



**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CURITIBA**

GERÊNCIA DE PESQUISA E PÓS-GRADUAÇÃO

**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA
ELÉTRICA E INFORMÁTICA INDUSTRIAL - CPGEI**

MONICA JORDAN

**PREDIÇÃO DE PALAVRAS BASEADA EM
MODELOS OCULTOS DE MARKOV**

DISSERTAÇÃO DE MESTRADO

**CURITIBA
SETEMBRO DE 2007**

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial

DISSERTAÇÃO
apresentada à UTFPR
para obtenção do Grau de

MESTRE EM CIÊNCIAS

por

MONICA JORDAN

**PREDIÇÃO DE PALAVRAS BASEADA EM
MODELOS OCULTOS DE MARKOV**

Banca Examinadora:

Presidente e Orientador:

PROF. DR. PERCY NOHAMA

UTFPR

Co-orientador:

PROF. DR. ALCEU DE SOUZA BRITTO JR

PUCPR

Examinadores:

PROF^a DR^a SONIA ANA CHARCHUT LESZCZYNSKI

UTFPR

PROF^a DR^a DILMEIRE SANT'ANNA RAMOS VOSGERAU

PUCPR

Curitiba, Setembro, 2007.

MONICA JORDAN

**PREDIÇÃO DE PALAVRAS BASEADA EM
MODELOS OCULTOS DE MARKOV**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do grau de “Mestre em Ciências” – Área de Concentração: Engenharia Biomédica.

Orientador: Prof. Dr. Percy Nohama

Co-orientador: Prof. Dr. Alceu de Souza Britto Jr.

Curitiba

2007

Ficha catalográfica elaborada pela Biblioteca da UTFPR – Campus Curitiba

J82p Jordan, Monica
Predição de palavras baseada em modelos ocultos de markov / Monica Jordan. Curitiba, UTFPR, 2008
XX, 168 p. : il. ; 30 cm

Orientador: Prof. Dr. Percy Nohama
Dissertação: (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial. Curitiba, 2008
Bibliografia: p. 159 – 168

1. Sistemas de informação. 2. Informática na educação. 3. Software. I. Nohama, Percy, orient. II. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial. III. Título

CDD: 621.3

AGRADECIMENTOS

A Deus por estar viva e ter encontrado em minha missão pessoas como o Professor Percy. Ao Professor Percy, pela personalidade e garra, e por ter me apresentado ao Professor Álvaro. Ao Professor Álvaro, pela receptividade, oportunidade e acompanhamento dedicado e ímpar.

Às escolas envolvidas na pesquisa, pelo espaço e confiança.

Aos voluntários, pela alegria e dedicação.

Aos meus pais, por tornarem possível minha vida.

Aos meus filhos, Rodolfo e Bernardo, por tornarem qualquer peso do caminho uma pena.

Ao meu irmão, pelo exemplo.

Ao Professor Alceu, pela figura fantástica que é.

A todos os pesquisadores da Terra, por acreditarem que tudo pode ser melhorado.

À Jesus por estar sempre ao meu lado.

E, enfim, ao meu companheiro Paulo César, por ser a melhor pessoa do mundo.

Ainda que eu fale as línguas dos homens e dos anjos,
se não tiver amor, serei como o bronze que
soa ou como o címbalo que retine.

Ainda que eu tenha o dom de profetizar e conheça
todos os mistérios e toda a ciência;
ainda que eu tenha tamanha fé,
a ponto de transportar montes,
se não tiver amor nada serei.

(Coríntios I, capítulo 13)

SUMÁRIO

LISTA DE FIGURAS.....	XIII
LISTA DE TABELAS.....	XV
LISTA DE ABREVIATURAS E SIGLAS.....	XVII
RESUMO.....	XIX
ABSTRACT.....	XX
CAPÍTULO 1 INTRODUÇÃO.....	1
1.1 MOTIVAÇÕES.....	1
1.2 OBJETIVOS.....	4
1.3 ESTRUTURA DA DISSERTAÇÃO.....	5
CAPÍTULO 2 FUNDAMENTAÇÃO TEÓRICA.....	7
2.1 ESTADO DA ARTE.....	7
2.1.1 As Deficiências no Brasil.....	7
2.1.2 Deficiência Verbal.....	8
2.1.3 Comunicação Alternativa e Aumentativa.....	9
2.1.4 Processamento de Linguagem Natural.....	11
2.1.5 Processamento de <i>corpus</i>	12
2.1.6 Predição de Palavras.....	13
2.1.7 Amplisoft.....	14
2.2 TÓPICOS TEÓRICOS.....	16
2.2.1 Acessibilidade.....	17
2.2.2 Inclusão Digital e Inclusão Social.....	19
2.2.3 Comunicação Natural, Deficiências Verbal e Relacional.....	21
2.2.4 Uso dos Computadores na Terapia da Deficiência Verbal.....	24
2.2.5 Comunicação Alternativa e Aumentativa (CAA).....	26
2.2.6 Desenvolvimento e Processamento de <i>Corpus</i>	29

2.2.7 Modelos Ocultos de Markov	31
CAPÍTULO 3 METODOLOGIA	41
3.1 MATERIAIS E INFRA-ESTRUTURA	41
3.2 DESENVOLVIMENTO	41
3.2.1 Desenvolvimento do <i>Corpus</i>	41
3.2.2 Determinação dos Parâmetros Markovianos	47
3.2.3 Algoritmo de Predição	53
3.2.4 Protocolo de Testes	59
3.2.5 Análise dos Resultados	61
3.2.6 Locais de Realização dos Testes de Aplicabilidade	61
3.2.7 Planos de Recrutamento dos Testes de Aplicabilidade	62
CAPÍTULO 4 RESULTADOS.....	63
4.1 CORPUS.....	63
4.2 ALGORITMO DE PREDIÇÃO	63
4.3 TESTES	64
CAPÍTULO 5 DISCUSSÃO.....	75
5.1 ANÁLISE DOS RESULTADOS.....	75
5.1.1 <i>Corpus</i>	75
5.1.2 Teclado Virtual Livre	76
5.1.3 Modelos Ocultos de Markov	77
5.1.4 Algoritmo de Predição.....	78
5.1.5 Testes Iniciais de Funcionamento do Teclado Virtual Livre com a Nova Técnica de Predição.....	79
5.1.6 Teste Individual.....	79
5.1.7 Testes para Obtenção de Parâmetros Passíveis de Comparação a Outra Técnica de Predição.....	80
5.1.8 Testes de Aplicabilidade	81
5.1.9 Respostas Dadas ao Questionário.....	85
5.2 TRABALHOS FUTUROS.....	86
CAPÍTULO 6 CONCLUSÕES	89
APÊNDICE I CABEÇALHO DOS TEXTOS QUE COMPÕEM O <i>CORPUS</i>.....	91

APÊNDICE II ROTINAS DESENVOLVIDAS PARA A TÉCNICA DE PREDIÇÃO	109
APÊNDICE III QUESTIONÁRIO ENTREGUE ÀS CRIANÇAS VOLUNTARIAS	
APÓS A DIGITAÇÃO DO TEXTO “A LAGARTA”	119
APÊNDICE IV CÓDIGO FONTE DO ALGORITMO DE PREDIÇÃO	121
ANEXO I TEXTOS DE TESTE	153
ANEXO II CARTA DE APROVAÇÃO DO COMITÊ DE ÉTICA	157
REFERÊNCIAS BIBLIOGRÁFICAS	159

LISTA DE FIGURAS

Figura 1: Taxa de atividade de pessoas deficientes ou não, do sexo feminino e masculino, segundo os grupos de idade, conforme censo de 2000 (extraída de CENSO, 2000).....	8
Figura 2: Fluxograma da predição de palavras do Teclado Virtual Livre.....	15
Figura 3: (a) Ponteiro de cabeça, (b) Linha Braille (extraídas de http://www.acessobrasil.org.br/).....	18
Figura 4: Exemplo de combinação de símbolos no Sistema Bliss de Comunicação, no qual a soma dos símbolos casa e Deus indicam a “casa de Deus” ou igreja (Adaptada de VASCONCELLOS, 1999).....	27
Figura 5: Exemplo de ícones PCS (extraída de http://www.clik.com.br/).....	27
Figura 6: Uma cadeia de Markov com 4 estados, representados por S_1, \dots, S_4 com suas possíveis transições e probabilidades inerentes, representadas por $a_{11}, a_{12}, \dots, a_{44}$ (Adaptada de RABINER, 1989).....	32
Figura 7: Modelo Ergótico, onde S_1, S_2 e S_3 são os estados do modelo (adaptada de BRITTO <i>et al.</i> , 2001).....	38
Figura 8: Modelo Sequencial, onde S_1, S_2 e S_3 são os estados do modelo (adaptada de BRITTO <i>et al.</i> , 2001).....	38
Figura 9: Modelo Paralelo, onde S_1, S_2, S_3 e S_4 são os estados do modelo (adaptada de BRITTO <i>et al.</i> , 2001).....	39
Figura 10: Modelo Oculto de Markov discreto e histograma de cores (adaptada de SINGH, 2006).....	39
Figura 11: Modelo Oculto de Markov contínuo e distribuição da probabilidade de ocorrência das cores (adaptada de SINGH, 2006).....	40
Figura 12: Texto Etiquetado (onde a seta amarela indica o símbolo indicativo do início da classe gramatical).....	49
Figura 13: Estrutura da Matriz B1.....	51
Figura 14: Indicação da Linha de Digitação do Aplicativo Teclado Virtual Livre por uma Seta.....	54
Figura 15: Estrutura da matriz de predição que armazena os dados necessários no algoritmo.....	55
Figura 16: Estrutura de um elemento “mtzPred”.....	56

Figura 17: Fluxograma do Algoritmo de Predição ao Comando de Avançar	58
Figura 18: Interface do Teclado Virtual Livre ajustada para ser testado por uma criança de 10 anos.	65
Figura 19: Valores obtidos na digitação do texto “A Lagarta” pela criança de 10 anos.....	66
Figura 20: Gráfico Comparativo dos Tempos de Execução Gastos pelos Voluntários 1, 2, 4, 6, 8, 9, 10 e 11 da 4ª A para Digitação do Texto “A Lagarta” com e sem a Predição de Palavras.	68
Figura 21: Gráfico Comparativo dos Números de Cliques Realizados pelos Voluntários da 4ªA da Escola Municipal Professora Genoveva Sicuro de Brito na Realização dos Testes.....	69
Figura 22: Gráfico Comparativo dos Tempos de Execução Despendidos pelos Voluntários da 4ª B para Digitação do Texto “A Lagarta” com e sem a Predição de Palavras, Desconsiderando-se os Dados do Voluntário 1	70
Figura 23: Gráfico Plotado sobre os Valores Obtidos na Digitação do Texto do Protocolo de Testes pelas Crianças Voluntárias da 4ª B	70
Figura 24: Gráfico Comparativo dos Tempos de Execução Despendidos pelos Voluntários da 4ª C para Digitação do Texto Referente Com e Sem a Predição de Palavras, Desconsiderando-se os Dados do Voluntário 9	71
Figura 25: Gráfico Comparativo Plotado sobre os Valores Obtidos na Digitação do Texto do Protocolo de Testes pelos Alunos Voluntários da 4ª C	72
Figura 26: Resultados Obtidos para o Parâmetro Tempo de Execução pelos Alunos da Escola Tia Vivian Marçal	73
Figura 27: Número de Cliques Realizados pelos Voluntários da Escola Tia Vivian Marçal para a Digitação do Texto de Teste	73

LISTA DE TABELAS

Tabela 1: Significados das siglas indicadas pelo etiquetador VISL.....	46
Tabela 2: Resultados obtidos no teste de coerência na busca de palavras.....	65
Tabela 3: Resultados obtidos na digitação do texto “A Lagarta” pela criança de 10 anos.	66
Tabela 4: Respostas dadas pela criança de 10 anos que testou o aplicativo.....	66
Tabela 5: Resultados obtidos na digitação de texto “O Menino e a Moeda” no teste realizado para comparação a dados obtidos em literatura referente.....	67
Tabela 6: Resultados dos testes realizados com as crianças voluntárias da 4ª série, turma A, obtidos com a digitação do texto sem e com o uso da predição.....	68
Tabela 7: Resultados dos testes realizados com as crianças voluntárias da 4ª série, turma B, obtidos com a digitação do texto sem e com o uso da predição.....	69
Tabela 8: Resultados dos testes realizados com as crianças voluntárias da 4ª série, turma C, obtidos com a digitação do texto sem e com o uso da predição.....	71
Tabela 9: Resultados dos testes realizados com os voluntários da Escola Tia Vivian Marçal,.....	72
Tabela 10: Respostas Obtidas na Apresentação do Questionário do Anexo I às Crianças Voluntárias da Escola Municipal Professora Genoveva Sicuro de Brito	74

LISTA DE ABREVIATURAS E SIGLAS

- AAC - *Augmentative Alternative Communication.*
- CAA - *Comunicação Alternativa e Aumentativa.*
- CNV - *Comunicação Não Verbal.*
- HMM - *Hidden Markov Models.*
- IBGE - *Instituto Brasileiro de Geografia e Estatística.*
- IBM - *International Business Machines.*
- ISAAC - *International Society Augmentative Alternative Communication.*
- PCS - *Picture Communication System.*
- PDA - *Personal Digital Assistants.*
- PIC - *Picture Ideogram Communication System.*
- PLN - *Processamento Automático da Linguagem Natural.*
- TIC - *Tecnologias da Informação e Comunicação.*
- VOCAs - *Voice Output Communication Aids.*
- www - *World Wide Web.*

RESUMO

A busca pela inclusão social tem promovido a pesquisa de ferramentas que ampliam o uso do computador por pessoas com necessidades especiais. Dentro da gama de ferramentas auxiliares, um importante acelerador de uso é a predição de palavras, que diminui o número de ações a ser executado e, conseqüentemente, tempo e esforços envolvidos. A antecipação de uma palavra dentro de um texto digitado é denominada predição de palavras, e pode ser desenvolvida a partir de considerações estatísticas como contagem de ocorrências, suas antecessoras e sucessoras. Um processo de análise estocástica muito utilizado na predição de acontecimentos, como reconhecimento da fala, envolve os Modelos Ocultos de Markov. A literatura aponta tal processo como um dos mais indicados para a predição de palavras, quando baseado num conjunto de textos etiquetado, de forma a apresentar a classe gramatical de cada uma de suas palavras, chamado *corpus*. Então, uniu-se a eficiência de um aplicativo desenvolvido para usuários com necessidades especiais à ferramenta de predição baseada em Modelos Ocultos de Markov. Desenvolveu-se um algoritmo, que foi associado a um simulador de teclado livre. O aplicativo final oferece ao usuário uma lista com as dez palavras próximas mais prováveis de ocorrerem, considerando a classe gramatical mais provável, e, dentro dos grupos de classes, as palavras de maior ocorrência. O algoritmo baseia-se num *corpus* elaborado com textos infantis criteriosamente selecionados. Realizaram-se testes de digitação de textos do *corpus* a fim de verificar a eficiência e coerência do algoritmo; testes contendo preparo de textos não presentes no *corpus*, observando a capacidade de predição e consistência textual do aplicativo; e, enfim, testes com dois grupos de voluntários, sendo um deles de pessoas com necessidades especiais, para verificar a funcionalidade e aplicabilidade. Os resultados mostram que o aplicativo diminui consideravelmente a dificuldade na produção textual, e oferece ao usuário palavras gramaticalmente corretas. Os Modelos Ocultos de Markov tornaram o processo de predição rápido e coerente. Assim, aliando o desenvolvimento de software à Engenharia de Reabilitação, este trabalho oferece uma técnica facilitadora para a produção de textos, colaborando para a inclusão digital e, conseqüentemente, social, de pessoas, muitas vezes, apontadas como incapazes de viverem independentemente.

Palavras-chave: software, educação inclusiva, informática educativa, produção de textos pelo aluno, apoio à pessoa com deficiência, modelos ocultos de Markov.

HIDDEN MARKOV MODELS-BASED PREDICTION OF WORDS

ABSTRACT

Social inclusion search has been promoting the technology development extending people with special needs computer's use. Amongst the innumerable auxiliary tools, words prediction is an important accelerator that diminishes the number of actions to be executed and, consequently, time and effort to do it. Word anticipation in a text being typed, called word prediction, can be developed from statistical considerations such as occurrences counting, its predecessors and successors. One of the most used techniques for stochastic analysis is the Hidden Markov Models, being the most indicated to word prediction in a collection of writings or recorded remarks used for linguistic analysis sets, called *corpus*. In such a way, this work added developed software for users with special necessities to the prediction tool based on Hidden Markov Models efficacy. The developed word prediction algorithm was associated to a freeware keyboard simulator, aiming to offer facilities and consequently better performance to the users. The resulting program offers a list containing the next ten most probable words, considering the grammatical class and frequently word inside classes groups. The algorithm is based on a *corpus* of infantile texts carefully chosen. It was developed in Visual Microsoft C++ 6.0[®] based on Viterbi's Algorithm and presents an innovative form to deal with its Matrix Occurrences. Digitizing tests of *corpus* texts were carried out in order to verify the algorithm uses; tests involving preparation of texts out the *corpus*, for testing the capacity of prediction and literal consistency; and tests with two groups of volunteers, one of them consisted of children with special needs, in order to verify its functionality and applicability. The prototype presented excellent performance, showing the facilities promoted by developed prediction technique uses. The innovative analysis and the Hidden Markov Models application had become the prediction process fast and accurate. Thus, the prototype reduces the effort in text production, collaborating for digital inclusion e, consequently, social inclusion of people that many times are pointed as disabled.

Keywords: Inclusive Education, Computer Education, Students Text Production, Disabled People Support, Hidden Markov Models.

CAPÍTULO 1

INTRODUÇÃO

O mundo atual clama pelo direito de todos ao saber e, também, busca formar uma geração que atenda às demandas de uma sociedade de conhecimento, delineada pelas diferenças (LODI, 2003). O acesso à informação parece estar ao alcance de todos, através da televisão, rádio, publicações impressas, bibliotecas e, cada vez mais presente, através da Internet.

Todo aquele que lê ou ouve pode aprender e quem se comunica pode trocar idéias com a sociedade em que vive. Mas, e quem não se comunica ou possui alguma dificuldade para comunicar-se?

A comunicação natural une os seres humanos e perpetua os conhecimentos (GASPERIN & LIMA, 2001), sendo a dimensão fundamental de toda estrutura humana psíquica e biológica (TUPY & PRAVETTONI, 1999). Nesta comunicação, a linguagem apresenta-se como principal elemento e, na inexistência de dificuldades, ela exige do homem um esforço extremamente reduzido (MORETTI, 1999).

A linguagem ainda atende outra necessidade humana essencial: a relacional. No entanto, quando o sistema responsável pela linguagem humana apresentar-se patologicamente atingido e seu processo é rompido ou modificado. Resta, então, estudar tais sistemas e processos e buscar alternativas para recuperá-los ou ampliá-los (MORETTI, 1999).

1.1 MOTIVAÇÕES

Quando uma patologia atinge a linguagem natural do ser humano, sua comunicação fica extremamente prejudicada e este ser, naturalmente social, acaba isolado e incompreendido. Este problema é chamado de deficiência verbal (GAVA, 1999).

Estima-se que uma em cada duzentas pessoas no mundo apresenta deficiência verbal devido a dificuldades cognitivas, motoras, neurológicas ou emocionais (CAPOVILLA, 1994).

A afasia, alteração de linguagem adquirida por uma lesão cerebral focal, por vezes apresenta-se isoladamente, mas pode estar associada a outras manifestações neurológicas ou

neuropsicológicas, como transtornos motores (hemiparesia¹ ou hemiplegia) que atinge 80% dos afásicos não-fluentes² e 20% dos afásicos fluentes³, transtornos sensoriais, visuais, demência e alterações de comportamento (PEÑA-CASANOVA & PAMIES, 2005).

Os transtornos motores, quando atingem os membros superiores, limitam a segunda forma de comunicação mais utilizada pelo homem: a escrita. A escrita, em sua história, vem proporcionando ao homem valores visuais lineares e uma consciência fragmentada, complementando a convivência auditiva e tornando a comunicação multivariada (composta de cores, imagens, sons e textos) (BARRETO, 1998).

De alguma forma, o conhecimento necessita ser publicado para sua validação e socialização, formando um ciclo em que sendo conhecido por outrem gera novas idéias que se tornam novas informações, e a escrita, desde a sua descoberta, torna possível e rápido este processo. Hoje, vive-se a era da comunicação eletrônica, que tendo por base o computador e a Internet, tem modificado o fluxo de informações e conhecimentos, tornando acessível toda informação àquele com acesso à Internet. Esta comunicação é, adicionalmente, ampliada pelas mensagens eletrônicas e aplicativos de conversação, como *Messenger*[®] e *ICQ*[®] (BARRETO, 1998). A importância da comunicação eletrônica é, dia-a-dia, ampliada pela invasão dos computadores nos lares, escolas e ambientes de trabalho (PINHEIRO, 2006).

Por isso, muitos projetos de pesquisa e programas governamentais, no Brasil e no mundo, objetivam a inclusão digital, buscando melhorias na cultura e qualidade de vida da população mundial (ABDALLA *et al.*, 2005). Para países como o Brasil, a inclusão digital promove fortemente a inclusão social, pois especializa a capacitação da população para o mercado de trabalho e, também, possibilita que uma notável camada da população passe a participar da política, decisões sociais e cultura (ABDALLA *et al.*, 2005).

O acesso à informática facilita a vida das pessoas por permitir que ações como movimentação bancária, compras, pesquisas acadêmicas, diversão (música, jogos e conversação) sejam feitas sem que seja preciso ausentar-se do ambiente domiciliar. Estudar e trabalhar também ficaram mais simples, pois grandes textos, tabelas e relatórios antigamente feitos manualmente, hoje são digitados, impressos e armazenados para posterior aplicação (PINHEIRO, 2006).

¹ Paralisia incompleta de nervo ou músculo de um dos lados do corpo que não perdeu inteiramente a sensibilidade e o movimento, ou seja, hemiplegia pouco acentuada (Bueno, 2000).

² Afasia não-fluente: caracteriza-se pela reduzida expressão, esforço e alterações articulatorias, vocabulário reduzido e frases pequenas. O afásico apresenta produção oral lenta e dificultosa, normalmente silábica (Peña-Casanova & Pamies, 2005).

³ Afasia fluente: transtorno de compreensão, sendo articulação e fluência normais. A expressão verbal pode apresentar um aumento, sendo inclusive excessivo (Peña-Casanova & Pamies, 2005).

Tais vantagens podem ser estendidas às pessoas com necessidades especiais, bastando que soluções para acesso, controle e aceleradores de uso do computador, diminuam suas dificuldades (JORDAN, MATIAS & NOHAMA, 2006). Atualmente, podem ser encontradas várias ferramentas de apoio, que visam ampliar a comunicação ou a acessibilidade de pessoas com alguma deficiência; porém, é comum tais ferramentas necessitem de ajuste ao seu usuário (o que nem sempre é possível, pois, além de falta de apoio técnico, não raramente é preciso ter acesso aos dados de projeto, como no caso de *softwares*), não serem encontradas facilmente no comércio de países em desenvolvimento (como produtos importados ou desenvolvidos dentro das faculdades) ou, quando encontrados, são muito caros (PESCINA, 2004).

