

4. Resultados Preliminares

Para o treinamento dos modelos acústicos, foi utilizado parte do corpus Spoltech, correspondendo a aproximadamente 3500 sentenças. Dentre essas, 2000 foram usadas para o

treinamento e 1500 para teste.

Dispondo-se dos recursos para o PB e do corpus Spoltech, torna-se possível realizar simulações para validar o framework. Ressalta-se que os scripts usados e os próprios recursos podem ser obtidos para fins acadêmicos junto aos autores. Nesta seção apresentamos resultados deveras preliminares, mas que indicam a gama de pesquisas e melhoramentos que podem ser realizados a partir do framewok desenvolvido.

Para a obtenção de bons resultados com o modelo acústico, é essencial que se leve em conta o efeito da coarticulação dos sons. Para esse fim, o uso de trifones é uma técnica consagrada. Contudo, para uma estimativa mais robusta dos mesmos, é importante a utilização de agrupamento (ou *clustering*) [Huang et al., 2001]. Existem basicamente duas técnicas: a *data-driven* e a baseada em árvores de decisão. Para a utilização da segunda, a qual é a mais popular em sistemas modernos, torna-se necessário contar com uma lista de "perguntas" baseadas em classes fonéticas [HTK, 2005], a qual direciona o agrupamento de sons semelhantes. Essa lista foi outro recurso que foi adaptado para o PB e conduziu a uma melhoria dos resultados.

O treinamento do modelo de linguagem foi baseado nas técnicas já discutidas, em especial a da equivalência de classes. Foi utilizado um corpus extraído majoritariamente de um jornal e formatado usando XML. O corpus tem aproximadamente 30 milhões de sentenças. A pontuação foi substituída por tags especiais tais como <EXCLAMAÇÃO>, <VÍRGULA>, etc. Os dados foram separados em três conjuntos disjuntos para treino, validação e teste. Tanto o conjunto de validação quanto o de teste foram mantidos em 2500 sentenças.

Para o processo de extração de parâmetros (*front end*) foram utilizados os consagrados MFCCs (*Mel-frequency cepstrum coefficients*), com 12 parâmetros "estáticos", a energia e estimativas das duas primeiras derivadas, compondo um total de 39 parâmetros por quadro. A duração do quadro foi de 20 milisegundos (ms), com um deslocamento de 10 ms. Os modelos acústicos, para monofones e *word-internal* trifones, foram construídos baseados em HMMs, com diferente número de Gaussianas por estado. Para avaliarmos o desenpenho dos diferentes modelos observamos a taxa de erro por palavras ou WER (*word error rate*). A Tabela 1 mostra o comportamento dos modelos estimados baseados em monofones. Nota-se que a WER é bastante alta, mas isso é esperado perante o grande número de palavras do vocabulário, perplexidade relativamente alta do ML e uma quantidade de voz digitalizada relativamente pequena para o treinamento do modelo acústico. Ao evoluirmos os modelos HMM de monofones para trifones, obtemos melhoria no desempenho do sistema, por exemplo, com uma WER de 37.42% para 8 Gaussianas por estado.

Tabela 1. Desempenho de diferentes modelos monofones para o corpus Spoltech.

# Gaussianas	Taxa de erro por palavra - WER (%)
6	63.80
7	63.61
8	58.12

5. Conclusões

O presente trabalho abordou o desenvolvimento de uma série de recursos tendo em vista a criação de um sistema LVCSR para o português brasileiro. Dentre esses, discutiu-se o dicionário fonético, mapeamentos entre os alfabetos e, em especial, o modelo de linguagem.

Foram apresentados resultados preliminares para o corpus Spoltech. A taxa de erro por palavra foi relativamente alta, situando-se em torno de 37%. Contudo, o objetivo maior do teste foi a validação dos recursos desenvolvidos. É incontestável que existe muito trabalho a ser feito no aperfeiçoamento dos diversos módulos que compõem o reconhecedor. É opinião dos autores de que, apenas o trabalho conjunto de diversos grupos pode diminuir o *gap* que separa os sistemas para o PB de outros para línguas como a inglesa. Com este intuito, tanto os recursos quanto os scripts para treinamento do sistema estão sendo disponibilizados para fins acadêmicos.

Além do aperfeiçoamento já citado, o prosseguimento dessa pesquisa inclui a criação de um corpus de voz digitalizada de tamanho adequado. Para isso, gravações de um programa de televisão estão sendo segmentadas e transcritas a nível de palavra.

Referências

Fagundes, R. and Sanches, I. (2003). Uma nova abordagem fonético-fonológica em sistemas de reconhecimento de fala espontânea. *Revista da Sociedade Brasileira de Telecomunicações*, 95.

HTK (2005). http://htk.eng.ac.uk.

Huang, X., Acero, A., and Hon, H.-W. (2001). *Spoken language processing*. Prentice-Hall.

Pessoa, L., Violaro, F., and Barbosa, P. (1999a). Modelo de língua baseado em gramática gerativa aplicado ao reconhecimento de fala contínua. In *XVII Simpósio Brasileiro de Telecomunicações*, pages 455–458.

Pessoa, L., Violaro, F., and Barbosa, P. (1999b). Modelos da língua baseados em classes de palavras para sistema de reconhecimento de fala contínua. *Revista da Sociedade Brasileira de Telecomunicações*, 14(2):75–84.

Santos, S. and Alcaim, A. (2002). Um sistema de reconhecimento de voz contínua dependente da tarefa em língua portuguesa. *Revista da Sociedade Brasileira de Telecomunicações*, 17(2):135–147.

Seara et al, I. (2003). Geração automática de variantes de léxicos do português brasileiro para sistemas de reconhecimento de fala. In *XX Simpósio Brasileiro de Telecomunicações*, pages v.1. p.1–6.

Sphinx (2005). http://cmusphinx.sourceforge.net/sphinx4/.

SPOLTECH (2001). Advancing human language technology in brazil and the united states through collaborative research on portuguese spoken language systems.

Ynoguti, C. A. and Violaro, F. (1999). Influência da transcrição fonética no desempenho de sistemas de reconhecimento de fala contínua. In *XVII Simpósio Brasileiro de Telecomunicações*, pages 449–454.