No entanto, sob vários aspectos, a sociedade não se encontra pronta para receber e colocar a pessoa com deficiência na vida cotidiana, desde calçadas não preparadas para cadeirantes até sistemas de transporte sem fonte sonora para deficientes visuais (PINHEIRO, 2006), o que, agregado ao preconceito e à superproteção que a família geralmente exerce sobre a pessoa com deficiência, acaba por isolá-la em sua própria casa, abrindo mão de uma vida socialmente ativa. Desta forma, o computador e a Internet aparecem como grandes aliados para melhorar a qualidade de vida do indivíduo com necessidades especiais, permitindo que mesmo em sua casa ele possa estabelecer uma rede de contatos para troca de idéias e notícias, além de saber o que acontece no mundo, divertir-se e poder desenvolver uma atividade remunerada (PINHEIRO, 2006).

Este contato virtual pode motivar o fim do isolamento, pois sítios dedicados às pessoas com deficiência narram histórias de pessoas que venceram o preconceito e as dificuldades e melhoraram suas qualidades de vida e auto-estima⁴.

Mostra-se essencial, neste ponto, o desenvolvimento e entrega aos usuários finais de dispositivos que tornem a acessibilidade destas ferramentas (computador e Internet) real, que sejam gratuitos e com dados de projeto disponíveis aos interessados. Assim, tendo por fim ampliar a comunicação daquele que tem dificuldade em se comunicar naturalmente e colaborar com a inclusão digital e, conseqüentemente, social da pessoa com deficiência física com transtornos motores em membros superiores, desenvolveu-se o trabalho descrito nesta dissertação. Tal pesquisa encontra suas motivações nos fatos descritos e na assertiva de que todo ser humano, com deficiência ou não, tem o máximo direito possível de autonomia, dignidade, liberdade e equilíbrio em relação a sua condição física (MORETTI, 1999).

⁴ Como nos sítios: <http://www.bengalalegal.com/>, <http://www.deficienteeficiente.com.br/>, <http://www.defnet.org.br/>.

O projeto desenvolvido contribui cientificamente com o preparo de um *corpus* dedicado (SARDINHA, 2004), de língua portuguesa e vocabulário infantil, elaborado com textos individualmente selecionados e contendo cerca de 40.000 palavras. Também, apresenta um algoritmo de predição de palavras baseado nos modelos de Viterbi com análise das classes gramaticais das palavras, técnica utilizada com sucesso para outros idiomas, aumentando os acertos de predição (NAKAMURA *et al.*, 1990, KUHN, NIEMANN & SCHUKAT-TALAMAZZINI, 1994, CLARKSON, 1999, GUSTAVII & PETERSSON, 2003, ZIMMERMANN & BUNKE, 2004, LAW & CHAN, 2006, HUNNICUTT, NOZADZE & CHIKOIDZE, 2006).

O uso de um simulador de teclado que já possui predição e tem obtido êxito com a predição das palavras por método estatístico simples (MATIAS & NOHAMA, 2003), tornou um desafio desenvolver uma técnica para a predição que supere tal método e seja inovador para a língua portuguesa. Baseado no sucesso de técnicas semelhantes desenvolvidas para outros idiomas, como para o inglês (KUHN, NIEMANN & SCHUKAT-TALAMAZZINI, 1994, ZIMMERMANN & BUNKE, 2004, LAW & CHAN, 2006, CLARKSON, 1999 e NAKAMURA *et al.*, 1990), sueco (GUSTAVII & PETERSSON, 2003), e russo (HUNNICUTT, NOZADZE & CHIKOIDZE, 2006), as quais utilizam os Modelos Ocultos de Markov como base para a predição das palavras, optou-se por adotar tais postulados no desenvolvimento desta dissertação. Os Modelos Ocultos de Markov foram escolhidos, também por tratar os parâmetros envolvidos como seqüências de acontecimentos e permitirem a análise destes acontecimentos em processo e retrocesso, além de serem indicados pela literatura como a técnica principal no processamento estatístico da linguagem natural (GASPERIN & LIMA, 2001, ROSENFELD, 2000).

1.2 OBJETIVOS

A deficiência verbal pode atrapalhar o desenvolvimento de uma pessoa, pois a dificuldade em expressar suas intenções comunicativas pode camuflar suas reais capacidades cognitivas; logo, atribuem-se tarefas muito mais simples do que ela poderia desempenhar, interrompendo seu desenvolvimento (STELLA, 1999). Caso semelhante pode acontecer com uma pessoa que tem o movimento ou controle de seus membros superiores comprometidos, pois sua dificuldade em escrever pode tirá-la da sala de aula e do mercado de trabalho.

Com base nisso, o principal objetivo desta dissertação é desenvolver uma técnica de predição de palavras que acelere a escrita e diminua a dificuldade de crianças usuárias de aplicativos de acessibilidade textuais, como simuladores de teclado, através de processamento computacional rápido e inteligência na tomada de decisões, realizadas com base nos Modelos Ocultos de Markov. O algoritmo de predição desenvolvido será disponibilizado pela Internet com código fonte aberto.

A fim de se atingir a premissa maior desta pesquisa, os seguintes objetivos secundários devem ser realizados:

1. elaborar um *corpus* com textos infantis contendo um número de palavras suficiente para promover a predição de palavras pelo público alvo do trabalho;
2. realizar o processamento do *corpus* elaborado, envolvendo: etiquetagem, produção de cabeçalhos individuais, organização por classes gramaticais das palavras que o compõe, eliminação de espaços em branco, análise da pontuação de finalização de sentenças e contagem das palavras;
3. elaborar lista de pares de palavras x (*versus*) classe gramatical do *corpus*;
4. determinar os parâmetros do Modelo Oculto de Markov, contendo as matrizes A (probabilidade de transição), B (probabilidade de observação) e π (probabilidade de cada estado ser o inicial);
5. desenvolver o algoritmo de predição, com base nos parâmetros do Modelo Oculto de Markov;
6. adaptar o algoritmo desenvolvido ao código fonte do Teclado Virtual Livre;
7. realizar testes com crianças alfabetizadas;
8. analisar estatisticamente os resultados.

1.3 ESTRUTURA DA DISSERTAÇÃO

Contendo cinco Capítulos, este trabalho apresenta a pesquisa na qual foi desenvolvido um algoritmo de predição de palavras com base em Modelos Ocultos de Markov.

No Capítulo 1, discorre-se sobre os fatos que contribuíram para a decisão do escopo da pesquisa, seus objetivos principal e secundários.

No Capítulo 2, encontram-se citações que serviram como base para o desenvolvimento realizado. Estas citações encontram-se divididas em dois itens: (1) revisão da literatura, para a qual foram extraídos textos de publicações que sustentam tecnicamente as decisões tomadas

no decorrer da pesquisa; e, (2) tópicos teóricos, no qual são discorridos os temas envolvidos no trabalho.

A Metodologia adotada para a concretização dos objetivos desta dissertação encontra-se exposta no Capítulo 3, com todos os detalhes e apontamentos necessários à compreensão do realizado.

O Capítulo 4 apresenta os resultados obtidos em todas as etapas do desenvolvimento da pesquisa, os quais são discutidos e comentados no Capítulo 5, que ainda apresenta as conclusões da pesquisa e as sugestões de trabalhos a serem realizados no futuro e que podem se basear nos estudos realizados e aqui apontados.

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

Este capítulo traz os apontamentos que ancoram a pesquisa realizada, percorrendo assuntos pertinentes e necessários à compreensão do trabalho desenvolvido, seus objetivos e justificativas. O item 2.1 mostra inicialmente o estado atual das pesquisas realizadas em áreas correlatas, percorrendo trabalhos que visam as inclusões digital e social, dispositivos alternativos de controle do computador para pessoas com necessidades especiais, comunicação alternativa e aumentativa e o uso dos Modelos Ocultos de Markov em técnicas de predição e reconhecimento de fala e escrita, os quais serviram de apoio nas tomadas de decisões.

Na seqüência, os tópicos teóricos focalizam os conhecimentos adquiridos nos estudos envolvidos no trabalho e resumem os temas citados. Explana-se sobre a linguagem humana, comunicação alternativa e aumentativa, aplicativos emuladores de teclado, predição de palavras, discute-se sobre *corpus* e seu processamento, e, enfim, apresenta-se a teoria envolvida nos Modelos de Markov.

2.1 ESTADO DA ARTE

O acesso às publicações recentes que tratam dos assuntos discorridos neste trabalho, permitiu que fosse estruturado o conhecimento necessário, estando tal estrutura baseada nos tópicos descritos a seguir.

2.1.1 As Deficiências no Brasil

Segundo o censo demográfico realizado em 2000 pelo Instituto Brasileiro de Geografia e Estatística (IBGE), 24.537.984 brasileiros apresentam algum tipo de necessidade especial, representando cerca de 15% da população do país (IBGE, 2000). O censo ilustra a taxa de atividade de pessoas com deficiência ou não dos sexos masculino e feminino, como mostra a Figura 1, na qual é possível observar que cerca de 20% de homens com deficiência em idade madura (25 a 50 anos) não exercem nenhuma atividade, o que é ainda mais saliente

para as mulheres, atingindo quase metade da população deficiente feminina brasileira. Os gráficos comparam a taxa da população ativa com pelo menos uma deficiência ou sem, sendo estas deficiências: mental, física, visual, auditiva e motora (IBGE, 2000).

Destaca-se que os termos deficiência e deficiente aplicam-se neste texto de acordo com as definições que o IBGE (IBGE, 2000) atribui a eles.

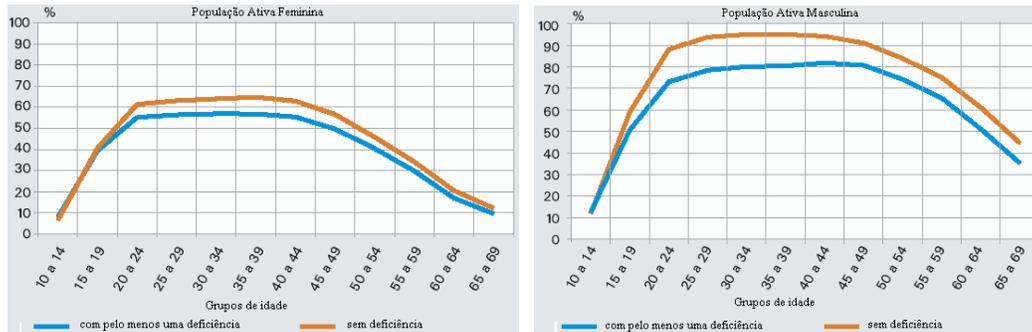


Figura 1: Taxa de atividade de pessoas deficientes ou não, do sexo feminino e masculino, segundo os grupos de idade, conforme censo de 2000 (extraída de IBGE, 2000).

O conjunto de pessoas com necessidades especiais, apontado pelo censo demográfico de 2000, apresenta a seguinte realidade: 11,56% das pessoas com alguma deficiência apresentam deficiência mental; 0,44% apresentam tetraplegia, paraplegia ou hemiplegia, para caminhar ou subir escadas; 22,70% apresentam alguma dificuldade; 7,21% relataram grande dificuldade e 2,30% disseram-se incapazes; 5,32% apresentam ausência de algum membro ou de parte dele; quanto à visão, 57,16% relataram alguma dificuldade para enxergar; 9,90% grande dificuldade e 0,60% são incapazes de enxergar; quanto à audição, 19,05% apresentaram alguma dificuldade para ouvir; 3,59% apresentaram grande dificuldade e 0,68% são incapazes (NERI & SOARES, 2004).

2.1.2 Deficiência Verbal

Associada a deficiências como a mental, ausência ou danos em membros superiores e tetraplegia ou hemiplegia está a dificuldade na comunicação seja oral ou escrita, o que pode afetar drasticamente a vida da pessoa e de sua família. Pinker e Trovato (1997) ressaltam que a linguagem entrelaça-se tão fortemente à vida humana que se torna difícil imaginar vida sem ela, e afirmam, ainda, ser a perda da comunicação oral devastadora, podendo fazer com que a sociedade, nos casos mais graves, não encontre mais as qualidades da pessoa que perdeu o poder de se expressar através da linguagem.

A deficiência verbal quebra o meio de interação direta entre as pessoas, pois impõe limites à relação interpessoal que se baseia na palavra. É comum que a pessoa com deficiência verbal apresente comprometimento motor, tornando outras formas de comunicação, como gestos e escrita, também dificultosos (GAVA, 1999).

As deficiências motoras ou danos que atinjam membros superiores impedem ou dificultam a comunicação escrita, tornando a pessoa um falante que só se comunica diretamente ou por sistemas de voz (telefone, celular, bip), mas que não consegue escrever uma carta, uma mensagem eletrônica e que se distancia das salas de aula e do mercado de trabalho. Sistemas reconhedores de voz, como o IBM *ViaVoice*[®], auxiliam aquele com a deficiência motora mas, no ambiente de trabalho, pode ser incômodo, e não auxilia a pessoa com deficiência verbal associada.

2.1.3 Comunicação Alternativa e Aumentativa

Considerando as limitações impostas à pessoa que tem uma deficiência na comunicação, seja escrita ou oral, nota-se que sistemas alternativos de comunicação fazem-se necessários. O termo alternativo pode estar ligado a vários significados como o de proporcionar economia nos esforços envolvidos na comunicação em termos de atenção, entusiasmo e recompensa (MORETTI, 1999). O desenvolvimento desses sistemas é orientado pela Comunicação Alternativa e Aumentativa (CAA) ou *Augmentative Alternative Communication* (AAC), sigla cunhada na década de 80, nos Estados Unidos da América na ocorrência da instituição de uma associação internacional chamada ISAAC (*International Society for Augmentative Alternative Communication*), fruto do trabalho multidisciplinar que objetiva o direito de comunicação mesmo para pessoas com grave impedimento verbal (GAVA, 1999).

Resultantes desse esforço multidisciplinar, vários aparatos têm sido disponibilizados à população com dificuldade em se comunicar:

1. sistemas de figuras que substituem as palavras por ícones, como a linguagem Bliss (GAVA, 1999), a *Picture Communication System* - PCS (JOHNSON, 1981 e 1985), o *Picture Ideogram Communication System* – PIC (MAHARAJ, 1980) e o sistema Rebus de Woodcock, Clark e Davies (1974);
2. meios de comunicação com síntese de voz (*Voice Output Communication Aids* – VOCAs), os quais são aparatos de voz eletrônica dedicados à comunicação verbal

percorrendo desde simples gravadores contendo mensagens mais usuais, pranchas portáteis de comunicação, como a *AlphaTalker*⁵, *SpringBoard*⁶ e *Pathfinder*⁶ e o Teclado para Comunicação Alternativa que busca oferecer uma alternativa de custo mais baixo e dedicada à criança brasileira (LIEGEL & NOHAMA, 2006), até computadores portáteis adaptados (MIZUKO, 1993);

3. *softwares* auxiliares como Amplisoft (JORDAN, MATIAS & NOHAMA, 2006), PCS em *software*⁷, emuladores de teclado (HENZEN, 2003, MATIAS & NOHAMA, 2003), e o Dosvox (BORGES, 2000) que através da síntese de voz auxilia deficientes visuais no estudo da informática e sua aplicação no ensino (FREITAS *et al.*, 2004);
4. dispositivos alternativos de controle do computador, como apontadores comandados por cabeça (JORDAN *et al.*, 2004, EVANS, DREW & BLENKHORN, 2000, MULLER *et al.*, 2004), ou olhos (DIAS *et al.* 2004) e acionadores, como o *Tash*⁷;
5. telefones celulares adaptados (LEE *et al.*, 2001).

Acredita-se que num futuro próximo até robôs pessoais que auxiliem a pessoa com deficiência a não ficar desocupada, sejam disponibilizados. Tal dispositivo poderá orientar a pessoa, incitá-la a realizar alguma atividade, auxiliar na tomada de decisões e no registro de dados, como uma agenda com alarme e reconhecimento de número de telefones e endereços. Ainda, poderá controlar objetos da casa, como portas, travas, iluminação e eletrodomésticos. Já sob o foco da visão, o robô pessoal poderá receber a informação, torná-la mais acessível e retransmiti-la ao usuário sob formas por ele compreensíveis (imagens, sons ou escrita), como também repetir informações e outras atividades a serem projetadas (PEÑA-CASANOVA & PAMIES, 2005).

Muitas dessas tecnologias desenvolvidas estão disponíveis à população, porém, para adquiri-las é necessário desembolsar um razoável montante de dinheiro, o que faz com que famílias e escolas abandonem a idéia de adquiri-las, pois há outros itens custosos e essenciais, como medicamentos, cadeiras de rodas, colchões especiais, alimentação e planos de saúde. A tecnologia é, então, descartada, mas a CAA não, bastando criatividade e envolvimento de professores e familiares para que, com recortes de revistas, fotografias e alguns materiais como embalagens, papelão, cola e tesoura sejam montadas pranchas de comunicação através

⁵ Disponível no endereço eletrônico: <http://www.coping.org/>.

⁶ Disponível no endereço eletrônico: <http://www.prentrom.com/>.

⁷ Disponível no endereço eletrônico: <http://www.clik.com.br/>.

das quais a pessoa com deficiência verbal possa comunicar-se, mesmo que de forma limitada (PESCINA, 2004).

No entanto, se mais técnicas de baixo custo ou gratuitas forem desenvolvidas e disponibilizadas à população, uma melhoria evidente de vida será promovida. Com esta visão, pesquisadores têm desenvolvido ferramentas gratuitas como os aplicativos Dosvox (BORGES, 2000), Comunique (PELOSI, 1999) e Amplisoft (JORDAN, MATIAS & NOHAMA, 2006). Também vários simuladores de teclado, aplicativo foco deste trabalho, encontram-se disponíveis, gratuitamente ou por custos baixos, na Internet (BARTH & SANTAROSA, 2006); porém, apresentam alguma limitação de uso, como pouca funcionalidade, pequeno espaço para a produção de textos, não oferecem opções ou não se adaptam às realidades do usuário, e poucos aceleradores de uso (p.ex., predição de palavras), como relatam em seu trabalho Juliato *et al.* (2004).

2.1.4 Processamento de Linguagem Natural

A fim de diminuir a distância entre homem e máquina, o Processamento Automático da Linguagem Natural (PLN), tem se destacado como um instrumento auxiliar, permitindo o desenvolvimento de ferramentas que tornam comunicação homem-máquina mais natural (GASPERIN & LIMA, 2001).

A tecnologia da informática permite o armazenamento e estudo digital de grandes *corpora* (plural de *corpus*), favorecendo a compilação, análise estatística, descrição e estudo da linguagem de um país ou região, classificação das palavras e aplicação de *corpora* no ensino de língua, reconhecimento de voz, tradução e predição de palavras (SARDINHA, 2004).

A análise estatística da linguagem é realizada com a aplicação de abordagens quantitativas no processamento automático de textos. De um *corpus* podem-se extrair informações sobre as palavras que o compõe, sobre a estrutura da linguagem apresentada, expressões coloquiais, vícios de linguagem, entre outros pontos interessantes para o total conhecimento de uma língua e aplicação desta em aplicativos informatizados. A frequência de co-ocorrência de determinadas palavras indicam expressões que têm um único significado, somente quando estas palavras estão juntas num texto (GASPERIN & LIMA, 2001).

A forma como os lexemas⁸ apresentam-se no *corpus* permite analisar seu comportamento para toda a linguagem, assim como suas propriedades (restrições de complementos e classes dos lexemas). Desta forma, palavras desconhecidas podem ter seu significado estimado com base nas propriedades das palavras conhecidas e das restrições obtidas no conhecimento lexical. Este processamento permite avanço em diversas áreas, como na própria lingüística, na tradução e em áreas específicas como a Medicina (ANDRADE *et al.*, 2005). O processamento estatístico de um *corpus* permite, ainda, quando baseado na estrutura da linguagem, o estudo da sintaxe das sentenças e, a partir de regras estabelecidas de seqüências de classes gramaticais encontradas no *corpus*, pode-se prever a classe da próxima palavra a compor um texto sendo digitado ou ditado, com auxílio de ferramentas estatísticas, como simples estatística ou Modelos de Markov (GASPERIN & LIMA, 2001).

2.1.5 Processamento de *corpus*

Todo estudo realizado sobre um *corpus* exige uma série de trabalhos para a seleção, coleta e pré-processamento dos textos, para determinar as unidades com que se deseja trabalhar. Para tal, existem trabalhos já realizados e que podem ser utilizados no desenvolvimento de novas pesquisas. Encontram-se *corpora* prontos em diversas línguas, baseados em textos específicos, como o TeMário, *corpus* em Português para o qual foram coletados 100 textos jornalísticos do Jornal Folha de São Paulo e Jornal do Brasil, totalizando 61.412 palavras (PARDO & RINO, 2003), o Natura-Público, constituído dos dois primeiros parágrafos de cada notícia do Jornal O Público, de Portugal, de 1991 a 1994, contendo 5.730.035 palavras, como pode-se observar no sítio dedicado a estas pesquisas: <http://acdc.linguateca.pt/acesso/contabilizacao.html#natura>.

Ferramentas para etiquetagem e softwares para operacionalização da descrição lingüística como, respectivamente, o *Visual Interactive Syntax Learning* (etiquetador *on-line* disponível para 19 línguas) e o *WordSmith Text Converter*[®], software para estudo estatístico dos textos (SARDINHA, 2004), auxiliam no processamento do conjunto de textos para aplicações lingüísticas.

Os textos de um *corpus* podem ser obtidos digitalmente ou gravados a partir de falantes. Os textos digitais atualmente são extraídos da Internet, devido ao seu vasto leque de

⁸ Lexema consiste em uma entrada no léxico (dicionário) que apresenta um único significado.

opções e facilidade de acesso. Para auxiliar esse trabalho, existem os programas que gravam fielmente e na íntegra o sítio de um servidor (*offline browsers*); assim, o usuário poderá navegar pelo sítio sem estar conectado à Internet. Neste processo, os textos acabam carregando informações não interessantes ao processamento lingüístico; então, necessita-se limpar os textos.

Para este serviço, podem ser utilizados aplicativos como o *Perl*[®] e o emulador de Unix *Cygwin*[®]. O *Perl*[®] é uma linguagem de programação com aplicações como encontrar, substituir e formatar. Já o *Cygwin*[®] emula o sistema operacional Unix no ambiente *Windows*[®] disponibilizando operações como busca e substituição (*search and replace*), muito úteis nesta fase do processamento (SARDINHA, 2004).

Com o *corpus* pré-processado em mãos, podem-se realizar os trabalhos lingüísticos desejados, bastando escolher qual o PLN a ser realizado e através de qual enfoque estatístico, com base em que técnica. Quando o PLN refere-se à palavra, pode-se estar buscando estudar as colocações (estudo de seqüências de palavras que, frequentemente, são utilizadas juntas); a inferência estatística (coleta de dados sobre freqüência das palavras e inferir sobre sua distribuição de probabilidade); a busca pela diminuição da ambigüidade semântica das palavras que apresentam diversos sentidos; e a aquisição léxica (descobrir e analisar as propriedades das palavras) (GASPERIN & LIMA, 2001).

2.1.6 Predição de Palavras

A inferência estatística é aplicada na previsão da próxima palavra durante a elaboração de uma sentença, quando conhecidas as palavras anteriores. Agrupando todas as seqüências presentes em um *corpus* de tamanho n e que são iniciadas pelas mesmas $n-1$ palavras, em uma classe, supondo que o contexto atual influencia o seguinte e estabelecendo um modelo de Markov de $(n-1)$ -ésima ordem, consegue-se prever a continuidade da seqüência de palavras (GASPERIN & LIMA, 2001).

Esta técnica conhecida por predição de palavras (*word prediction*), é muito utilizada no reconhecimento da fala e da escrita, destacando-se como principal ferramenta matemática os Modelos Ocultos de Markov, conhecidos como HMM, do inglês *Hidden Markov Models* (RABINER, 1989, SCAVONE, 1996, RAY & CRAVEN, 2001, BRITTO, 2001, BRITTO *et al.*, 2003 e ZIMMERMANN & BUNKE, 2004.). Ainda, o HMM é considerado a principal ferramenta no processamento estatístico da linguagem natural (GASPERIN & LIMA, 2001).

Muitos são os trabalhos encontrados na literatura técnica em que os HMMs são a base da busca pela solução de problemas estatísticos que apresentam seus fatos de forma seqüencial e em que uma ocorrência depende da anterior (BORGES, 2004).

Também se encontra a predição de palavras aplicada a teclados de celulares para a edição de mensagens, citando as técnicas T9[®], também presentes nos PDAs (*Personal Digital Assistants*, ou Assistente Pessoal Digital), patente da *Tegic Communications Inc.* (SOUKOREEF, 2002), e ITaP, desenvolvida pela Motorola[®] (KLARLUND & RILEY, 2003); e, como um acelerador de uso de simuladores de teclado, a predição é encontrada.

Os simuladores Click-N-Type e Dasher (FONSECA & CARVALHO, 2002, JULIATO *et al.*, 2004, MACKAY, 2005), possuem predição de palavras a fim de diminuir a dificuldade de seus usuários em produzir textos. Suas técnicas baseiam-se numa lista de textos (*corpus*), a partir do qual são encontradas as palavras já digitadas e contadas as que mais ocorreram na seqüência destas.

2.1.7 Amplisoft

Outro aplicativo com predição é encontrado no grupo de *softwares* Amplisoft desenvolvido pelo grupo do Prof. Nohama da Pontifícia Universidade Católica do Paraná e na Universidade Tecnológica Federal do Paraná, no qual se disponibiliza um emulador de teclado que oferece diversas opções de aumento na acessibilidade, como: tamanhos diferenciados do teclado na tela do computador; auto-clique que dispensa o clique manual do mouse convencional, bastando apontar o botão desejado com o cursor do mouse, através do uso de apontadores alternativos anteriormente citados, e aguardar o tempo pré-ajustado para que o botão seja selecionado; varredura eletrônica dos botões do teclado; interface com outros aplicativos da plataforma Windows[®] permitindo que o texto digitado através dele seja enviado para editores de texto, como Word[®] e Notepad[®], navegadores da Internet (p.ex., Google), aplicativos matemáticos (Excel[®] e calculadora) e sistemas de conversação (p.ex., Messenger[®]); teclas de atalho para abrir os principais aplicativos de texto, matemática e Internet; predição de palavras (JORDAN, MATIAS & NOHAMA, 2006); e síntese de voz.

A predição de palavras do Teclado Virtual Livre pode sugerir uma palavra a partir de uma letra digitada (ato de completar) ou prever a mais provável de suceder uma palavra digitada. A seleção das palavras ocorre pelo reconhecimento do símbolo (caractér) digitado: o sistema busca as combinações mais ocorridas com este símbolo numa lista de 3635 palavras

extraídas de literatura pré-escolar (POLLO, 2003) e indica, em campo dedicado, os dez mais prováveis de ocorrerem, caso o símbolo digitado seja uma palavra, o sistema indicará as dez mais prováveis de seguirem-na. A avaliação ocorre com base no histórico de uso, com pesquisa do próximo símbolo assim que um é selecionado, atualizando a área de predição e reiniciando esse sistema, como ilustra o fluxograma da Figura 2.

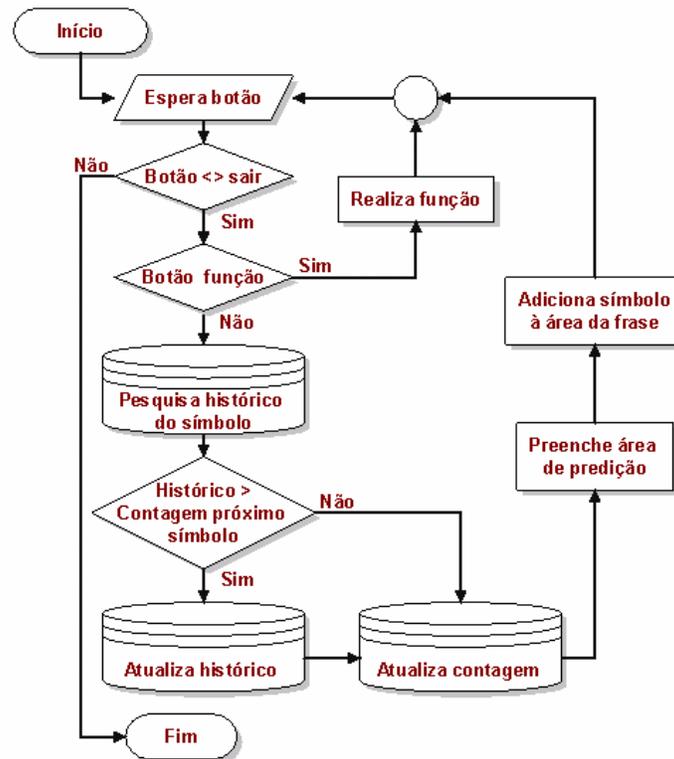


Figura 2: Fluxograma da predição de palavras do Teclado Virtual Livre (adaptada de MATIAS & NOHAMA, 2003).

Este tipo de avaliação é também utilizado em outros simuladores de teclado como SofType (JULIATO *et al.*, 2004) e o Teclado Virtual Dinâmico (FONSECA & CARVALHO, 2002). Sua técnica acaba gerando alguns problemas, pois como ele se baseia no histórico que, inicialmente contém somente uma lista de palavras, a predição da próxima palavra só ocorrerá quando a palavra digitada já tiver sido utilizada em algum contexto, e o ato de completar indicará as palavras que ocorrem com a letra selecionada em ordem alfabética. Somente após um período de utilização que o aplicativo se identificará ao seu usuário, predizendo as palavras e textos mais utilizados por ele. Outro problema é a desconsideração de erros gramaticais (como concordância verbal e nominal) e de ortografias.

Estes problemas ocorrem devido ao fato da técnica de predição utilizada realizar análise estatística simples e não alguma técnica de tomada de decisão mais elaborada, como

redes neurais e mistas (ROSA, 2002, NAKAMURA *et al.*, 1990), HMM (THEDE & HARPER, 1999), e algoritmos genéticos (MARTINEZ, 1998); e baseada em um *corpus* composto por textos de linguagem específica, de acordo com a aplicação vislumbrada, como de textos infantis, no caso deste trabalho.

O trabalho de Nakamura *et al.* (1990) baseia-se numa rede mista de decisões com redes neurais associadas ao HMM, no qual, para realizar o reconhecimento da fala, prevê a classe gramatical da próxima palavra e, quando esta é falada, busca sua identificação dentro dos grupos de classes indicados como mais prováveis. Caso a palavra falada não seja encontrada, ela é inserida pela técnica de redes mistas na classe mais provável de ocorrer naquela posição da sentença falada. Desta forma, a base de textos torna-se dinâmica e adapta-se ao usuário, fazendo com que o número de acertos aumente. Trata-se de um sistema computacional mais elaborado que, conseqüentemente, exige maior esforço computacional.

Tem-se desenvolvido pesquisas para a predição de palavras, utilizando-se HMM aliado à classificação das palavras conforme suas categorias gramaticais. Desta forma, reduz-se o número de iterações matemáticas envolvidas na busca pelas seqüências de palavras num *corpus*, pois prediz inicialmente as classes gramaticais mais prováveis de completarem a seqüência para depois buscar, dentro de cada classe, a palavra mais indicada. Ainda, esta metodologia permite que, ao trabalhar com diferentes *corpora*, uma palavra não presente no treinamento seja de possível predição. Discorrem sobre este tema Niesler & Woodland (1999), Zimmermann & Bunke, (2004), Kuhn, Niemann & Schukat-Talamazzini, (1994), Law & Chan (2006) e Clarkson (1999).

Na seqüência, descrever-se-ão os tópicos citados nesta revisão literária de forma mais detalhada.

2.2 TÓPICOS TEÓRICOS

Como visto no texto anterior a este tomo, o trabalho envolve diversas áreas do saber, percorrendo desde os conceitos de acessibilidade, linguagem, estatística de *corpus*, predição de palavras e desenvolvimento de algoritmo. Para que nenhum dos conhecimentos envolvidos seja desconhecido, a seguir introduzir-se-ão as teorias envolvidas.

2.2.1 Acessibilidade

O termo acessibilidade, que no dicionário se encontra com o significado “facilidade na aproximação, no trato ou na obtenção” de algo (BUENO, 2000), quando relacionada à pessoa com alguma deficiência, refere-se à liberdade, aos direitos à independência, ao ir e vir, à educação e à saúde. Logo, abrange: ter ruas com condições para trafegar com cadeira de rodas ou bengalas (SOUSA, 2003), dispor de escolas preparadas para receber igualmente as crianças com e sem deficiências (ABDALLA *et al.*, 2005), embalagens, placas e semáforos adaptados às necessidades do deficiente visual, com dispositivos Braille e sinalizações sonoras (MOECKEL, 1992), respectivamente, postos de saúde, cinemas, bibliotecas, hotéis, supermercados, bancos e toda instalação comercial necessária à vida independente do ser social.

Ainda, na informática, a acessibilidade, tem um significado especial, representa ao usuário o direito de ter acesso a toda teia de informações que a informática forma e, também, o direito ao controle de seus utensílios (computadores pessoais e portáteis, PDAs e seus periféricos), vencendo as barreiras arquitetônicas. Desta forma, todo usuário deve ter disponibilidade de comunicação, de dispositivos e aplicativos adequados às suas necessidades, de conteúdo e interface adaptados, sempre que necessário, em formatos alternativos. Estas exigências são tratadas e apresentadas no sítio <http://www.acessobrasil.org.br/>.

O programa Acessibilidade Brasil indica quais as condições especiais a serem consideradas e transpostas pelos criadores de aparatos de informática e desenvolvedores de sítios, objetivando extinguir a falta de acesso de qualquer cidadão, sendo elas:

1. incapacidade total ou parcial de ver, ouvir ou deslocar-se e a dificuldade de interpretação de algumas informações;
2. baixa visão;
3. incapacidade de uso de periféricos como teclado e mouse, por características físicas ou indisponibilidade dos mesmos;
4. conexão limitada à Internet;
5. deficiência verbal, afasia ou dificuldade de compreensão de textos escritos em outro idioma;
6. liberação do uso dos olhos, permitindo outras atividades concomitantes;
7. uso de navegadores com versão obsoleta ou diferente dos usuais, sistemas operacionais menos difundidos, ou, ainda, controlados por voz.

No site <http://www.acessobrasil.org.br/>, é possível encontrar regras para o desenvolvimento de páginas para a Internet e um *software* para avaliação de sítios quanto à acessibilidade, denominado “daSilva”.

Apesar da força tarefa mundial em tornar o computador um objeto comum dentro dos lares, nem sempre a distância entre a máquina e seu usuário fica limitada aos seus periféricos padrões (mouse e teclado). Uma pessoa com deficiência motora pode necessitar de um apontador de cabeça, como o ilustrado pela Figura 3a ou um mouse alternativo (JORDAN *et al.*, 2004, EVANS, DREW & BLENKHORN, 2000); já uma pessoa cega necessita que as páginas da Internet sejam lidas eletronicamente ou passadas a um meio de leitura, como a Linha Braille, ilustrada pela Figura 3b, que reproduz os textos presentes na tela do monitor em uma seqüência de células Braille eletronicamente⁹. Analogamente, pessoas com as mais diferentes necessidades especiais podem exigir a presença de algum tipo de adaptador à sua realidade para controlar o computador.



(a)



(b)

Figura 3: (a) Ponteiro de cabeça, (b) Linha Braille (extraídas de <http://www.acessobrasil.org.br/>).

Há os casos em que somente um adaptador externo (*hardware*) não é suficiente, necessitando de aplicativos (*software*) para o total controle da máquina. Nesses casos, inúmeros aplicativos são encontrados, como softwares para linguagens alternativas (JOHNSON, 1981 e 1985), emuladores de teclado e mouse (HENZEN, 2003, MATIAS & NOHAMA, 2003, PELOSI, 2003), reconhecedores de voz (BORGES, 2002), entre outros.

⁹ Disponível no endereço eletrônico: <http://www.acessibilidade.net/>.

Aplicativos e dispositivos externos aproximam a pessoa com deficiência da informática e de todas suas vantagens. Tal acessibilidade possibilita projetos de inclusão digital e social, focos de atenção do mundo atual.

2.2.2 Inclusão Digital e Inclusão Social

Para entender o porquê dos programas de inclusão social é necessário conhecer os quadros mundial e nacional de exclusão social. Em 2002, estimava-se que 125 milhões de crianças (sendo 83 milhões de meninas), em todo o mundo, não freqüentavam nenhum tipo de escola; que 150 milhões de crianças deixavam a escola antes mesmo de serem alfabetizadas; que 2 milhões de crianças, das 12 milhões que falecem todos os anos por doenças relacionadas à pobreza, poderiam ser salvas se fossem à escola; e que somente um número próximo a 1% das crianças com deficiência residentes nos países em desenvolvimento freqüenta algum tipo de escola (MITTLER, 2002).

A fim de reduzir tais números, países como a Inglaterra e o País de Gales, há mais de trinta anos proíbem qualquer tipo de exclusão nos setores da educação. No entanto, apesar do esforço político, não é incomum encontrar crianças matriculadas em instituições de ensino sendo atendidas por pessoal sem treinamento ou qualificação educacional (MITTLER, 2002).

No Brasil, um projeto intitulado “Programa Ética e Cidadania” objetiva implantar nas reuniões escolares os tópicos relacionados aos temas inclusão e exclusão social, principalmente sob o foco do âmbito escolar, esperando colaborar para a construção de ambientes de ensino verdadeiramente inclusivos, adaptados às diferenças e concentrados nos interesses e necessidades de todos os seres humanos (LODI, 2003).

O tema inclusão social pede que se esteja aberto às necessidades das pessoas com alguma deficiência; ao histórico de crianças e adolescentes egressas de sistemas sócio-educativos; à exclusão das mulheres, dos pobres, homossexuais, indígenas, afro-descendentes, sem-teto e idosos; pois, segundo o artigo 2º da Declaração Universal dos Direitos Humanos¹⁰, todo ser humano pode alegar em seu favor seus direitos e liberdades, sem distinção alguma “de raça, de cor, de sexo, de língua, de religião, de opinião política ou outra, de origem nacional ou social, de fortuna, de nascimento ou de qualquer outra situação”.

¹⁰ Disponível no endereço eletrônico: http://www.mj.gov.br/sedh/rndh/declara_uni.html .

Tem sido tema comum de programas de inclusão a discussão acerca da socialização de pessoas com algum tipo de deficiência. Busca-se uma sociedade que, alterando suas estruturas, crie acessos e abra espaços para que cada pessoa com deficiência seja capaz de interagir naturalmente com a sociedade em que vive. Assim, a pessoa com deficiência poderá ser reconhecida pelo seu potencial e habilidades. Esta Informação pode ser visualizada no sítio http://pt.wikipedia.org/wiki/Inclus%C3%A3o_social

Luta-se, então, para superar a exclusão e o individualismo, numa nova fase de humanização social, sendo necessário superar os problemas impostos pela contemporaneidade, adaptando carros, ônibus, calçadas, teatros, cinemas, bibliotecas e escolas, transpondo as barreiras físicas, psicológicas, espaciais, temporais, culturais, visuais, de comunicação e, acima de tudo, garantindo o acesso irrestrito de todos os seres vivos aos bens e riquezas de todo tipo, destacando-se entre eles, o conhecimento (LODI, 2003).

As instituições de ensino que decidem se encaixar nos programas de inclusão social devem se mostrar habilitadas a ensinar uma turma com crianças especiais e não, precisando reorganizar, muitas vezes, completamente os processos pedagógicos adotados; fazendo uso de ferramentas auxiliares que permitam ensinar a todos os alunos, sem discriminações. Para isso, fazem-se necessários o uso de computadores, pranchas de comunicação e gravadores de voz na sala de aula (LODI, 2003). Enfim, o acesso de que se fala exige uma plástica nos conceitos de educação que, na medida em que se entende por estética, refere-se a todos os seus fatores: paisagístico, arquitetônico, comunicacionais, pessoais, etc... (MONTEIRO, 1998 *apud* LODI, 2003).

Inclusão digital ou info-inclusão é a universalização do acesso aos meios, dispositivos, conteúdos e conhecimentos da informática. Programas de inclusão digital buscam prover o acesso às Tecnologias da Informação e Comunicação (TIC), a todas as pessoas, independente de condições financeiras e físicas, oferecendo a elas o ingresso ao universo de informações disponíveis na Internet. Tais projetos exigem consideráveis desafios dos países em desenvolvimento por necessitarem de grandes investimentos.

No entanto, o Brasil vem desenvolvendo programas de inclusão digital, tais como o projeto do Governo Federal, Computador para Todos - Projeto Cidadão Conectado¹¹, o qual permitiu o financiamento de mais de 19 mil máquinas. Apesar de tais esforços, o Censo Escolar de 2000 mostrou que menos de 40% dos estudantes de ensino médio frequentam escolas que possuem acesso à Internet (SILVEIRA, 2005).

¹¹ Disponível no endereço eletrônico: <http://www.computadorparatodos.gov.br>.

A invasão dos computadores na vida das pessoas, fortalecida por tais programas de inclusão, mostra que, além da população de baixa renda, outros grupos necessitam de apoio e atenção: idosos, pessoas que desconhecem totalmente a informática e deficientes, pois acredita-se que as TIC vieram e se estabeleceram de forma que, no futuro, quem não se encontrar preparado estará ameaçado por uma limitação social.

Especialmente, dentre os outros grupos alvos da inclusão digital, as pessoas com deficiência necessitam, como já citado, de dispositivos auxiliares para o controle da máquina. Logo, os programas governamentais devem não só mirar na aquisição ou acesso à máquina, mas na disponibilização de tais dispositivos, que geralmente são caros (PESCINA, 2004). Muitos dispositivos são desenvolvidos nas instituições de ensino superior e de pesquisa, mas pouco se divulga à população, fazendo com que pessoas que necessitam de ferramentas facilitadoras para o controle do computador acabem sem as mesmas, excluindo-se digitalmente. Este fato é grave, podendo interferir no desenvolvimento de uma pessoa, por exemplo, da criança que não consegue se comunicar naturalmente e é tida como cognitivamente comprometida, e que, através do computador, poderia mostrar suas habilidades (STELLA, 2002).

2.2.3 Comunicação Natural, Deficiências Verbal e Relacional

No estabelecimento da comunicação natural entre pessoas, a linguagem destaca-se como agente fundamental. A linguagem é a capacidade da manipulação das palavras para transmitir ao próximo um conceito elaborado por alguém e vice-versa, ou seja, a percepção das palavras proferidas pelo próximo e a concepção de uma idéia a partir delas. É responsável pela divulgação das descobertas, dos acontecimentos, dos receios e dos desejos da humanidade, podendo envolver não só os setores do corpo diretamente ligados à voz, como todo o corpo. Tal ato é visualizado na ocorrência dos sinais da Comunicação Não Verbal (CNV) que confirmam ou desmentem o que se fala através da quantidade e tipo de gestos, postura e olhares que acompanham uma mensagem (GAVA, 1999).

Ainda, pode-se definir a linguagem oral como sendo o sistema formado pelos signos fônicos (a linguagem escrita seria o sistema de signos gráficos), os quais são processados e transmitidos através de um meio próprio, pelo emissor para ser recebido pelo receptor. Entende-se por emissor aquele possuidor de uma informação e que deseja repassá-la acrescida de motivos; o meio próprio depende da linguagem, se oral será o ar, se escrita poderá ser um

papel, um celular, um PDA ou um computador, e depende de um código conhecido pelo emissor e pelo receptor (idioma ou linguagem de gestos e pictórica); enquanto o receptor será caracterizado por uma ou um grupo de pessoas que acolhe a mensagem e a interpreta através de sua estrutura psicológica (PEÑA-CASANOVA & PAMIES, 2005).

Lecours e colegas (*apud* PEÑA-CASANOVA & PAMIES, 2005) definem de forma técnica a linguagem, definindo-a como o fruto de um processo que envolve uma complexa atividade nervosa, através da qual a transmissão de estados psíquicos convertidos em signos convencionados por uma linguagem pode ocorrer. O ser humano facilmente compreende e é capaz de estruturar com naturalidade expressões e mensagens complexas através dos signos da linguagem falada no meio em que ele vive desde sua infância (GASPERIN & LIMA, 2001).

No entanto, os processos lingüísticos envolvidos em uma mensagem não amadurecem proporcionalmente ao preparo das capacidades conceituadas, o que se evidencia no caso de crianças que, apesar de apresentarem danos nos processos conceituais, são alfabetizadas. Isto aponta para a dedução de que ocorre de forma autônoma o amadurecimento dos sistemas neurais envolvidos nos processos lingüísticos do ser humano (GAVA, 1999). Assim, esclarece-se que uma pessoa, mesmo sem apresentar comunicação oral pode ter suas estruturas conceituais formadas e ser dona de potenciais escondidos pelo silêncio.

A falta do uso da palavra por uma pessoa é tida como uma deficiência verbal, podendo ser temporária (pós-trauma) ou permanente, e é resultado de uma patologia neurológica que bloqueia a comunicação oral (GAVA, 1999). Esta deficiência alerta para uma situação preocupante, pois o paciente estará impedido de manipular a linguagem, ou seja, estará sem condições de gerenciar grande parte dos processos de inferência (CANNAO, 1999).

Um tipo de deficiência verbal é a afasia, diagnosticada quando o paciente apresenta um déficit de linguagem, caracterizado pelo vocabulário reduzido, dificuldades de estruturar e compreender mensagens (ORTIZ, 2005). A afasia pode ser a dificuldade na manipulação das linguagens oral e escrita, como nas afasias expressivas ou emissivas, chamadas afasias motoras; ou pode ser a limitação da compreensão das linguagens oral e escrita, presente nas afasias receptivas ou de compreensão, chamadas de afasias sensoriais (ORTIZ, 2005).

O tratamento das afasias, no âmbito da terapia de reabilitação, visa ajudar o paciente a promover suas habilidades residuais, ou seja, em casos de distúrbios de fala, ampliar o uso da comunicação escrita ou simbólica (ORTIZ, 2005), especializando suas capacidades comunicativas e maximizando sua adaptação psicológica, emocional, familiar e social (PEÑA-CASANOVA & PAMIES, 2005).

Atualmente, a ciência fornece alternativas focadas em diversas abordagens, como técnicas de neuroimagem para diagnóstico (o desenvolvimento da tomografia computadorizada e da ressonância nuclear magnética representou um considerável progresso nas descobertas da relação clínico-topográfica), e a adoção de tecnologias e de computadores de uso pessoal, tanto para a investigação patológica como na terapia. O uso de computadores na terapia auxilia nos processos de dimensionamento dos danos causados pela patologia, registro, controle e relacionamento do paciente com o instrumento terapêutico.

Os métodos de terapia podem ser reunidos em dois grandes grupos, sendo um o de reeducação, o qual tenta recuperar a habilidade perdida pelo paciente, e o outro de compensação, que oferece ao paciente uma forma alternativa para a substituição da capacidade perdida (PEÑA-CASANOVA & PAMIES, 2005).

Um dos mais importantes fundamentos da terapia das disfunções comunicativas é a aplicação intensa de estímulos auditivos. Cabe ao terapeuta dimensionar o estímulo mais adequado a cada paciente, aquele que desperte sua atenção. A combinação entre a estimulação visual e auditiva pode auxiliá-lo nisto (ORTIZ, 2005).

A decisão do método terapêutico a ser adotado exige uma cuidadosa análise das habilidades do paciente para que não sejam oferecidas a ele tarefas que subestimem suas capacidades, mas que apenas facilitem o processo da comunicação perdida (CANNAO, 1999).

Pessoas afásicas vivenciam com frequência o “fenômeno da ponta da língua”, quando se sabe o que se deseja falar, mas não se recorda da palavra a ser utilizada. Para colaborar na solução deste problema, pode-se oferecer uma lista de palavras escritas ao afásico, desde que ele tenha capacidade de identificar a desejada na lista. Isto pode ocorrer no processo da fala ou da escrita, logo um aplicativo simulador de teclado, que permite a escrita, com predição de palavras pode ser indicado. Nota-se que ambos os métodos, de reeducação e de compensação, podem ser realizados com auxílio do computador (PEÑA-CASANOVA & PAMIES, 2005).

Um tipo de afasia, conhecida por agramatismo, apresenta como característica o esquecimento de alguns elementos comuns à estrutura de uma sentença. A omissão de palavras como artigos e preposições é a mais frequente ocorrência. O paciente, então, comunica-se através das estruturas ‘sujeito + verbo’ ou ‘verbo + objeto’. O agramatismo pode acometer a mensagem falada e escrita, mas não existe paralelismo entre elas (ORTIZ, 2005). Indica-se que sejam oferecidos a este paciente, métodos estimulatórios com diferentes formas de motivação sensorial (imagem, som, cores, escrita, etc.) e solicitar grande quantidade de tarefas evitando aprendizagens fixas que distanciariam o indivíduo da realidade (PEÑA-

CASANOVA & PAMIES, 2005). Para isso, o uso do computador e, para usuários com transtornos motores associados, um aplicativo facilitador da produção de textos, é útil.

Processos degenerativos podem desencadear um afastamento do indivíduo dos processos de leitura e escrita. A Doença de Alzheimer é o processo degenerativo demencial mais pesquisado na atualidade (ORTIZ, 2005) e é caracterizada pelas alterações comportamentais e pela perda gradativa das capacidades cognitivas. A observação de pacientes com Alzheimer de leve à moderada, mostrou que a produção textual proposta a partir da estimulação visual é afetada. A falta de organização das idéias, o excesso de repetição de palavras e trechos insignificantes caracterizavam os textos produzidos. Ainda se destacam o grande número de erros ortográficos (troca, acréscimo, alterações de posição ou supressão de letras na palavra) (ORTIZ, 2005). Novamente, ferramentas de apoio na estruturação de textos podem ser de grande valia a tais pacientes.

2.2.4 Uso dos Computadores na Terapia da Deficiência Verbal

Nas técnicas terapêuticas de indivíduos com distúrbios de comunicação, a implantação computacional deve superar três barreiras (STELLA, 1999):

1. instrumental: sob este foco, a máquina serve como prótese que desempenha a atividade executiva, gerenciada pelo paciente. Cabem neste problema as soluções de interface para que indesejáveis desentendimentos ocorram, como sobreposição de mensagens ou mensagens falsas, pois a interface deve respeitar o tempo compreendido entre a intenção da informação até sua realização completa;
2. interpretação: o computador pode tentar ampliar ou substituir os transtornos sensoriais presentes, transformando as mensagens recebidas em sinais compreensíveis a seu usuário (sinais sonoros em táteis, por exemplo). Há que se destacar neste tema as dificuldades de interpretação de uma mensagem, pois a mesma mensagem pode ter diferentes significados, dependendo do tom de voz, p.ex., ou quando um mesmo símbolo possui mais de um significado. Exige-se da máquina, então, que tome decisões;
3. representação: seria a máquina como prótese mental do paciente que está impossibilitado de desempenhar atividades intelectuais. Seria responsabilidade, total ou parcial, da máquina a apresentação dos objetivos a serem alcançados numa atividade, a programação das atividades, as decisões e o controle das tarefas executadas.

O uso dos computadores como apoio nos métodos convencionais de terapia apresenta as seguintes vantagens citadas por Peña-Casanova & Pamies, (2005):

1. aplicabilidade, motivação e flexibilidade: o computador permite que sejam conhecidos e manipulados dados obtidos numa sessão de terapia, como tempo de estímulo e de reação. Através de *softwares* dedicados, por exemplo, pode-se alterar o estímulo (imagem na tela) dependendo da resposta recebida, percorrendo etapas de um jogo (NOHAMA *et al.*, 2001). O uso do computador por pessoas com deficiências motoras é garantido por dispositivos alternativos de acessibilidade, já citados;
2. controle, manipulação de dados e realimentação: facilmente, os aplicativos permitem visualizar o desempenho do paciente, seus progressos e retrocessos. Esta aplicação elimina o envolvimento pessoal do terapeuta com o paciente que, por tendência intrinsecamente humana, tenta dissimular os erros de seus pacientes. Além disso, Bradley e colegas (*apud* PEÑA-CASANOVA & PAMIES, 2005) citam que pacientes relatam sentirem-se menos humilhados ao serem “corrigidos” pelo computador do que pelo terapeuta;
3. super-aprendizagem: com a apresentação repetida de informações, agregada às estimulações sonoras e visuais, além da esperada aprendizagem por pacientes com lesão cerebral, pode ocorrer a super-aprendizagem, devido aos estímulos oferecidos e à grande capacidade de repetição do computador;
4. aprendizagem ocasional: o gosto desenvolvido pela informática nas sessões de terapia, leva pacientes a adquirirem um computador pessoal e desenvolverem habilidades adicionais com o uso de novos aplicativos.

O funcionamento cognitivo também pode ser relacionado ao uso do computador. Stella (1999) relata que crianças tratadas com auxílio do computador apresentavam maior motivação e um incremento na atenção prestada. Os elementos básicos do funcionamento mental são estimulados pelas cores e sons da máquina, o que facilita a compreensão das atividades a serem executadas. Ainda, por ser a resposta dada pela criança através do computador completamente entendida pelo terapeuta, independente de escuta, escrita, desenhos, etc., evita-se repetição de tarefas já superadas pela criança e gera-se uma maior interação com a tarefa e seu objetivo comunicativo e terapêutico (STELLA, 1999).

Stella (1999) narra que, em sua experiência profissional, os aprendizados estimulados pela máquina ultrapassam as atividades manuais, alcançando diferentes estruturas mentais e ativando novas formas de funcionamento.

É importante analisar a aceitação da máquina pelo paciente, o aspecto inovador pode parecer agressivo para idosos ou pessoas com pouco acesso a tecnologias atuais, sempre contrabalançando com o cuidado de não propor ao usuário tarefas abaixo de sua linha de competências praticáveis (CANNAO, 1999).

Constata-se que o computador pode ser uma ferramenta muito útil quando aplicado como apoio na terapia de deficiências verbais. Mas, para isso são necessários aplicativos dedicados e dispositivos de acessibilidade, os quais devem ser desenvolvidos sob o foco da ampliação ou da apresentação de alternativas de comunicação à pessoa com deficiência verbal. Tais trabalhos são regidos pela ciência conhecida por Comunicação Alternativa e Aumentativa.

2.2.5 Comunicação Alternativa e Aumentativa (CAA)

Hoje, a CAA é definida como uma ciência multidisciplinar (oferecida como especialização profissional), que abraça um universo de elaborações teóricas, soluções por sistemas de sinais, sustentação técnica e estratégias de intervenção destinadas à substituição ou incremento da fala, da escrita ou do controle do meio em que se vive quando as vias naturais encontram-se lesadas (PEÑA-CASANOVA & PAMIES, 2005).

Entende-se por alternativa a substituição da comunicação natural por símbolos (ícones ou gestos) ou voz eletrônica, quando a pessoa não apresenta formas de se expressar; e por aumentativa, a diminuição das dificuldades em se comunicar como déficit de atenção, falta de estímulo e de compreensão, o que é representado pela pessoa que ao utilizar uma forma própria de se expressar não é entendida, ou seja, quando a pessoa consegue de alguma forma natural se comunicar, mas não de forma suficiente às interações sociais (MORETTI, 1999, PELOSI, 2003).

Enfim, a CAA tem como ideal permitir que a pessoa com deficiência verbal apresente-se à sociedade como um ser comunicativo, utilizando-se de instrumentos que a auxiliem a transpor as barreiras impostas pela deficiência (CANNAO, 1999).

A hoje tão difundida CAA surgiu na década de 50, com o desenvolvimento de pranchas de comunicação por terapeutas e seus pacientes deficientes verbais. Em 1963, surge o primeiro aparato de CAA, composto de uma máquina de escrever controlada por um acionador e dotada de um sistema de varredura em suas teclas. Na década de 70, adotou-se a linguagem de sinais usada para a comunicação de pessoas com deficiência auditiva, para

peças com afasia, deficiência mental e autismo. Nos anos 80, a constituição da ISAAC divulgou os objetivos da CAA e os trabalhos realizados por ela. Tal associação internacional foi criada com os objetivos de divulgar as técnicas desenvolvidas para auxiliar a pessoa com dificuldades em se comunicar, instigar pesquisas na área e promover uma alteração cultural para que a sociedade aceite a substituição da palavra (PELOSI, 2003, GAVA, 1999).

Os cientistas daquela época perceberam que os resultados positivos obtidos com o uso da linguagem de sinais por parte de pessoas com deficiência auditiva e do Braille, pelos deficientes visuais, poderiam ser ampliados às pessoas tidas como incapazes, por não se comunicarem. Desta forma, desenvolveram-se sistemas simbólicos alternativos ao uso da palavra, trabalho que exigiu muita dedicação e pesquisa, pois seriam representados não somente objetos, mas expressões e sentimentos, conceitos abstratos que deveriam ser compreendidos, memorizados e utilizados por crianças e por pessoas com danos cognitivos (GAVA, 1999). Surgiram as linguagens Bliss e PCS (PELOSI, 2003).

O sistema Bliss constitui-se por símbolos gráficos que devem ser combinados para assumirem um significado (NUNES, 1999), como exemplificado pela Figura 4.

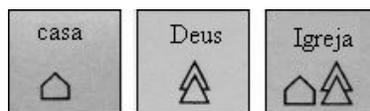


Figura 4: Exemplo de combinação de símbolos no Sistema Bliss de Comunicação, no qual a soma dos símbolos casa e Deus indicam a “casa de Deus” ou igreja (adaptada de VASCONCELLOS, 1999).

Já a linguagem pictórica PCS (JOHNSON, 1981 e 1985), mostrada na Figura 5, inicialmente desenvolvida para uso em materiais de apoio, como tábuas, pranchas, aventais e quadros, e tinha 700 símbolos, hoje apresenta mais de 3000 ícones e continua sendo uma ferramenta utilizada para a comunicação, estando disponível até em *software* comercial.



Figura 5: Exemplo de ícones PCS (extraída de <http://www.clik.com.br/>).

Desde então, ininterruptamente dispositivos, sistemas e todo tipo de aparato tem sido entregue à sociedade com o intuito de ampliar a comunicação de quem tem dificuldade em obtê-la, desenvolvidos por uma rede de cientistas, sendo que, em 1997, o número de membros da ISAAC beirava 2500 em 50 países (PELOSI, 2003).

No Brasil, terapeutas e pesquisadores das principais cidades, como Rio de Janeiro, São Paulo, Curitiba e Porto Alegre têm aplicado os conceitos e técnicas da CAA e participado do desenvolvimento da mesma. Também se encontra CAA nas escolas de ensino especial e nas clínicas de terapia ocupacional (PELOSI, 2003).

Como o público alvo da CAA apresenta uma diversidade de necessidades distintas, apresentando o comum fato de não se comunicarem naturalmente, diferentes técnicas são necessárias dentro de uma única escola ou clínica, como somente pranchas de comunicação, papéis e cartazes para pessoas com cognitivo e habilidade motora íntegras, até sistemas computadorizados para pessoas com transtornos motores e relacionais (GAVA, 1999). Todo o estudo que se faz em CAA baseia-se no anseio de sustentar o desejo, nas pessoas com deficiência verbal, de se comunicar (TUPY & PRAVETTONI, 1999).

No sítio da Professora Miryam Pelosi¹², encontra-se uma breve lista dos recursos mais utilizados na CAA, sendo eles divididos em dois grupos. O primeiro, composto por dispositivos de baixa tecnologia, engloba:

1. objetos reais: exemplares idênticos ou semelhantes do significado, podendo ter variações de tamanho, cor ou outra característica;
2. miniaturas: objetos de tamanho diminuído devem considerar possíveis deficiências visuais e intelectuais dos pacientes;
3. apresentação de parte de objetos: quando o objeto a ser representado for muito grande, pode-se utilizar uma parte para representar o todo;
4. fotografias, recortes de revistas ou jornais e embalagens;
5. símbolos gráficos: como PCS (JOHNSON, 1981 e 1985), Amplisoft (JORDAN, MATIAS & NOHAMA, 2006), Bliss (NUNES, 1999), etc.;
6. pranchas de comunicação: de uso individual, as pranchas são elaboradas a partir do uso de objetos ou símbolos, letras, palavras inteiras ou fracionadas, entre outros, ícones que são apontados pelo paciente dependendo de sua vontade ou quando solicitado. Elas podem ser utilizadas de forma avulsa ou agrupadas;

¹²

Endereço eletrônico: <http://www.comunicacaoalternativa.com.br/>.

7. *eye-gaze*: pranchas cujos ícones são selecionados pelo olhar ou por fontes luminosas colocadas na cabeça do paciente. Elaboradas e dispostas em formato de ferradura, são posicionadas sobre uma mesa ou colocadas na vertical;
8. avental: confeccionado em tecido que permita a fixação de símbolos velcro.

E o segundo grupo envolve recursos de alta tecnologia, entre eles:

1. comunicadores com voz gravada;
2. comunicadores com voz sintetizada: nos quais as mensagens são lidas por voz eletrônica;
3. computadores: no incentivo ao aprendizado, apresentação e controle de tarefas.

Os sistemas computacionais da atualidade oferecem ao terapeuta e ao usuário diversos tipos de auxílio, como sistemas reconhedores de voz (NOHAMA *et al.*, 1991), simuladores de prancha de comunicação (JORDAN, MATIAS & NOHAMA, 2006), softwares de comunicação, como o Comunique (PELOSI, 2003), e simuladores de teclado e mouse, foco deste trabalho (HENZEN, 2003), entre outros.

2.2.6 Desenvolvimento e Processamento de *Corpus*

O desenvolvimento de ferramentas computacionais com tomada de decisão nos processos textuais envolve estudos lingüísticos que servem como base às tarefas de decisão, como, por exemplo, um simulador de teclado com predição de palavras, do qual se espera obter não somente uma lista de palavras com maior ocorrência, mas uma lista que se baseie na interligação inteligente entre essas palavras e suas ocorrências sequenciais.

Para que essas tarefas sejam possíveis, o aplicativo deve ter em suas fontes de dados uma coletânea de textos com as características desejadas pelos seus desenvolvedores.

Esta coleção de textos é conhecida como *corpus* e definida como um conjunto de textos naturais e autênticos selecionados com o fim de caracterizar um estado ou uma variação da linguagem que se deseja analisar, sustentando uma investigação lingüística (SARDINHA, 2004). Um *corpus* é sempre elaborado com um pré-projeto de objetivos exclusivos e deve constituir-se somente de textos oriundos da produção humana, excluindo-se textos produzidos por sistemas de geração textual, podendo ser escritos ou falados.

Geralmente, um *corpus* é extenso para que seja representativo da linguagem ou de um idioma alvo da pesquisa e necessita dispor-se de tal forma que torne possível a sua

manipulação por processadores computacionais que extraem dados utilizados na investigação a ser feita. Sardinha (2004) afirma que a extensão do *corpus* assume ser tridimensional. Sua primeira dimensão é o número de palavras que ele contém, salienta-se que quanto maior esta dimensão, maior a probabilidade de palavras de baixa frequência de uso em uma linguagem ou idioma ocorram. A segunda caracteriza o número de textos, a qual está relacionada aos *corpora* de textos característicos de algum grupo, como dialetos, pois quanto maior a quantidade de textos mais fácil se torna a tarefa de caracterizar esta especificidade. A terceira dimensão refere-se ao número de gêneros ou tipos de texto, quando os *corpora* são utilizados para configurar uma linguagem toda.

Devido a atual facilidade de acesso à *World Wide Web* (www), ou seja, à Internet, possibilidade de acessar textos sem necessitar armazená-los e encontrar um universo de mensagens multilíngüe e de assuntos variados, o computador conectado passou a ser a maior ferramenta de garimpo textual para a estruturação de *corpus*. Esta tarefa pode ser realizada com facilidade se utilizado um aplicativo que archive eletronicamente os textos extraídos da Internet (*offline browser*), agregado a um aplicativo de limpeza de textos que são armazenados com informações irrelevantes tais como cabeçalhos e propagandas, como descrito na revisão sobre *corpus*, no item 2.1.5.

O *corpus* estruturado deve estar organizado de forma a facilitar as tarefas de pesquisa e propiciar a utilização do mesmo por outros grupos de pessoas. Não havendo regras para tal tarefa, Sardinha (2004), em seu livro sobre *Linguística de Corpus*, indica um método para tal (pp.72-82).

Existem *corpora* disponíveis para que novos estudos sejam feitos, como se pode encontrar no sítio <http://acdc.linguateca.pt/acesso/contabilizacao.html>, no qual há *corpora* da língua portuguesa de Portugal e do Brasil.

A base de textos estruturada, limpa e organizada permite que sejam realizados sobre ela diversos estudos estatísticos, para que sejam determinados os dados necessários ao desenvolvimento das ferramentas desejadas, como o processamento estatístico da linguagem natural (PLN). O PLN consiste na coleta de informações quantitativas sobre uma grande base de textos, para a determinação de parâmetros como a tabulação da ocorrência de palavras, seqüências de palavras, classes gramaticais, etc... (GASPERIN & LIMA, 2001).

Sobre um *corpus*, pode-se desenvolver vários estudos e captar informações sobre as palavras, a estrutura da linguagem e determinar uma próxima palavra com base nas estruturas das sentenças do *corpus*, quando a pesquisa foca a análise da frequência de co-ocorrência das palavras, o que torna possível conhecer as expressões formadas pelas palavras que, juntas,

mostram um significado único e diferente de seus significados isolados (GASPERIN & LIMA, 2001).

2.2.7 Modelos Ocultos de Markov

Para poder discorrer claramente sobre modelos ocultos de Markov, é necessário primeiro focar a atenção sobre um processo estocástico e, na seqüência, nas Cadeias de Markov ou Processos Discretos de Markov (WANG, 1997).

2.2.7.1 Cadeias de Markov

Um processo estocástico é constituído de um conjunto de variáveis randômicas (X) indexadas e cujo índice descreve uma rota sobre um outro conjunto determinado. Geralmente, este outro conjunto é representado pela variável tempo (t), com valores inteiros não-negativos. Este tipo de processo é utilizado para descrever como um sistema qualquer, $X(t)$, comporta-se com a evolução do tempo, ou seja, é possível conhecer o estado do sistema no parâmetro (t) desejado (TERCEIRO, 2003, NOGUEIRA, 2005). Para se determinar um processo estocástico, especificam-se as probabilidades da ocorrência dos eventos e as relações entre essas probabilidades em tempos diferentes.

Sob esta ótica, um processo pode ser chamado Markoviano se a probabilidade em um determinado instante condiciona-se somente ao estado presente, independentemente dos estados passados. Isto caracteriza o processo Markoviano como um processo estocástico simples, tornando-o apropriado às propostas de Engenharia, pois limita as interações matemáticas necessárias (WANG, 1997). Processos de Markov foram, inicialmente, aplicados no estudo da seqüência de letras encontradas na literatura russa, sendo desenvolvido para tal por Andrei Andreyevich Markov, na década de 60 (GASPERIN & LIMA, 2001, NOGUEIRA, 2005).

Desde sua introdução na Matemática, os métodos estocásticos com fontes Markovianas têm sido cada vez mais utilizados, devido a duas características do modelo: (1) os processos Markovianos oferecem uma rica estrutura matemática, formando bases teóricas aplicáveis em diversas áreas; (2) os modelos, quando bem desenvolvidos, respondem muito bem às mais diversas aplicações (RABINER, 1989).

Um fenômeno pode ser modelado por Markov se apresenta saídas observáveis que podem ser vistas como sinais, como a voz, por exemplo, sustentando a hipótese de que um sinal pode ser caracterizado como um processo paramétrico aleatório com parâmetros determináveis com precisão (RABINER, 1989).

Então, considerando-se um sistema descrito pelos N estados que os sinais podem assumir, esboça-se o diagrama da Figura 6.

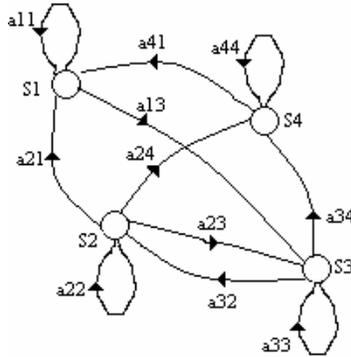


Figura 6: Uma cadeia de Markov com 4 estados, representados por S_1, \dots, S_4 com suas possíveis transições e probabilidades inerentes, representadas por $a_{11}, a_{12}, \dots, a_{44}$ (adaptada de RABINER, 1989).

Com base na Figura 6, pode-se observar que o sistema descrito permite que transições entre estados ocorram, inclusive retornando ao próprio estado, em tempos discretos, de acordo com o conjunto de probabilidades características ao sistema. Associando-se ao sistema o parâmetro tempo, $t = 1, 2, 3, \dots, T$, assume-se que em dado tempo t o estado atual é q_t .

O número de estados anteriores a serem considerados na determinação do próximo, determina a ordem da cadeia de Markov, ou seja, se forem considerados na análise o estado atual e seu predecessor, o processo Markoviano será dito de 1ª ordem, e pode ser descrito pela equação 1 (RABINER, 1989):

$$P[q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \dots] = P[q_t = S_j | q_{t-1} = S_i] \quad (\text{eq. 1})$$

Nota-se que a equação 1 indica a probabilidade q_t de ocorrer um estado S_j , baseia-se somente na probabilidade do anterior q_{t-1} .

Ainda, considera-se que o segundo elemento da equação 1 independe do tempo, logo as probabilidades do sistema descrevem-se pela matriz de transição, vista na equação 2 e denominada Matriz A :

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i], 1 \leq i, j \leq N \quad (\text{eq. 2})$$

E, toma-se por definição o estabelecido pelas equações 3 e 4:

$$a_{ij} \geq 0 \quad (\text{eq. 3})$$

$$\sum_{j=1}^N a_{ij} = 1 \quad (\text{eq.4})$$

No entanto, é comum em sistemas naturais que cada estado possa ser representado por um conjunto de possíveis observações. Tem-se, então, um processo em que os estados não são visíveis ao sistema, mas sim ocultos. Estes processos são denominados Modelos Ocultos ou Escondidos de Markov, mais citados como HMM (*Hidden Markov Models*) (RABINER, 1989).

2.2.7.2 Modelos Ocultos de Markov

Os HMM exigem a análise de dois processos estocásticos, um para determinar a probabilidade de transição entre os estados (Matriz A) e outro envolvendo as observações de saída para cada estado (Matriz B) (WANG, 1997, DIAS, 2000). Pode-se, então, afirmar que Modelos Ocultos de Markov são um duplo processo estocástico, no qual são considerados estados internos e símbolos observáveis externos. Os estados e suas transições são ocultos, ou seja, não podem ser visualizados, constituindo uma parte interna ao modelo correspondente a uma Cadeia de Markov (finita e tendo o estado como variável aleatória). Para tal cadeia, é determinada uma matriz cujos elementos representam a função densidade de probabilidade para as transições entre os estados (SCAVONE, 1996).

Para cada estado, associa-se um símbolo de saída e a seqüência desses símbolos constitui a parte observável do processo. Esses símbolos, justificando a definição de processo estocástico duplo, representam outra variável aleatória com uma função densidade de probabilidade própria para cada estado (SCAVONE, 1996).

Pode-se definir um HMM através dos parâmetros (DIAS, 2000, RABINER, 1989):

1. Um conjunto contendo os N estados $\{S_j\}$;
2. Uma matriz de transições $A = \{a_{ij}\}$;
3. O comprimento da seqüência observada (número de símbolos mostrados nela), representado por T ;
4. Uma seqüência de observações $O = \{o_1, o_2, \dots, o_T\}$;
5. O número M de possíveis observações para os estados, constituindo o conjunto $V = \{v_1, v_2, \dots, v_M\}$;
6. Uma matriz de ocorrências contendo as probabilidades de saída $B = \{b_{jk}\}$, na qual os elementos b_{jk} definem a probabilidade de ocorrência do símbolo k no estado j , tal que (equação 5):

$$b_{jk} = P[V_k(t) | q_t = S_j], 1 \leq j \leq N, 1 \leq k \leq M \quad (\text{eq.5})$$

7. Uma matriz inicial π contendo a probabilidade de cada estado iniciar uma seqüência, dada por:

$$\pi_j = P[q_1 = S_j], 1 \leq j \leq N \quad (\text{eq.6})$$

A exemplo dos elementos da matriz A , as condições para as matrizes B e π , são dadas pelas equações 7 e 8:

$$\sum_{k=1}^M b_{jk} = 1, \text{ sendo } b_{jk} \geq 0 \quad (\text{eq.7})$$

$$\sum_{i=1}^N \pi_i = 1, \text{ sendo } \pi_i \geq 0 \quad (\text{eq.8})$$

Nota-se que a especificação de um HMM envolve a determinação de dois conjuntos de parâmetros (N e M), a determinação dos símbolos observáveis e a especificação das probabilidades envolvidas (matrizes A , B e π). A notação utilizada para representar um conjunto completo de parâmetros que definem um HMM é representada pela equação 9:

$$\lambda = (A, B, \pi) \quad (\text{eq.9})$$

Definido o modelo de Markov, têm-se três tarefas a cumprir: (1) avaliação; (2) decodificação; e (3) aprendizado (SCAVONE, 1996).

A etapa de avaliação consiste em: conhecidos um modelo $\lambda = (A, B, \pi)$ e uma seqüência de T observações $O_{1...T}$, analisar qual a chance de que a dada seqüência tenha sido gerada a partir do conhecido modelo. Para isto, devem ser consideradas todas as seqüências de estados responsáveis pela saída $O_{1...T}$, como descreve a equação 10 (RABINER, 1989):

$$P(O_{1...t} = o_{1...t}) = \sum_{S=1}^t P(S_{1...t} = s_{1...t}) P(O_{1...t} = o_{1...t} | S_{1...t} = s_{1...t}) \quad (\text{eq.10})$$

Essa tarefa requer, além da análise das matrizes do modelo $\lambda = (A, B, \pi)$, a determinação de todos os caminhos possíveis para se atingir a seqüência $O_{1...T}$, sendo a quantidade de iterações necessárias diretamente proporcional a exponencial de T . Utilizando-se o cálculo recursivo em t , considerando que os cálculos envolvidos na determinação do instante t dependem somente das parcelas de $t-n$, sendo n a ordem do modelo, pode-se simplificar o processo (SCAVONE, 1996).

Definindo uma variável $\alpha_s(t)$ como sendo a probabilidade do sistema encontrar-se no estado s já tendo gerado uma seqüência parcial $O_{1...t}$, e assumindo que o modelo é de 1ª ordem, tem-se a equação 11 (SCAVONE, 1996).

$$P(O_{1...T} = o_{1...T}) = \sum_{s=1}^N \alpha_s(T), \quad (\text{eq.11})$$

onde a seqüência apresenta probabilidade definida pela equação 12:

$$\alpha_s(t) = P(O_{1...t} = o_{1...t} | S_t = s_t) \quad (\text{eq.12})$$

Desta forma, fazendo $t = 1$, obtém-se a equação 13:

$$\alpha_s(1) = \pi_s \cdot b_{sk | k=o_1}, \quad 1 \leq s \leq N \quad (\text{eq.13})$$

E para $t = 1...T-1$, a equação 14:

$$\alpha_j(t+1) = \left[\sum_{s=1}^N \alpha_s(t) a_{ij | i=s} \right] \cdot b_{jk | k=o_{t+1}}, \quad 1 \leq s \leq N \quad (\text{eq.14})$$

Para visualizar a diminuição dos cálculos envolvidos, deve-se observar que no instante $t = 1$, o sistema encontra-se no estado S_1 , com probabilidade π_1 e símbolo O_1 , que apresenta probabilidade $b_{1k|k=O_1}$. Com o incremento de t para 2, ocorre, também, a transição entre o estado anterior e o atual S_2 , com a probabilidade a_{12} , e símbolo de saída O_2 com probabilidade $b_{1k|k=O_2}$. O processo continua com os valores de t indo até T , envolvendo $2T.N^T$ iterações, sendo $(2T-1)N^T$ multiplicações e N^T-1 somas. Se for utilizado o cálculo recursivo em t , baseado nas equações 12 a 14, observa-se que as iterações necessárias para a determinação dos $\alpha_s(t)$, com $1 \leq s \leq N$ e $1 \leq t \leq T$, totalizam N^2T (RABINER, 1989).

Este cálculo recursivo é aplicado no algoritmo *forward*, utilizado na execução da primeira tarefa a ser realizada num modelo markoviano, ou seja, no processo de avaliação (SCAVONE, 1996, RABINER, 1989).

Na tarefa de decodificação, deve-se analisar, conhecido um modelo $\lambda = (A, B, \pi)$ e uma seqüência de observações $O_{1...T}$, qual a seqüência de estados mais provável, ou seja, determinar os caminhos mais prováveis que uma seqüência de estados percorreu para apresentar como saída a seqüência $O_{1...T}$ conhecida. As considerações matemáticas, neste caso, são semelhantes às feitas para a avaliação do modelo, com a diferença de não computar todos os caminhos percorridos com saída de símbolos iguais, mas somente o mais provável. Essas considerações são aplicadas no Algoritmo de Viterbi, utilizado neste passo (RABINER, 1989, PATEL, 1995, BORGES, 2004).

O algoritmo considera duas variáveis: $\delta_t(i)$, a qual indica a maior probabilidade encontrada ao percorrer um caminho com t observações e que apresenta como final o estado i ; e um vetor, $\psi_t(i)$, que armazena os estados percorridos até T (RABINER, 1989).

Considerando-se $t = 1$, tem-se as equações 15 e 16 (SCAVONE, 1996):

$$\delta_s(1) = \pi_s \cdot b_{sk|k=O_1}, 1 \leq s \leq N \quad (\text{eq.15})$$

$$\psi_s(1) = 0 \quad (\text{eq.16})$$

Para $2 \leq t \leq T$ e $1 \leq j \leq N$, obtêm-se as equações 17 e 18 (SCAVONE, 1996):

$$\delta_j(t) = \max_{1 \leq s \leq N} \left[\delta_s(t-1) \cdot \alpha_{ij|i=s} \right] \cdot b_{jk|k=o_t} \quad (\text{eq.17})$$

$$\psi_j(t) = \arg \max_{1 \leq s \leq N} \left[\delta_s(t-1) \cdot \alpha_{ij|i=s} \right] \quad (\text{eq.18})$$

Ao finalizar as iterações necessárias, o algoritmo considera o estabelecido pelas equações 19 e 20 (SCAVONE, 1996):

$$P^* = \max_{1 \leq s \leq N} [\delta_s(T)] \quad (\text{eq.19})$$

$$s^* = \arg \max_{1 \leq s \leq N} [\delta_s(T)] \quad (\text{eq.20})$$

E determina a melhor seqüência de estados pela equação 21 (SCAVONE, 1996):

$$s_{t-1}^* = \psi_t(s_t^*) \quad (\text{eq.21})$$

Para o aprendizado, conhecidas as estruturas do modelo e uma seqüência de observações gerada por ele, procura-se determinar os parâmetros $\lambda = (A, B, \pi)$ que maximizem as chances de tal seqüência ter sido gerada pelo modelo.

Segundo SCAVONE, (1996), esta é a tarefa mais complexa, pois não há uma solução analítica para ela. A fim de maximizar as probabilidades das seqüências de observações a serem analisadas pelo modelo, adequam-se os parâmetros das matrizes A , B , e π .

Uma ferramenta utilizada nesta tarefa é o algoritmo iterativo de Baum-Welch, o qual pode ser encontrado em vasta literatura (LEVINSON *et al.*, 1983, SCAVONE, 1996, RABINER, 1996, MARTINS, 1997, DIAS, 2000, BRITTO, 2001, EISNER, 2002, BORGES, 2004).

Como, nesta pesquisa, a execução dessa terceira tarefa foi realizada sem a utilização do Algoritmo de Baum-Welch, não se discorrerá sobre tal ferramenta, visto, também, ser vasta a literatura referente a este algoritmo.

2.2.7.3 Tipos de HMM

Genericamente, podem-se dividir os modelos ocultos de Markov em três grupos estruturais: ergóticos, seqüenciais e paralelos (BORGES, 2004).

Os modelos ergóticos ou sem restrição são aqueles em que todas as transições entre todos os estados do modelo são permitidas e um estado pode ser alcançado sequencialmente a qualquer outro ou num número finito de transições, como mostra a Figura 7.

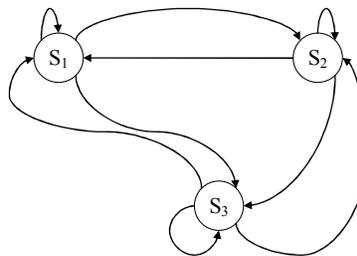


Figura 7: Modelo Ergótico, onde S_1 , S_2 e S_3 são os estados do modelo (adaptada de BRITTO *et al.*, 2001).

Os seqüenciais não permitem transições entre um estado qualquer e um anterior a ele, mas permite que estados sejam saltados, como mostra a Figura 8.

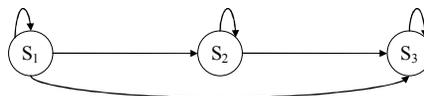


Figura 8: Modelo Seqüencial, onde S_1 , S_2 e S_3 são os estados do modelo (adaptada de BRITTO *et al.*, 2001).

Já nos modelos paralelos, permitem-se vários traçados e um ou mais estados podem ser saltados, sem que ocorra retorno a estados anteriores, como mostra a Figura 9.

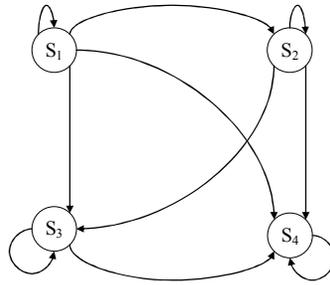


Figura 9: Modelo Paralelo, onde S_1 , S_2 , S_3 e S_4 são os estados do modelo (adaptada de BRITTO *et al.*, 2001).

Os HMM podem ainda ser classificados quanto à distribuição da probabilidade de observação de um símbolo em um determinado estado (Matriz B); sendo, nestes casos, classificados como discreto, contínuo e semi-contínuo (MARTINS, 1997, DIAS, 2000).

Um HMM pode ser definido como discreto se apresenta finitas (k) possibilidades de símbolos de saída e a probabilidade de ocorrer O_k no estado S_j é igual a b_{jk} , a qual é determinada sobre a distribuição do conjunto dos símbolos possíveis. Tal distribuição não exige um prévio conhecimento de sua forma para o levantamento do modelo do sinal analisado, sendo que cada observação possível liga-se a um valor discreto, o que torna o aprendizado simples, podendo utilizar uma amostra pequena de sinais se comparada aos outros tipos (contínuo e semi-contínuo) (BORGES, 2004).

A Figura 10 ilustra um HMM discreto que representa uma entre quatro cores possíveis de um sistema, no qual se observa que cada estado apresenta uma densidade de probabilidade diferente para as cores e que o número de possíveis símbolos de saída é finito (SINGH, 2006).

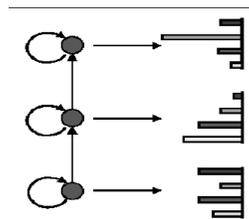


Figura 10: Modelo Oculto de Markov discreto e histograma de cores (adaptada de SINGH, 2006).

O HMM contínuo apresenta função densidade de probabilidade contínua (BORGES, 2004). Na Figura 11, vê-se que cada estado apresenta uma densidade associada ao símbolo possível de saída, neste caso, cores. Quando o sistema encontra-se em um determinado estado num instante t , determina-se um vetor a partir da função densidade do símbolo de saída. O HMM apresenta uma variável aleatória como saída, com valor contínuo para cada estado

(SINGH, 2006). As funções densidades dos símbolos são determinadas por Gaussianas (SCAVONE, 1996, MARTINS, 1997, BORGES, 2004).

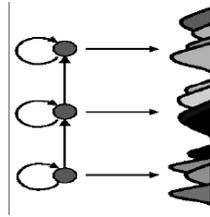


Figura 11: Modelo Oculto de Markov contínuo e distribuição da probabilidade de ocorrência das cores
(adaptada de SINGH, 2006).

O HMM semi-contínuo caracteriza-se como um modelo intermediário entre o discreto e o contínuo, tendo o conjunto de funções densidade de probabilidade semelhante para todos os estados, diferenciando-se unicamente pelos coeficientes de ponderação, ou seja, pela probabilidade de uma ocorrência O_t acontecer em um estado S , em um instante t (DIAS, 2000).

CAPÍTULO 3

METODOLOGIA

Neste capítulo, relata-se minuciosamente toda força-tarefa envolvida no desenvolvimento desta pesquisa, envolvendo as atividades executadas na confecção do *corpus* e seu processamento, desenvolvimento do algoritmo de predição, protocolo de testes, sua realização e análise dos resultados obtidos.

3.1 MATERIAIS E INFRA-ESTRUTURA

Por se tratar de desenvolvimento de *software*, esta pesquisa necessitou somente de um computador e aplicativos para programação (*Microsoft Visual Basic*[®], *Excel 2003*[®] e *Microsoft Visual C++ 6.0*[®]), além do necessário ao processamento do *corpus* (*WordSmith Text Converter*[®]) e do Teclado Virtual Livre.

Para a realização dos testes *in vivo*, contou-se com a infra-estrutura oferecida pelas instituições: Escola Estadual Ângelo Trevisan, localizada na Cidade de Curitiba, bairro Santa Felicidade; Escola Municipal Professora Genoveva Sicuro de Brito, no município de São José dos Pinhais; e, ainda, Escola de Ensino Especial Tia Vivian Marçal, localizada em Curitiba, que também auxiliou na tomada de decisão da estrutura de *corpus*.

3.2 DESENVOLVIMENTO

As etapas de desenvolvimento desta dissertação foram divididas em quatro grupos: Desenvolvimento do Corpus, Determinação dos Parâmetros Markovianos, Técnica de Predição e Testes. A seguir, descrevem-se tais grupos.

3.2.1 Desenvolvimento do *Corpus*

Conforme definição, um *corpus* constitui um conjunto de dados textuais e lingüísticos que serve como amostragem de uma população (dialetos, idiomas ou a linguagem como um

todo) de dimensão desconhecida. Tendo como público alvo desta pesquisa, crianças alfabetizadas da educação especial, que apresentem dificuldades de comunicação ou com problemas motores que dificultam o controle do teclado convencional, definiu-se como população a ser amostrada a linguagem e literatura infantis.

Visto ser grande o número de sítios infantis na Internet, os quais apresentam histórias feitas para e por crianças, lendas e contos, iniciou-se a tarefa de estruturação do *corpus* por selecionar tais sítios. Foram eleitos os *sites*:

<http://www1.uol.com.br/bibliaworld/paulodebs/hist001.htm>

<http://members.chello.nl/t.lage/anarcolivro/PDF/praxedes.pdf>

<http://www.releituras.com/>

<http://virtualbooks.terra.com.br/>

<http://sitededicas.uol.com.br/index.htm>

<http://www.graudez.com.br/litinf/textos.htm#Fabulas>

<http://www.contos.poesias.nom.br>

<http://www.usinadeletras.com.br/>

Fez-se busca utilizando a ferramenta Google™, pelas palavras chaves “*sites infantis*”, “*literatura infantil*”, “*contos*”, “*lendas brasileiras*” e “*histórias para crianças*”. Selecionados alguns sítios, aplicou-se o automatismo no arquivamento de textos, com o uso da ferramenta *Offline Downloader 3.5*, da *InternetSoft*®, o qual possui versão teste livre para uso, disponível no endereço eletrônico: <http://www.offlinedownloader.com/>. Com os conteúdos dos sítios desejados disponíveis *offline*, iniciou-se a seleção dos textos que comporiam o *corpus*.

Como os textos retirados da Internet podem apresentar formatações e conteúdo não interessantes ao objetivo do trabalho, é usual a aplicação de um filtro, anterior ao processamento do texto, para retirar os dados não relevantes, como o *Perl*® e o emulador *Cygwin*®, citados no item 2.1.5 desta dissertação. No entanto, realizou-se tal etapa sem automatismo, ou seja, o conteúdo dos sítios foi lido, desconsiderando-se cabeçalhos, propagandas e outros itens não referentes aos textos e, os textos eleitos, foram selecionados e transformados em arquivo com extensão txt (com auxílio do Bloco de Notas do Windows®).

Como não é objetivo deste trabalho uma análise lingüística, mas sim o uso do *corpus* para treinamento do aplicativo desenvolvido, era importante que o conteúdo-base apresentasse o mínimo possível de erros ortográficos e gramaticais, poucas figuras de linguagem e evitasse o uso exagerado de gírias.

Buscou-se formar um *corpus* que contivesse:

1. lendas brasileiras com personagens comuns à literatura infantil, como “Saci Pererê”, “Mula Sem Cabeça” e “Lobisomem”, entre outros;
2. textos de cultura popular, com histórias regionais como “O Negrinho do Pastoreio”;
3. diálogos;
4. textos religiosos de linguagem infantil que fizessem referência a Deus, Jesus e Maria, personagens conhecidos por crianças de diferentes religiões;
5. fábulas;
6. nomes de animais mais citados em histórias infantis, como leão, cachorro, macaco, gato, lobo, rato, boi, raposa, águia, burro, urso e pássaros;
7. expressões populares, como: “Nossa Senhora!” e “Meu Deus!”;
8. corpos da natureza: árvore, mar, cachoeira, rio, Sol, Lua, etc.;
9. objetos do dia-a-dia doméstico: vassoura, pano, prato, cama, cadeira, etc.;
10. cômodos da casa;
11. nomes de flores comuns, como margarida e rosa.

Para a determinação desses itens acompanharam-se, por um mês, as aulas de uma turma de alfabetização na Escola de Ensino Especial Tia Vivian Marçal, durante as quais observaram-se as palavras e materiais de apoio mais utilizados. Também foi considerada a lista elaborada por Pollo (2003), contendo 3635 palavras extraídas de literatura pré-escolar.

3.2.1.1 Tipologia do *Corpus*

Com os textos selecionados, definiu-se a tipologia do *corpus* a ser implementado. Para tal, tomaram-se como base os principais tipos citados na literatura (SARDINHA, 2004), obtendo-se um conjunto de dados com as características descritas a seguir.

Quanto ao modo, o *corpus* é do tipo escrito, pois constitui-se de textos escritos. Como os textos base foram retirados de sítios da Internet atuais, representando a linguagem utilizada no tempo corrente, o *corpus* é classificado como contemporâneo.

Quanto aos critérios de seleção, a coletânea de textos é dita de amostragem e estática; isso por não permitir alteração em seu tamanho, e, aquilo, por objetivar representar finitamente a linguagem como um todo, sendo, neste caso, textos infantis da língua

portuguesa (falada no Brasil). Por isso, ainda, o *corpus*, pode ser classificado quanto ao conteúdo, como especializado, por representar somente um tipo específico da linguagem.

Quanto à autoria, é de língua nativa e, quanto à finalidade, é para treinamento ou teste, tendo sido composto para servir como base de aprendizado de uma ferramenta de predição de palavras na produção de textos.

Definido o *corpus*, era necessário determinar sua representatividade, ou seja, o tamanho que ele deveria apresentar tal que justificasse, nesta pesquisa, sua função representativa.

Tal determinação deveria estar embasada no todo de quem o *corpus* será um representante. No entanto, não se conhece a dimensão da linguagem como um todo, o que impede a determinação exata do tamanho do conjunto representativo dela. Conhecem-se algumas características da língua portuguesa, a qual permite uma infinidade de figuras de linguagem e um vasto léxico. Logo, sobre tais afirmações, deduz-se que quanto maior o *corpus*, maiores as probabilidades da ocorrência de palavras com sentidos diferentes e de vocábulos de baixa frequência de uso.

Como o objetivo deste trabalho não é uma análise lingüística, mas o de treinamento de um algoritmo e, ainda, voltado para o público infantil, não se deseja a ocorrência de excessivas figuras de linguagem e de vocábulos de pouco uso (SARDINHA, 2004).

Na literatura referente, pouco se encontra sobre tamanhos de *corpus* ideal, mas uma classificação mais abrangente considera como pequeno um *corpus* com menos de 80 mil palavras (SARDINHA, 2004).

Conjuntos de textos que não objetivam estudos de um idioma como um todo, mas somente partes de uma língua, como o TeMário (PARDO & RINO, 2003), ECI-EE, FrasesPP e FrasesPB (<http://acdc.linguateca.pt/acesso/contabilizacao.html#natura>), possuem menos de 62 mil palavras, sendo que o FrasesPP apresenta pouco mais que 19 mil vocábulos. Não foram encontrados *corpora* com aplicação semelhante ao trabalho desenvolvido nesta pesquisa.

Desta forma, a magnitude média estipulada para o *corpus* foi de 40 mil palavras, e o número de textos estipulado em 100, a exemplo do TeMário (PARDO & RINO, 2003).

3.2.1.2 Organização do *Corpus*

Sobre o conjunto de textos pré-selecionados, elegeram-se os 100 que compusessem o *corpus* final, com 40.329 palavras, sendo 7066 palavras diferentes.

Para a organização do *corpus*, de forma a deixá-lo inteligível para futuras aplicações, criou-se uma pasta denominada *CORPUS*, na qual foram criadas duas sub-pastas: TEXTOS e TEXTOS ETIQUETADOS.

Na pasta *CORPUS*, há um arquivo chamado textos.doc contendo todos os 100 textos enumerados. Já na pasta TEXTOS encontram-se os 100 arquivos dos textos individuais salvos com o título do texto como nome do arquivo e extensão .txt.

Para cada um destes arquivos criou-se um adicional, com nome formado pelo título do texto acrescido da extensão .hea (do inglês *header* que significa cabeçalho), com a seguinte estrutura:

- < título >
- < fonte >
- < autor >
- < língua (idioma) >
- < tipo de texto >

Os dados constantes no arquivo cabeçalho de cada texto visa promover a facilitação de aplicações futuras do banco de textos em estudos lingüísticos ou outros que necessitem de uma base de textos infantis. O cabeçalho dos textos encontra-se no Anexo I.

3.2.1.3 Processamento do *Corpus*

A fim de preparar o conjunto de textos para sua aplicação no desenvolvimento desta pesquisa, necessitou-se processá-lo. Este processamento envolveu inicialmente a etiquetagem dos vocábulos que compõem os textos.

Esta tarefa foi realizada pela Internet, com auxílio do *Visual Interactive Syntax Learning* (VISL), o qual adiciona a cada palavra do texto sua classe gramatical como etiqueta principal e gênero, número, tempo verbal e outras características gramaticais, como sub-etiquetas, conforme Tabela 1.

O etiquetador foi utilizado na opção *flat structure* e com etiquetagem *morphological tagging*, por apresentar nesta configuração as classes gramaticais necessárias ao

desenvolvimento da técnica em questão. Os textos foram etiquetados individualmente e armazenados em arquivos com nome seguindo a estrutura: título_do_texto.txt, na pasta TEXTOS ETIQUETADOS.

Tabela 1: Significados das siglas indicadas pelo etiquetador VISL.

	SÍMBOLO	SIGNIFICADO
CLASSES DE PALAVRAS	N	Substantivo
	PROP	Nome Próprio
	SPEC	Especificadores (que, que, isto, esse, etc.)
	DET	Determinantes (Artigos, Demonstrativos, etc.)
	PERS	Pronome Pessoal
	ADJ	Adjetivo
	ADV	Advérbio
	V	Verbo
	NUM	Numeral
	KS	Conjunção Subordinada
	KC	Conjunção Coordenada
	IN	Interjeição
	EC	Elemento Composto
	INFLEXÕES	M
F		Feminino
S		Singular
P		Plural
NOM		Nominativo
ACC		Acusativo
PIV		Prepositivo
1		Primeira Pessoa
2		Segunda Pessoa
3		Terceira Pessoa
PR		Presente
IMPF		Pretérito Imperfeito
PS		Pretérito Perfeito Simples
MQP		Pretérito Mais que Perfeito
FUT		Futuro
COND		Condicional
IND		Indicativo
SUBJ		Subjetivo
IMP		Imperativo
VFIN		Verbo Finitivo
INF		Verbo Infinitivo
PCP		Particípio
GER		Gerúndio

Algumas palavras foram interpretadas pelo etiquetador como podendo assumir diferentes sentidos dentro da estrutura da sentença. Sempre que isso ocorreu, o etiquetador classificou a palavra conforme as suas possíveis ocorrências. Nesses casos, a etiquetagem teve que ser finalizada manualmente sendo realizada a análise sintática da sentença e decidida qual a classe gramatical correta da palavra em questão.

O etiquetador coloca cada palavra em uma linha, acompanhada de sua classificação. O mesmo ocorre para pontuações. O padrão da etiquetagem pode ser observado na amostra:

bom [bom] **ADJ M S**

dia [dia] **N M S**

para [para] **PRP**

todos [todo] <quant> **DET M P**

Na seqüência, os 100 textos etiquetados foram unidos num arquivo denominado “Concatenados.txt”, através da rotina denominada ConcatenaTextos(), desenvolvida na linguagem de programação *Microsoft Visual Basic for Application*[®], no ambiente do aplicativo Word[®]. A listagem da rotina encontra-se no Anexo II.

O ficheiro de texto concatenados.txt serviu como base para todo o processamento, desenvolvimento e treinamento da técnica de predição desenvolvida e descrita na seqüência deste capítulo.

3.2.2 Determinação dos Parâmetros Markovianos

A aplicação dos postulados dos modelos markovianos, no objetivo principal desse trabalho, a predição de palavras, requer o conhecimento dos resultados esperados com o seu uso. É esperada, na utilização de um simulador de teclado, a antecipação ao usuário das palavras que compõem o texto sendo por ele digitado. Encontraram-se, então, as seguintes situações:

1. quando o simulador de teclado acabou de ser iniciado ou uma nova frase começa a ser digitada, ou seja, o campo de digitação temporário está vazio;
2. quando já existe uma palavra digitada e se deve indicar ao usuário as próximas mais prováveis.

Haveria de se estipular, neste momento, quais seriam os estados, as ocorrências e a ordem do modelo.

Denota-se nos itens citados que as observações, neste estudo, referem-se às palavras, pois é o que transparece ao usuário. A fim de indicar ao usuário o conjunto de observações mais prováveis, estas devem estar ligadas entre si por uma estrutura que permita analisar as probabilidades delas acontecerem seqüencialmente.

Baseando-se nos trabalhos de Kuhn, Niemann & Schukat-Talamazzini, (1994), Zimmermann & Bunke (2004), Law & Chan, (2006), Clarkson, (1999) e Nakamura *et al.*, (1990) no idioma Inglês, Gustavii & Pettersson, (2003) para o Sueco, e Hunnicutt, Nozadze & Chikoidze, (2006), na língua russa, viu-se que dividir as palavras componentes do *corpus* em suas classes gramaticais permite que o processamento estatístico e a determinação do grupo seguinte de ocorrências sejam mais rápidos. Assim como nos estudos citados, este assume como sendo os estados markovianos as classes gramaticais anexadas pelo etiquetador, a cada palavra do *corpus*.

O modelo foi determinado como sendo de segunda ordem, as palavras sendo analisadas por bigramas, ou seja, em pares de classes gramaticais que ocorrem seqüencialmente no *corpus*, tal como nos trabalhos de Thede & Harper, (1999) e Zimmermann & Bunke, (2004).

Pontuações para finalização de sentença (ponto final, exclamação, interrogação e ponto-e-vírgula), indicaram sempre o fim de uma sentença, sendo a próxima palavra entendida como início de frase. Da mesma forma, outros símbolos como vírgula, dois pontos, travessão e reticências foram também interpretados como interruptores de oração. Esta identificação foi facilitada pelo etiquetador que reserva para cada um desses símbolos uma linha exclusiva.

Desta forma, os elementos do modelo de Markov aplicado nesta pesquisas são:

1. conjunto contendo os N estados $\{S_j\}$;
2. matriz de transições $A = \{a_{ij}\}$;
3. comprimento da seqüência observada (número de símbolos mostrados nela), representado por T ;
4. seqüência de observações $O = \{o_1, o_2, \dots, o_T\}$;
5. número M de possíveis observações para os estados, constituindo o conjunto $V = \{v_1, v_2, \dots, v_M\}$;

6. matriz de ocorrências contendo as probabilidades de saída $B = \{b_{jk}\}$, na qual os elementos b_{jk} definem a probabilidade de ocorrência do símbolo k no estado j , como descrito na equação 22:

$$b_{jk} = P\left[V_k(t) \mid q_t = S_j\right], 1 \leq j \leq N, 1 \leq k \leq M \quad (\text{eq.22})$$

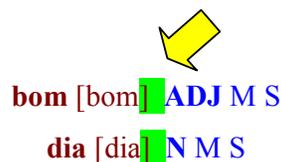
7. matriz inicial π contendo a probabilidade de cada estado iniciar uma seqüência, dada pela equação 23:

$$\pi_j = P[q_1 = S_j], 1 \leq j \leq N \quad (\text{eq.23})$$

Para a determinação dos parâmetros do modelo oculto de Markov, foram desenvolvidas rotinas na linguagem *Microsoft Visual Basic for Application*[®], porém, no ambiente do aplicativo Excel[®], também listadas no Anexo II.

Conforme as determinações do modelo, o conjunto de estados $\{S_j\}$ constitui-se das classes gramaticais ocorrentes no *corpus*, sendo N igual a 254, conforme resultado do processamento da rotina denominada *FrequenciaDeClasses()*. Esta, ao ser executada, inicialmente, checa a existência da planilha denominada “Contagem de Classes”. Caso negativo, é criada uma planilha em Excel[®] com esse nome, procede-se à leitura seqüencial por linha do ficheiro “Concatenados.txt” para a identificação das classes. Caso positivo, carrega-se a informação contida em tal planilha para a memória, numa variável chamada “ListaDeEstados” consistindo em matriz ($N \times 2$), onde os elementos da primeira coluna são do tipo *string* e os da segunda do tipo *integer*.

A identificação das classes resulta de busca, da direita para a esquerda, dos elementos “]”, ou seja, colchetes seguidos de espaço, como se indica na Figura 12 pela seta. Todo o elemento localizado à direita de “]” é uma classe gramatical.



bom [bom] ADJ M S
dia [dia] N M S

Figura 12: Texto Etiquetado (onde a seta amarela indica o símbolo indicativo do início da classe gramatical).

À medida que o ficheiro de texto é lido, toda vez que uma nova classe é identificada esta é inserida abaixo da última linha preenchida da coluna A, e na célula adjacente da coluna

B é armazenado o algarismo 1, indicando uma ocorrência. Durante a leitura do ficheiro, se classes já listadas na coluna A são identificadas no texto, incrementa-se o valor da respectiva célula de contagem na coluna B. Ao final, carregam-se os dados da planilha para a variável “ListaDeEstados”.

Para gerar a Matriz π , elaborou-se outra rotina denominada “CriaVetorPi()”. A partir da informação contida na variável “ListaDeEstados”, percorre-se o ficheiro “Concatenados.txt” buscando os inícios de sentença que ocorrem no início do ficheiro e nas linhas que sucedem as de pontuação. A cada ocorrência de início de sentença, identifica-se a classe em questão e computa-se a sua frequência. Ao término, somam-se todas as frequências e divide-se cada uma delas por este total, obtendo-se as probabilidades da Matriz π , de dimensão $(N \times I)$, que são salvas no ficheiro de texto “VetorPi.txt”.

Sequencialmente, criou-se outra rotina denominada CriaMatrizA() para a geração da Matriz A , de dimensão $(N \times N)$. Esta rotina percorre o *corpus* etiquetado (Concatenados.txt), localizando os pares de classe ocorrentes, computando-os em uma variável chamada “PseudoMatrizA”, que é uma matriz de dimensão $(N \times N)$ de elementos do tipo *single*. Cada elemento desta matriz é, inicialmente, zerado. À medida que os pares ocorrem, incrementa-se a intersecção da linha, que representa a classe gramatical antecessora do par, com a referente coluna, que representa a classe gramatical sucessora. Observar que a ordem das classes gramaticais, tanto nas linhas quanto nas colunas, é regida pela ordem das classes tabuladas na planilha. Ao final da leitura do *corpus*, somam-se os valores presentes em cada linha e divide-se o valor de cada elemento da linha pelo total da soma, obtendo-se as probabilidades de ocorrência da intersecção. Em seguida, esses dados são armazenados no ficheiro de texto “MatrizA.txt”.

De modo semelhante ao reconhecimento das classes gramaticais, ocorre o reconhecimento de cada palavra, buscando-se da esquerda para a direita a posição da primeira aparição de um espaço seguido por colchete de abertura (“ [“). Entende-se como palavra o que existe à esquerda deste espaço, desta forma obtém-se o conjunto $O = \{o_1, o_2, \dots, o_T\}$.

Para compor a matriz de ocorrências B , duas estruturas complementares de dados foram concebidas, uma chamada *Matriz de Busca* e outra BI .

A estrutura BI possui N linhas, cada qual com informações sobre a gama de palavras (símbolos observados) que ocorrem no *corpus* para cada classe (estados). A posição de cada linha está associada a uma classe, na mesma seqüência tabulada na planilha “Contagem das

Classes”. A Figura 13 detalha a estrutura da Matriz BI . A informação para cada classe (linha) da Matriz BI segue a estrutura $\{p, o_0, o_1, o_2, \dots, o_p\}$, na qual:

p informa o número de palavras observadas no *corpus* para a respectiva classe; e, o_i é a probabilidade com que cada palavra observada em uma classe ocorre ($0 < i < p$).

A posição de cada linha está associada a uma classe na mesma ordem da planilha “Contagem de Classes”

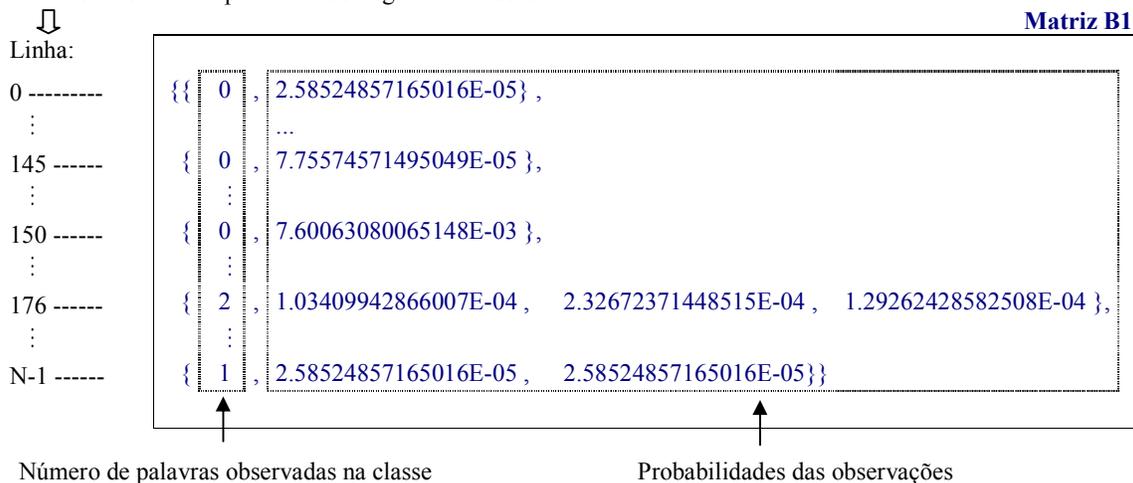


Figura 13: Estrutura da Matriz BI.

A *Matriz de Busca* possui tantas linhas quanto o número de diferentes palavras observadas no *corpus*. Cada linha traz informação referente a cada palavra segundo a seguinte estrutura {"palavra", número de classes em notação base zero na qual ela é observada, pares de dados}, na qual os pares de dados referem-se a: {índice da linha da classe em questão em BI , posição da probabilidade de observação da palavra na classe na linha da Matriz BI }. O número de pares de dados da estrutura da linha da Matriz de Busca é concordante ao número de classes em que a palavra é observada. O exemplo a seguir esclarece a estrutura da *Matriz de Busca*:

```

{"a", 2, 0, 0, 150, 0, 176, 2},
...
{"abacaxi", 0, 145, 0},
...
{"de", 0, 176, 1},
...

```

No exemplo está hipoteticamente sendo informado que a palavra “a” manifesta-se como símbolo observado em 3 classes, visto que o número que sucede a palavra informada é

o 2 e o sistema de contagem ser de base zero. Na seqüência, são informados os pares classe-probabilidade, cada qual dizendo que a palavra em questão manifesta-se como observação de uma determinada classe, com uma dada probabilidade. O primeiro par indica que a palavra “a” apresentou uma observação na classe de índice 0 (primeira linha da Matriz *BI*), com a probabilidade explicitada na posição de índice 0 (a primeira posição) na parte que informa as probabilidades das observações da Matriz *BI*. O segundo par indica que foi observada na classe de índice 150 (linha 151 da Matriz *BI*), com probabilidade expressa na posição 0. O terceiro na classe índice 176, com probabilidade na posição 2.

No exemplo, a palavra “abacaxi” foi observada apenas para uma classe, a com índice de linha 145, com probabilidade dada na posição 0. Já, a palavra “de” foi observada na classe 176, com probabilidade expressa na posição de índice 1 da respectiva linha de classe, na parte das probabilidades de observações da Matriz *BI*.

Notar que, no exemplo, as palavras “a” e “de” aparecem como observações de uma mesma classe, a de índice 176, porém, cada qual tem sua probabilidade expressa por posição diferente na linha da classe, em *BI*. Isto denotar, também, que as palavras “a” e “de” são preposições, ou seja, são observações da mesma classe gramatical, cada uma com sua probabilidade de ocorrência.

A montagem da *Matriz de Busca* e da Matriz *BI* é resultado da execução da rotina *Cria_MatrizBI_MatrizDeBusca()*. Esta rotina, inicialmente cria um esqueleto da Matriz *BI*, ou seja, um vetor vazio composto de *N* linhas, uma para cada classe da planilha “Contagem de Classes”, por 2 colunas de *strings*. E, como meio para a posterior criação *Matriz de Busca*, declara-se uma matriz denominada *B2*, formada por *N* linhas de *strings*. A rotina segue, então, com a leitura do ficheiro de texto “Concatenados.txt”. À medida que o ficheiro é lido, identifica-se em cada linha de texto a existência do caractere identificador de classe. Caso exista, filtra-se a palavra e a sua classe. Conta-se e armazena-se o número destas ocorrências. Em seguida, verifica-se a qual índice de classe gramatical as informações filtradas correspondem.

Checa-se, então, se a palavra reconhecida já foi registrada na Matriz *B2*, na correspondente linha de classe identificada. Caso não tenha sido, a palavra é inserida na linha de índice correspondente da Matriz *B2*. Na Matriz *BI*, a primeira coluna da linha de mesmo índice, tem seu valor incrementado. A segunda coluna da Matriz *BI* é redimensionada com a quantidade de ocorrências na classe detectada, e seu último elemento recebe o valor unitário.

Quando o par de informações filtradas de “Concatenados.txt” é detectado em $B2$, ou seja, uma dada palavra já presente na linha de sua classe em $B2$, apenas incrementa-se o valor da segunda coluna de $B1$, respeitando-se a posição referente à palavra na mesma linha em $B2$.

Ao terminar a leitura do ficheiro de texto tem-se armazenado o número de palavras do *corpus*. Então, ocorre a substituição de cada contagem da segunda coluna de $B1$ pelo resultado da divisão destes valores pelo número total de palavras do *corpus*, obtendo-se a probabilidade de ocorrência das palavras.

Finalizado este processo, parte-se para a estrutura da *Matriz de Busca*. Faz-se a leitura de cada elemento da Matriz $B2$, inserindo cada palavra diferente encontrada nestes elementos, em uma nova linha da *Matriz de Busca*. Cada linha desta matriz traz junto de cada palavra mais dois elementos de tipos distintos, sendo um para indicar a quantidade de classes em que a palavra ocorre e outro formado por pares de informação contendo o índice da classe de ocorrência da palavra e a posição no elemento da segunda coluna da Matriz $B1$.

Finalizando, geram-se os ficheiros de texto das referidas matrizes, denominados “ $B1.txt$ ” e “*Matriz de Busca.txt*”.

Com todas as informações necessárias ao processamento da predição recuperáveis, deu-se a programação das rotinas de predição de palavra, propriamente ditas.

3.2.3 Algoritmo de Predição

Como descrito, os dados manipulados são palavras as quais são divididas em classes gramaticais, sendo estas os estados markovianos e aquelas as ocorrências. A partir do momento em que uma seqüência de palavras é iniciada, um caminho passa a ser traçado, contendo os estados mais prováveis de ocorrerem nas posições seqüenciais. Isso se deve ao fato de uma mesma palavra poder pertencer a mais de uma classe gramatical. Uma palavra, ao ser digitada, tem carregadas suas características gramaticais e, conforme sua posição na cadeia, as classes possíveis de ocuparem a posição seguinte são determinadas. No entanto, esta palavra pode ser parcialmente apagada pelo usuário, e suas características gramaticais devem, neste momento, ser novamente buscadas e consideradas. Assim como a última palavra digitada pode ser totalmente apagada, sendo, então, necessárias as informações da cadeia anterior a esta palavra. Isto requer armazenamento dos caminhos prováveis de terem sido percorridos pela seqüência, até a posição em que ela se encontra.

Para o processamento da cadeia markoviana, é necessário delinear as situações possíveis de acontecer em função das ações do usuário do aplicativo, sendo elas:

1. verificação do tamanho da seqüência;
2. seleção de uma palavra da lista de predição;
3. digitação de uma palavra encontrada na base de textos e
4. digitação de uma palavra não encontrada no *corpus*.

Cada uma dessas situações apresenta particularidades que são detalhadas na seqüência.

3.2.3.1 Predição das Palavras Iniciais de uma Sentença

Em termos de predição de palavras, fundem-se aqui os termos sentença e seqüência, pois cada elemento desta equivale a uma palavra daquela.

Quando o início de uma sentença ocorre, o algoritmo de predição deve percebê-lo e indicar as palavras mais prováveis de iniciarem a sentença. É considerado início de sentença quando a linha de digitação indicada na Figura 14 por uma seta estiver vazia, retornando uma resposta correspondente a um conjunto vazio; e, também, caso a linha de digitação contenha algum texto, faz-se uma varredura da direita para esquerda para checar se um sinal ortográfico interpretado como fim de sentença (citados no item 3.2.2 deste Capítulo) acabou de ocorrer.

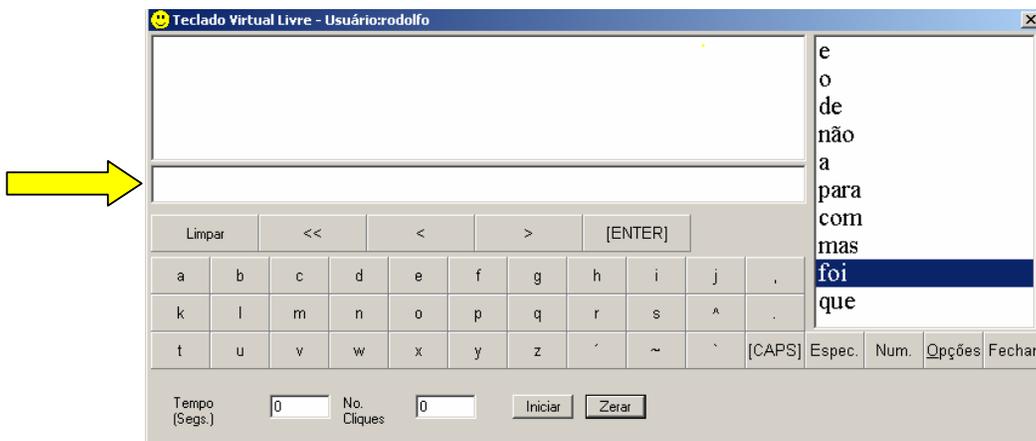


Figura 14: Indicação da Linha de Digitação do Aplicativo Teclado Virtual Livre por uma Seta.

Nos casos em que uma situação de início é detectada, a Matriz π deve ser envolvida nas determinações matemáticas. O algoritmo, então, realiza a consideração da equação 24:

$$\delta_s(i) = \pi_s \cdot b_{sk} \quad (\text{eq.24})$$

S corresponde às linhas das Matrizes π e B1 e k à posição da probabilidade na Matriz B1.

Neste momento, a matriz de predição começa a ser preenchida. Esta matriz é composta por 100 estruturas “mtzPred”, como ilustra a Figura 15.

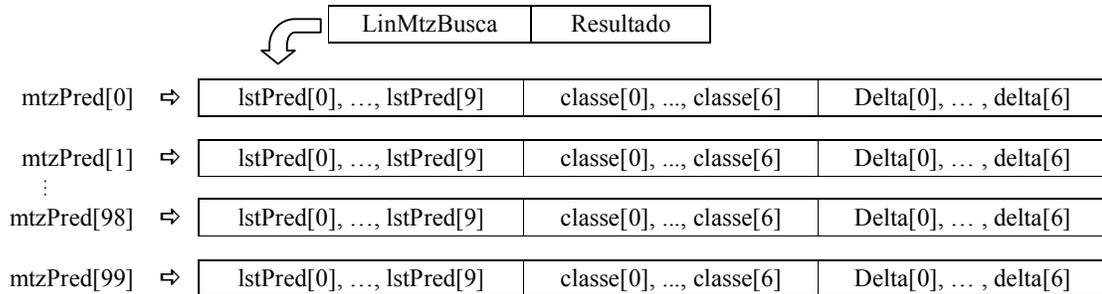


Figura 15: Estrutura da matriz de predição que armazena os dados necessários no algoritmo.

Serão preenchidas com informação de predição, tantas estruturas “mtzPred” conforme a quantidade de palavras na linha de digitação do Teclado Virtual Livre.

No programa de predição, utilizou-se uma variável tipo *integer*, nomeada “indPred”, referida aqui com índice de predição, para indexar os elementos “mtzPred” da matriz de predição. O índice zero caracteriza a situação em que nenhuma palavra foi digitada.

A estrutura “mtzPred”, esquematizada na Figura 16, é composta por:

1. uma lista de predição, “lstPred”, usada para armazenar até dez pares de dados correlacionados às palavras a serem exibidas na lista de predição. Estes pares são compostos por duas variáveis, uma variável do tipo *integer*, a “LinMtzBusca” usada para guardar o índice da linha em que a palavra está situada na Matriz de Busca, e outra do tipo *double*, “Resultado”, que serve para armazenar a maior probabilidade de ocorrência desta palavra;
2. um vetor com sete elementos tipo *integer*, denominado “Classe”, para armazenar as classes da palavra presente na linha de digitação cuja posição corresponda ao índice de predição;
3. um vetor com sete elementos tipo *double*, denominado “Delta”, para armazenar as probabilidades parciais dos melhores caminhos para se atingir a palavra em questão.

Ressalta-se que o número de sete elementos nos vetores descritos decorre da quantidade máxima de diferentes classes gramaticais em que uma mesma palavra foi etiquetada no *corpus* deste projeto.

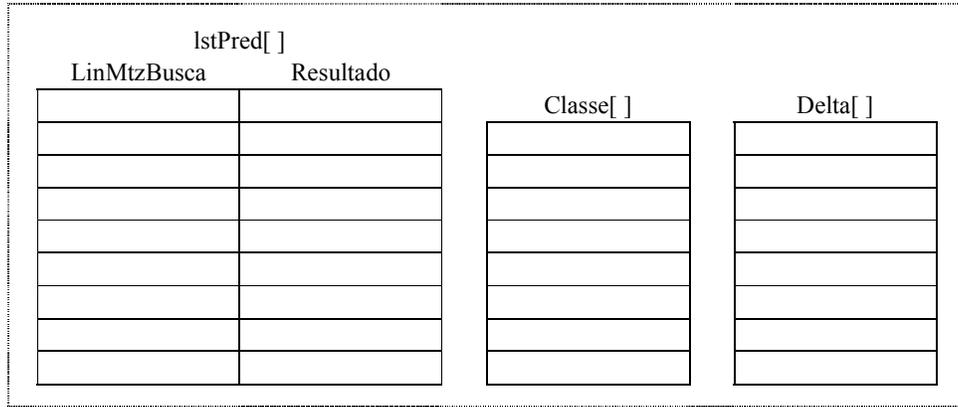


Figura 16: Estrutura de um elemento “mtzPred”.

Como na situação de início de sentença não há palavra a ser detectada na linha de digitação, o algoritmo recorre à Matriz π para eleger as dez observações mais prováveis que serão ordenadas através da sub-rotina “Maiores”, que se encarrega de registrar e manter classificados em ordem decrescente os maiores valores registrados numa lista de predição de um elemento da matriz de predição, cuja indexação depende do valor armazenado no índice de predição no momento em que “Maiores” é chamada. Ela compara o parâmetro “valor” aos 10 valores registrados na lista de predição e, quando este suplanta algum deles, ela o inclui na listagem, descartando o menor deles.

Desta forma, obtém-se o índice das dez palavras mais prováveis de iniciarem uma sentença, as quais são mostradas no campo de predição do Teclado Virtual Livre, considerando uma base de textos e as classes gramaticais. Nota-se que a determinação dessas palavras, neste caso, é constante para uma mesma base de textos.

3.2.3.2 Predição da Palavra Seguinte a uma Conhecida

Sempre que se der entrada a uma palavra na linha de digitação do aplicativo, seja através da lista de predição ou da digitação da palavra completa, o algoritmo deverá

identificar esta palavra na *Matriz de Busca* e determinar quais as dez palavras mais prováveis de ocorrerem na seqüência.

Previamente à identificação, checka-se se a palavra inicia ou não uma sentença. Identifica-se a palavra, recuperando, da *Matriz de Busca*, as informações de suas classes, armazenando-as no vetor “Classe”, descrito no item anterior. Preenchem-se, também, os elementos do vetor “Delta” com o resultado obtido na aplicação das determinações de Viterbi.

Sendo identificado um início de sentença, este vetor é preenchido com os resultados obtidos pela equação 24. Caso contrário, os valores usados no preenchimento do vetor resultam da equação 25:

$$\delta_j(t) = \max_{1 \leq s \leq N} [\delta_s(t-1) \cdot \alpha_{sj}] \cdot b_{jk} \quad (\text{eq. 25})$$

onde:

$\delta_s(t-1)$ o valor conhecido da palavra anterior

α_{sj} a probabilidade da classe gramatical seguinte (Matriz A)

b_{jk} a probabilidade de um símbolo acontecer em um dado estado.

Após incrementar o índice de predição, aplica-se a sub-rotina “Maiores” para ordenação os dez maiores $\delta_j(t)$ calculados, indicando-se ao usuário as dez palavras mais prováveis de ocorrerem.

3.2.3.3 Busca da Palavra Sendo Digitada pelo Usuário

Enquanto se digita uma palavra, o algoritmo de predição procura na *Matriz de Busca* palavras iniciadas pelo trecho digitado pelo usuário, sem ser caso sensitivo. A cada coincidência encontrada, as considerações de Viterbi são realizadas, semelhantemente ao descrito no item anterior, e determinadas as dez palavras mais prováveis. Há casos em que o trecho digitado coincide com menos de dez palavras do *corpus*. Nesses casos, a lista de predição indica menos de dez palavras ou até nenhuma.

Caso a palavra sob digitação seja início de sentença (primeira palavra), a busca ocorre somente nas palavras cujas classes pertençam à Matriz π .

3.2.3.4 Busca da Palavra Digitada pelo Usuário

Caso seja dado um comando de avançar após a digitação, ou seja, uma palavra foi digitada na íntegra pelo usuário, o algoritmo preverá as próximas palavras mais prováveis como descrito no fluxograma da Figura 17.

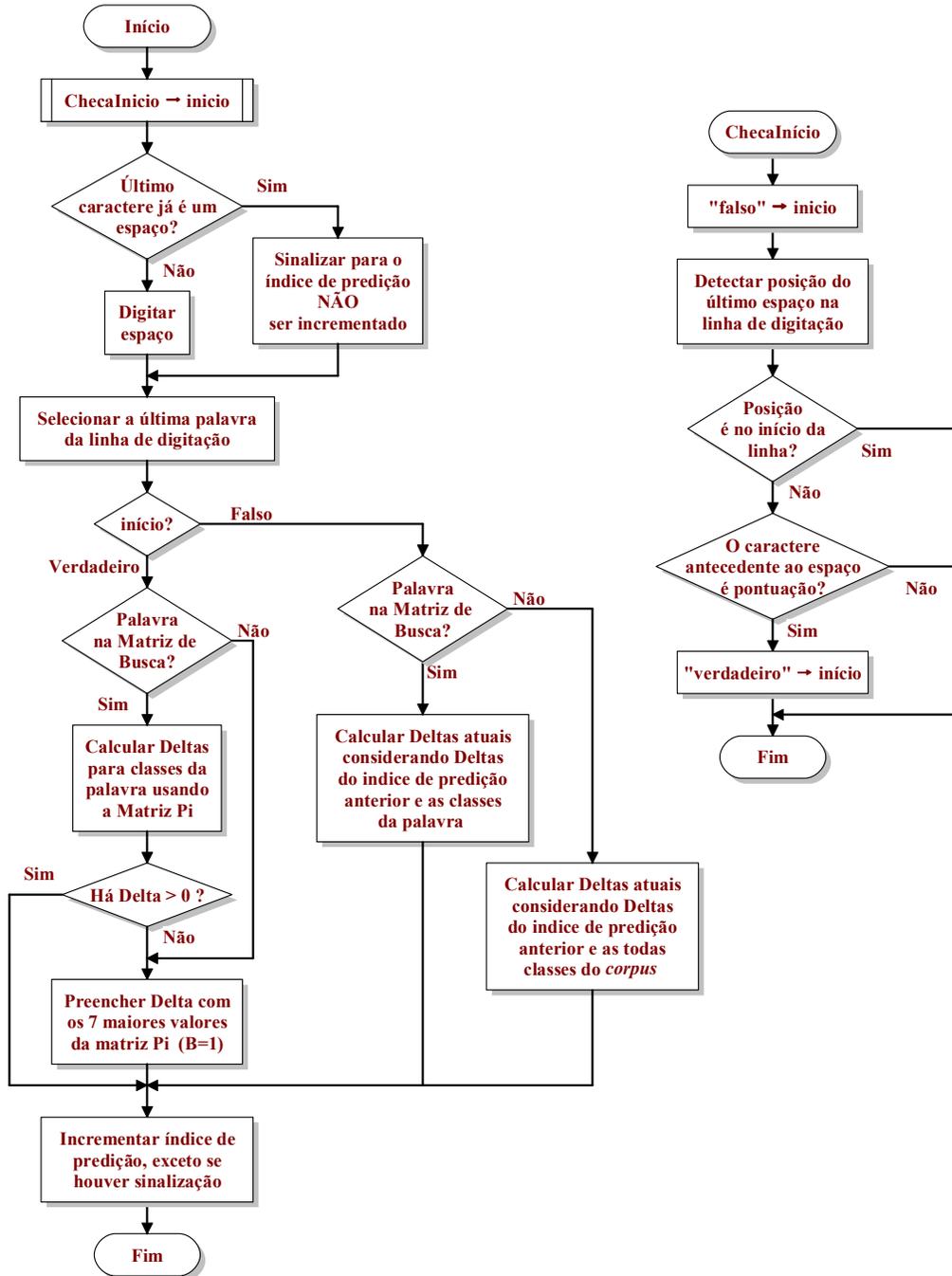


Figura 17: Fluxograma do Algoritmo de Predição ao Comando de Avançar.

No fluxograma da Figura 17, observa-se que ao ser digitada uma palavra, esta pode ou não pertencer ao *corpus*. Caso a palavra seja encontrada na *Matriz de Busca*, pertencendo ao *corpus*, o processo de predição ocorre semelhantemente ao descrito no item 3.2.3.2. Do contrário, considera-se que todas as classes gramaticais têm a mesma probabilidade de representar a palavra desconhecida e assume-se a probabilidade b_{jk} igual a um. Caso a palavra desconhecida inicie uma sentença, considera-se que ela pertença às classes com maior probabilidade na *Matriz* π .

3.2.4 Protocolo de Testes

Foram realizados testes para verificar a funcionalidade, funcionamento, fidelidade da técnica de predição a sua base de textos, assim como sua coerência quando do uso de palavras repetidas e, enfim, testes de aplicabilidade. Os testes foram realizados mediante aprovação do Comitê de Ética da Pontifícia Universidade Católica do Paraná, conforme carta presente no Anexo VI.

A funcionalidade do Teclado Virtual Livre foi verificada com a digitação contínua, por duas horas, em três dias seguidos, de textos aleatórios copiados do sítio <http://www.usinadeletras.com.br/>.

As ferramentas do aplicativo, como varredura, auto-clique em diferentes velocidades, tamanhos, disposições de letras, teclas especiais, inclusão e exclusão de usuários, foram testadas no início do uso do aplicativo, durante o uso, aproximadamente a cada 10 min e ao fim das 2 h.

O funcionamento foi avaliado com a digitação do texto “A Lagarta”, por duas pessoas, sendo um adulto e uma criança alfabetizada, com e sem predição.

A fidelidade à base de textos foi realizada com a digitação de partes do *corpus* determinada por sorteio realizado como se descreve.

Foram impressos números de 1 a 100 correspondentes à seqüência dos textos presentes no arquivo “concatenados.txt” e mais três cartões com os textos: (1) parágrafos 1, 2 e 3; (2) dois parágrafos centrais do texto; e (3) parágrafo final do texto.

Era, então, sorteado um número (de 1 a 100) e verificado de que texto se tratava. Caso o texto fosse pequeno (até 2 parágrafos), o mesmo era digitado na íntegra, senão, um segundo cartão era sorteado, determinando qual ou quais parágrafos do texto deveriam ser digitados.

A fim de verificar a consistência textual e a coerência da predição, selecionou-se um texto com palavras repetidas em posições distintas dentro das sentenças, sendo eleito o texto “As Letrinhas” e o mesmo foi digitado com e sem a predição.

Objetivando comparar a técnica desenvolvida a ferramentas semelhantes encontradas na literatura citada neste trabalho, testou-se o aplicativo através da digitação de um texto selecionado na Internet por uma criança de 10 anos. Este teste focava a contagem do número de palavras que eram preditas, ou seja, o número de palavras acertadas pelo algoritmo de predição.

O texto em questão foi “O Menino e a Moeda”, com 279 palavras, extraído do sítio com endereço eletrônico: <http://www.contos.poesias.nom.br/>. Contaram-se as palavras preditas sem nenhum clique, com um e até com oito cliques, relacionadas à quantidade de letras de cada uma delas. Desta maneira, é possível obter um percentual de acerto da próxima palavra no texto alcançado pela técnica desenvolvida.

Para verificar a aplicabilidade e vantagem do uso da técnica de predição no simulador de teclado adotado, foram realizados testes com um grupo de 41 crianças de diferentes realidades cognitivas. Tais testes consistiram no uso do computador para produção de um texto com auxílio do Teclado Virtual Livre. Era solicitado ao usuário que digitasse o texto “A Lagarta”, presente no Anexo III, por duas vezes, sendo uma sem a predição de palavras e outra com.

Foram anotados o número de cliques e o tempo (segundos) gasto para a execução das tarefas. Todos os testes foram acompanhados de uma pedagoga, fonoaudióloga ou fisioterapeuta da instituição onde se realizaram os testes.

A todas as crianças participantes foi inicialmente entregue um termo de consentimento a ser assinado pelos pais ou responsáveis para a realização dos testes.

Ao final de cada digitação foi entregue a cada criança o questionário presente no modelo do Anexo IV.

3.2.4.1 Tamanho da Amostra e Grupos Controle dos Testes de Aplicabilidade

A amostra foi composta de quatro grupos distintos, sendo três deles formados pelos alunos das três turmas de 4ª série da Escola Municipal Professora Genoveva Sicuro de Brito, no município de São José dos Pinhais e o outro grupo formado por alunos alfabetizados e com habilidade cognitiva suficiente para utilizar o aplicativo e digitar um texto, da Escola de

Ensino Especial Tia Vivian Marçal. Os quatro grupos contêm 11, 11, 12 e 07 alunos, sendo o grupo de 07 alunos o grupo da Escola Tia Vivian Marçal.

O tamanho da amostra determinou-se pelo número de crianças autorizadas pelos responsáveis a realizar os testes e que estiveram presentes nas aulas de informática em que os testes foram realizados.

Foi considerado grupo de controle o conjunto de resultados obtidos sem o uso da técnica de predição. Desta forma, os resultados obtidos com a aplicação da técnica são comparados ao grupo controle e os valores de redução ou incremento de tempo e quantidade de cliques determinados percentualmente.

3.2.5 Análise dos Resultados

Os parâmetros tempo e número de cliques disponibilizados pelo próprio aplicativo, durante seu uso, foram analisados comparativamente, obtendo-se um percentual de redução ou incremento dos parâmetros com e sem o uso da predição.

Na análise da aplicabilidade não houve comparações entre os dados obtidos em cada escola, ou seja, os valores foram determinados e analisados dentro de cada grupo da amostra.

A análise baseou-se, também, em comparação direta dos resultados obtidos, determinando-se, em percentual, a redução ou aumento dos parâmetros envolvidos. E a comparação de valores obtidos com os encontrados na literatura referente. Também foi analisado o desvio padrão obtido em cada grupo.

O questionário do Anexo IV, apresentado aos voluntários, auxiliou na determinação da dificuldade em digitar o texto, permitindo, através da opinião dos voluntários uma análise do uso do aplicativo.

3.2.6 Locais de Realização dos Testes de Aplicabilidade

Os testes foram realizados em dois locais:

1. Escola Municipal Professora Genoveva Sicuro de Brito, no município de São José dos Pinhais - PR;
2. Escola de Ensino Especial Tia Vivian Marçal, localizada na cidade de Curitiba - PR, bairro Mercês.

3.2.7 Planos de Recrutamento dos Testes de Aplicabilidade

Os critérios de exclusão para a determinação do grupo de voluntários foram:

1. não ser alfabetizado;
2. estar impossibilitado de controlar o computador através do periférico *mouse* convencional ou dispositivos auxiliares;
3. não estar participando de aulas de informática dentro de sua respectiva instituição de ensino.

Os voluntários da Escola Tia Vivian Marçal foram selecionados através dos seguintes critérios:

1. serem alunos regulares da Escola;
2. apresentarem boa frequência às aulas (acima de 70%);
3. apresentarem dificuldade na comunicação natural (fala, gestos ou sinais);
4. serem alunos das aulas de informática oferecida pela escola, ou seja, estarem adaptados ao uso do computador;
5. serem alfabetizados.

As crianças da Escola Municipal Professora Genoveva Sicuro de Brito que participaram da pesquisa apresentaram as seguintes características:

1. processo final de alfabetização, ou seja, cursarem 3^a ou 4^a série;
2. participação nas aulas de informática.

O grupo componente da amostra de testes é diversificado, ou seja, composto por crianças de níveis cognitivos diferentes, para que se possa analisar a aplicabilidade da técnica de predição para todos, ou para que se perceba a necessidade de adaptação da mesma para algum grupo em especial. Enfatizando-se que não serão comparados os dados obtidos pelos grupos de cada escola, sendo que estas apresentam realidades distintas.

CAPÍTULO 4

RESULTADOS

Este tomo apresenta-se em três subdivisões que elucidam os resultados obtidos neste trabalho dentro de seus universos. Primeiramente, faz-se referência ao *corpus* desenvolvido. Na seqüência, descreve-se o algoritmo de predição e as alterações realizadas no Teclado Virtual Livre. Finalizando o Capítulo, apresentam-se os resultados obtidos com os testes realizados na utilização do aplicativo por voluntários.

4.1 CORPUS

O primeiro resultado obtido no decorrer da pesquisa foi o banco de textos armazenado. Composto de textos infantis escritos, em Língua Portuguesa, sem excessivas figuras de linguagem e boa qualidade gramatical o *corpus* contém 40.329 palavras, sendo 7066 diferentes entre si.

O *corpus* apresenta-se organizado em pastas e sub-pastas, sendo arquivados individualmente os textos e acompanhados de um arquivo cabeçalho com as características do mesmo, além de sua fonte.

4.2 ALGORITMO DE PREDIÇÃO

O algoritmo de predição foi o principal resultado obtido nesta pesquisa. Seu desenvolvimento transcorreu sob enfoque de ordem inversa, iniciando as análises a partir de uma matriz que computa cada observação (palavra) detectada no *corpus*, associando aos seus estados (classes gramaticais).

O algoritmo não computa toda a Matriz de Observação, ou seja, não é preciso correlacionar o estado a cada observação. Isto ocorreria, num algoritmo normal de Viterbi (RABINER, 1989, BORGES, 2004, PATEL, 1995), através da Matriz de Observação (Matriz *B*). No algoritmo desenvolvido, optou-se por computar apenas as probabilidades não nulas,

porém, de maneira inversa, ou seja, os estados em função das observações e não o contrário como de costume nos apontamentos de trabalhos semelhantes (RABINER, 1989).

Seu funcionamento ininterrupto não demonstrou problemas como travamento, desligamento espontâneo ou conflito com outros aplicativos.

4.3 TESTES

Inicialmente, realizaram-se testes de funcionalidade do aplicativo, a fim de verificar o funcionamento do mesmo apesar das alterações realizadas em seu código fonte.

Verificaram-se funcionamento das ferramentas de varredura e auto-clique, com digitação de textos aleatórios retirados da literatura infantil, do sítio <http://www.usinadeletras.com.br/>.

Também os tamanhos do aplicativo e a criação de novos usuários foram conferidos, sendo verificados em diversos pontos de utilização, no início do funcionamento do aplicativo, durante o uso e depois de 10 min de uso. Estando todas as ferramentas com sua funcionalidade garantida, partiu-se para a verificação do funcionamento e fidelidade da predição desenvolvida.

O texto escolhido para o teste inicial de funcionamento do aplicativo foi o de título “A Lagarta”, eleito por ser simples, conter palavras longas e pontuações diferentes, como vírgula, ponto final e exclamação. Este texto, assim como os demais utilizados nos testes, encontram-se no Anexo III.

Verificando a fidelidade do algoritmo desenvolvido foram sorteados três trechos do *corpus* para serem digitados, tal como descrito no capítulo da Metodologia. Primeiramente, sorteou-se o 36 que por ser pequeno, foi digitado na íntegra necessitando para tal de 278 cliques e 602 s. O texto contém 103 palavras e 551 caracteres, considerando-se os espaços. Na seqüência, foi sorteado o texto 98 e o último parágrafo deste. Foram necessários 273 cliques e 575 s para a execução desta tarefa, sendo que os parágrafos em questão somam 79 palavras e 478 caracteres (com espaços). Finalizando esta etapa, sortearam-se o texto 54 e os parágrafos 1, 2 e 3, do mesmo, totalizando 376 caracteres (com espaço) e 67 palavras. Foram necessários 193 cliques e 433 s.

No sítio <http://www.usinadeletras.com.br/>, escolheu-se um texto com frases curtas e repetição para testar a coerência da busca de palavras, selecionando-se o texto “As Letrinhas”,

presente no Anexo III, e digitou-se o mesmo com e sem a predição, obtendo-se os resultados presentes na Tabela 2. O texto contém 101 palavras e 404 caracteres com espaços.

Tabela 2: Resultados obtidos no teste de coerência na busca de palavras.

	Sem predição		Com predição	
	Cliques	Tempo de Execução (s)	Cliques	Tempo de Execução (s)
"As Letrinhas"	418	468	330	564

Antes de apresentar o aplicativo aos voluntários, conforme protocolo de testes descritos na Metodologia, realizou-se teste individual com uma criança de 10 anos, cursando a 4ª série. Solicitou-se a essa criança, que já conhecia o aplicativo, que digitasse o texto “A Lagarta” por duas vezes, uma primeira com a predição e outra sem.

O texto possui 40 palavras e 237 caracteres, considerados os espaços entre palavras. O Teclado Virtual Livre foi configurado no formato ABCDE para a disposição de letras e no tamanho 3, mostrando-se ao usuário como se observa na Figura 18.



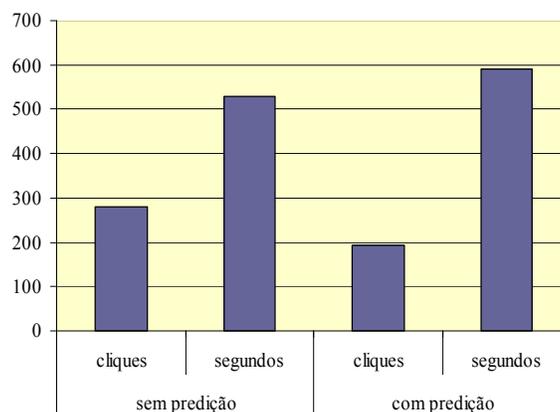
Figura 18: Interface do Teclado Virtual Livre ajustada para ser testado por uma criança de 10 anos.

As digitações sem e com a predição ocorreram necessitando dos números de cliques e segundos expostos na Tabela 3.

Tabela 3: Resultados obtidos na digitação do texto “A Lagarta” pela criança de 10 anos.

	Sem predição		Com predição	
	Cliques	Tempo de Execução (s)	Cliques	Tempo de Execução (s)
"A Lagarta"	281	528	194	590

Com estes resultados gerou-se a na Figura 19.

**Figura 19:** Valores obtidos na digitação do texto “A Lagarta” pela criança de 10 anos.

Aplicando-se o questionário do Anexo IV à criança, obtiveram-se as respostas da Tabela 4.

Tabela 4: Respostas dadas pela criança de 10 anos que testou o aplicativo.

	Sem predição		Com predição	
	BOM	RUIM	MELHOR	PIOR
Como foi escrever a história?	X		X	
	SIM	NÃO	MAIS	MENOS
Você cansou?		X		X
Você gostou?	X		X	

O teste realizado para obter dados passíveis de comparação à literatura referente deu origem à Tabela 5, a qual mostra o número de letras das 279 palavras do texto “O Menino e a Moeda”, e o número de cliques necessários para que a palavra desejada aparecesse na lista de predição do Teclado Virtual Livre com o uso da técnica de predição desenvolvida.

Tabela 5: Resultados obtidos na digitação de texto “O Menino e a Moeda” no teste realizado para comparação a dados obtidos em literatura referente.

Número de cliques para que a palavra desejada fosse indicada pelo aplicativo	Número de letras da palavra desejada									
	Uma	Duas	Três	Quatro	Cinco	Seis	Sete	Oito	Nove	Dez
Zero	29	12	11	10	5	7	1	0	0	0
Um	7	13	22	6	7	8	3	1	1	0
Dois	0	7	7	3	6	5	1	4	2	0
Três	0	0	4	11	19	4	8	3	3	0
Quatro	0	0	0	2	7	3	1	2	1	1
Cinco	0	0	0	0	1	1	3	0	0	0
Seis	0	0	0	0	0	0	0	0	0	0
Sete	0	0	0	0	0	0	0	0	0	0
Oito	0	0	0	0	0	0	0	0	1	0

Os testes de aplicabilidade, realizados por 43 voluntários indicaram os resultados das tarefas de digitação sem e com o uso da predição de palavras, apresentados nas Tabela 6, Tabela 7, Tabela 8 e Tabela 9, sendo as três primeiras obtidas na Escola Municipal Professora Genoveva Sicuro de Brito e a última na Escola Tia Vivian Marçal. Por ser o parâmetro tempo de caráter subjetivo, apresentam-se nas Tabelas 6 a 9 a redução percentual apenas do parâmetro número de cliques para os voluntários que apresentaram resultados que permitissem a determinação deste.

O uso do aplicativo nos computadores das Escolas envolvidas envolveu 11 máquinas de diferentes características técnicas, as quais não cabem, neste trabalho, de serem descritas, apresentando bom desempenho e rapidez no processamento.

A Tabela 6 refere-se aos dados obtidos com as crianças voluntárias da 4ª série A, do período matutino. Os valores representados por “xx” nas Tabelas 6 a 9, correspondem aos dados que não conseguiram ser obtidos por razões diversas, descritas detalhadamente na discussão dos resultados desta dissertação.

Tabela 6: Resultados dos testes realizados com as crianças voluntárias da 4ª série, turma A, obtidos com a digitação do texto sem e com o uso da predição.

	Sem predição		Com predição		Redução do número de cliques
	Cliques	Tempo de Execução (s)	Cliques	Tempo de Execução (s)	
Voluntário 1	258	732	189	812	26,74%
Voluntário 2	233	743	182	850	21,89%
Voluntário 3	96	xx	77	xx	xx
Voluntário 4	256	939	210	907	17,97%
Voluntário 5	318	1747	xx	xx	xx
Voluntário 6	283	883	205	919	27,56%
Voluntário 7	295	1733	86	650	xx
Voluntário 8	264	1019	185	781	29,92%
Voluntário 9	259	733	188	757	27,41%
Voluntário 10	282	1189	238	1079	15,60%
Voluntário 11	273	677	156	438	42,86%

Com base nos dados restantes da Tabela 6 – Voluntários 1, 2, 4, 6, 8, 9, 10 e 11 – foram gerados dois gráficos para análise comparativa dos parâmetros tomados nos testes, os quais são ilustrados pelas Figura 20 e Figura 21. Nota-se que nos gráficos seguintes, as siglas V1, V2, ..., etc. referem-se à, respectivamente, Voluntário 1, Voluntário 2, ..., etc.

Para obter os gráficos para a base comparativa dos parâmetros analisados, foram dispensados os resultados obtidos pelos voluntários 3, 5 e 7.

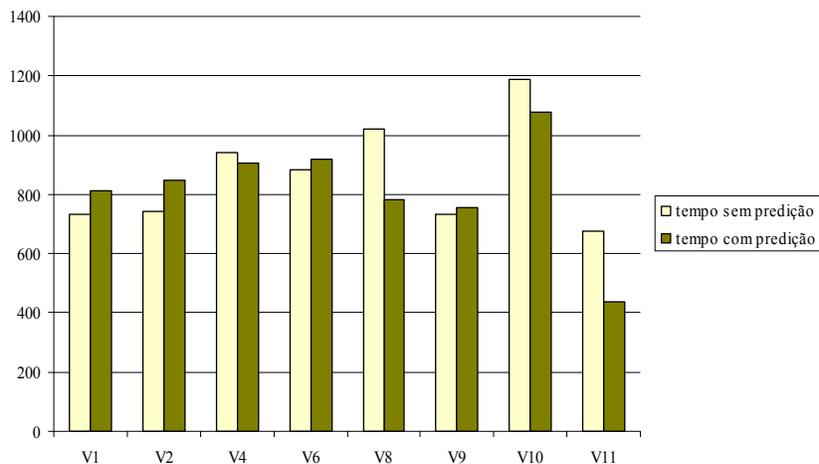


Figura 20: Gráfico Comparativo dos Tempos de Execução Gastos pelos Voluntários 1, 2, 4, 6, 8, 9, 10 e 11 da 4ª A para Digitação do Texto “A Lagarta” com e sem a Predição de Palavras.

Para o parâmetro número de cliques, os dados da Tabela 6 permitiram a geração do gráfico exposto na Figura 21.

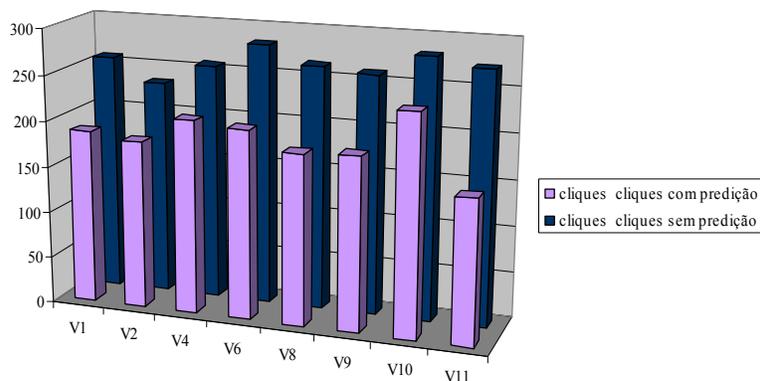


Figura 21: Gráfico Comparativo dos Números de Cliques Realizados pelos Voluntários da 4ªA da Escola Municipal Professora Genoveva Sicuro de Brito na Realização dos Testes.

A Tabela 7 mostra os números referentes aos resultados obtidos com os voluntários da turma B da 4ª série do período matutino.

Tabela 7: Resultados dos testes realizados com as crianças voluntárias da 4ª série, turma B, obtidos com a digitação do texto sem e com o uso da previsão.

	Sem previsão		Com previsão		Redução do número de cliques
	Cliques	Tempo de Execução (s)	Cliques	Tempo de Execução (s) (s)	
Voluntário 1	294	1360	xx	xx	xx
Voluntário 2	315	1104	250	834	20,63%
Voluntário 3	251	730	232	544	7,57%
Voluntário 4	326	950	273	850	16,26%
Voluntário 5	270	992	222	730	17,78%
Voluntário 6	220	623	199	595	9,55%
Voluntário 7	300	585	175	470	41,67%
Voluntário 8	305	850	173	840	43,28%
Voluntário 9	279	590	220	685	21,15%
Voluntário 10	237	1040	175	881	26,16%
Voluntário 11	242	610	227	710	6,20%

Os conjuntos de dados passíveis de análise comparativa da Tabela 7 geraram os gráficos das Figura 22 e Figura 23.

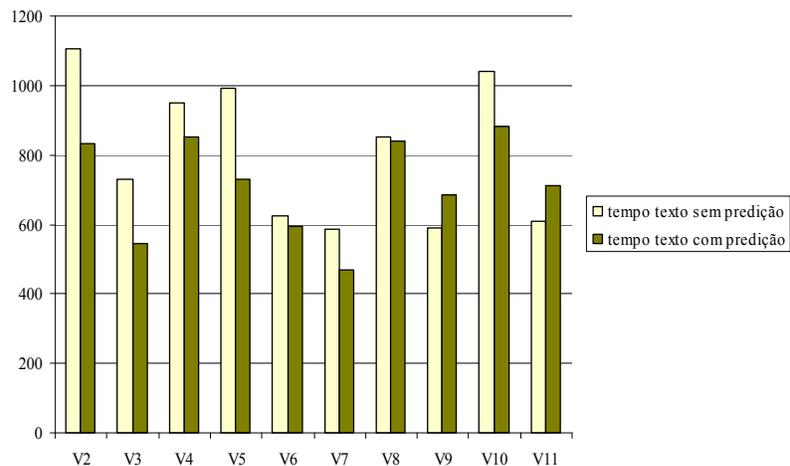


Figura 22: Gráfico Comparativo dos Tempos de Execução Despendidos pelos Voluntários da 4ª B para Digitação do Texto “A Lagarta” com e sem a Predição de Palavras, Desconsiderando-se os Dados do Voluntário 1.

Considerando-se os valores referentes redução do número de cliques necessários para a digitação do texto do protocolo de testes, plotou-se o gráfico da Figura 23.

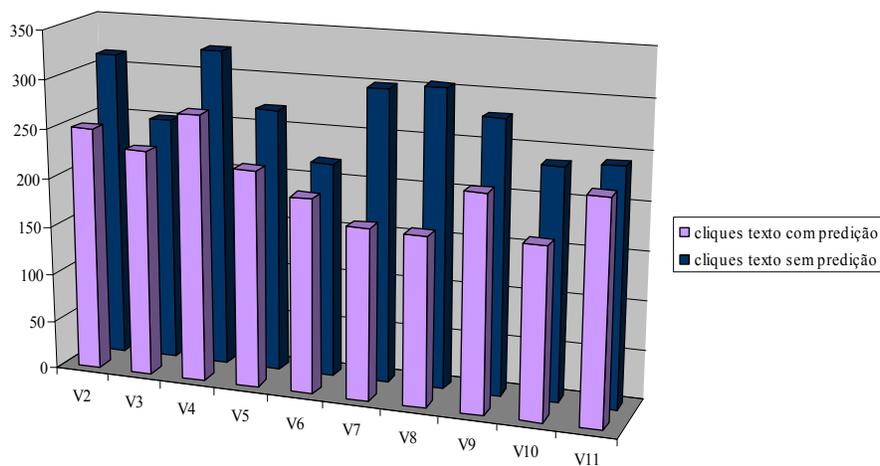


Figura 23: Gráfico Plotado sobre os Valores Obtidos na Digitação do Texto do Protocolo de Testes pelas Crianças Voluntárias da 4ª B.

Na Tabela 8, dispõe-se os resultados apresentados pelas crianças voluntárias da turma C da 4ª série do período vespertino.

Tabela 8: Resultados dos testes realizados com as crianças voluntárias da 4ª série, turma C, obtidos com a digitação do texto sem e com o uso da predição.

	Texto sem predição		Texto com predição		Redução do número de cliques
	Cliques	Tempo de Execução (s)	Cliques	Tempo de Execução (s)	
Voluntário 1	291	751	199	751	31,62%
Voluntário 2	304	1020	235	794	22,70%
Voluntário 3	248	840	230	963	7,26%
Voluntário 4	292	1175	224	1373	23,29%
Voluntário 5	287	951	273	830	4,88%
Voluntário 6	261	677	205	731	21,46%
Voluntário 7	272	950	256	1020	5,88%
Voluntário 8	293	669	186	735	36,52%
Voluntário 9	265	1500	xx	xx	xx
Voluntário 10	324	853	190	865	41,36%
Voluntário 11	253	840	163	761	35,57%
Voluntário 12	254	1050	192	997	24,41%

Para o parâmetro tempo da Tabela 8, gerou-se o gráfico da Figura 24.

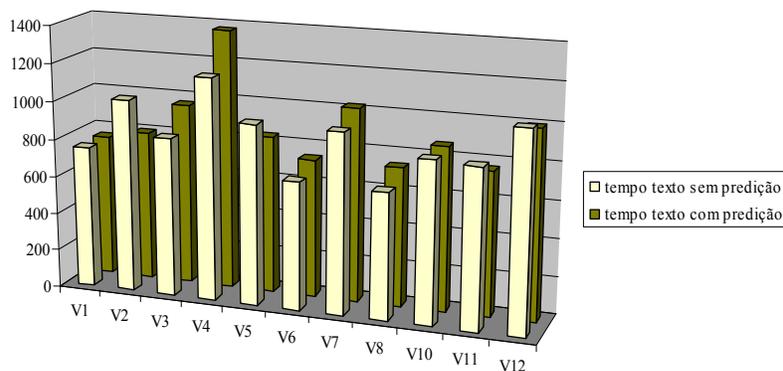


Figura 24: Gráfico Comparativo dos Tempos de Execução Despendidos pelos Voluntários da 4ª C para Digitação do Texto Referente Com e Sem a Predição de Palavras, Desconsiderando-se os Dados do Voluntário 9.

Já o parâmetro número de cliques apresentou as características ilustradas pela Figura 25.

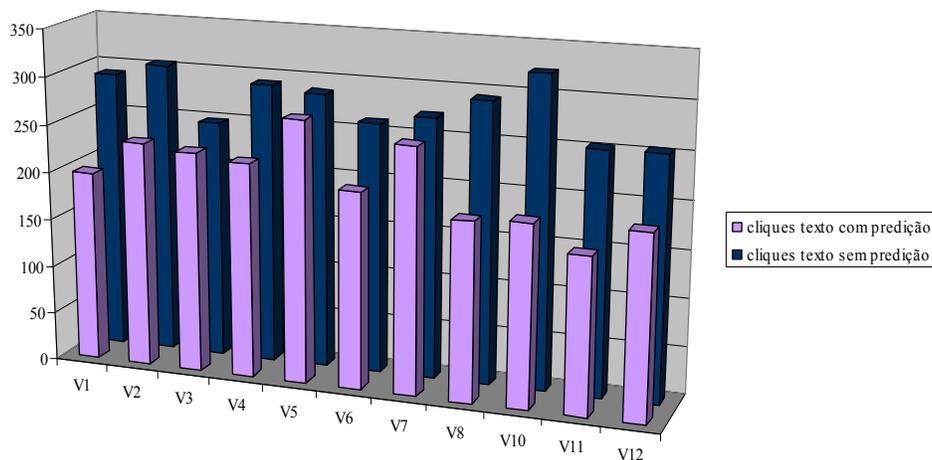


Figura 25: Gráfico Comparativo Plotado sobre os Valores Obtidos na Digitação do Texto do Protocolo de Testes pelos Alunos Voluntários da 4ª C.

A Tabela 9 mostra os resultados obtidos com o uso do aplicativo para a digitação do texto com e sem a predição, pelos alunos da Escola Tia Vivian Marçal. Destaca-se que os Voluntários 1, 3 e 5 desta Tabela, necessitaram que o texto fosse dividido em duas partes para a sua completa digitação sem a predição, devido a sério comprometimento motor e ansiedade apresentados na execução da tarefa; o que não ocorreu para a digitação com a técnica de predição.

Tabela 9: Resultados dos testes realizados com os voluntários da Escola Tia Vivian Marçal, obtidos com a digitação do texto sem e com o uso da predição.

	Texto sem predição		Texto com predição		Redução do número de cliques
	Cliques	Tempo de Execução (s)	Cliques	Tempo de Execução (s)	
Voluntário 1	314	5648	169	3959	46,18%
Voluntário 2	304	5216	215	3929	29,28%
Voluntário 3	343	4275	167	2394	51,31%
Voluntário 4	297	1864	245	2744	17,51%
Voluntário 5	388	3155	260	2374	32,99%
Voluntário 6	298	4733	195	1420	34,56%
Voluntário 7	372	5620	258	3781	30,65%

A Figura 26 ilustra os dados demonstrados na Tabela 9, considerando-se somente o parâmetro tempo.

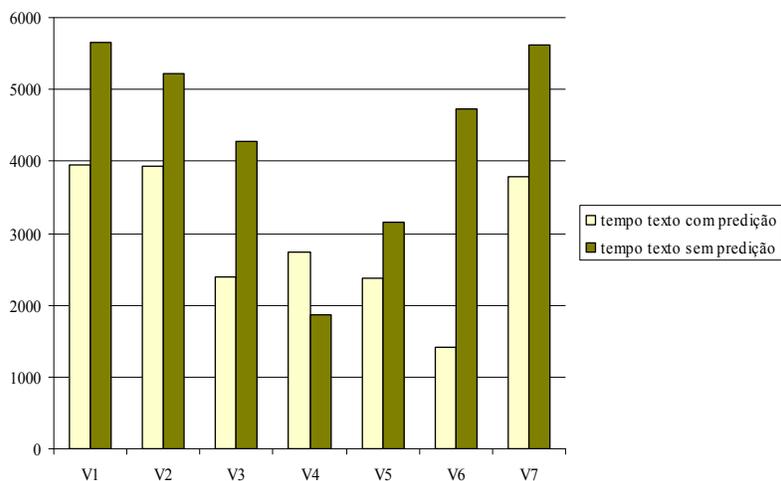


Figura 26: Resultados Obtidos para o Parâmetro Tempo de Execução pelos Alunos da Escola Tia Vivian Marçal.

Enquanto que a Figura 27 trata dos dados obtidos considerando-se o número de cliques realizado por cada voluntário.

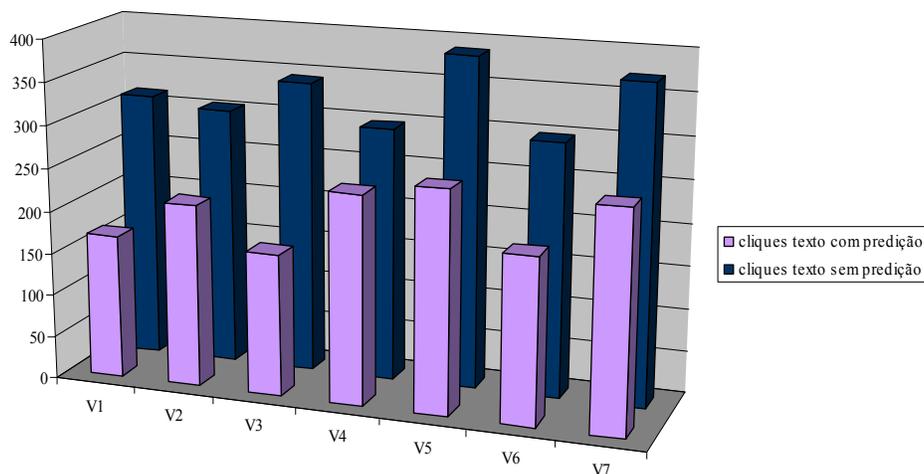


Figura 27: Número de Cliques Realizados pelos Voluntários da Escola Tia Vivian Marçal para a Digitação do Texto de Teste.

Quanto ao questionário (Anexo IV) entregue às crianças da Escola Municipal Professora Genoveva Sicuro de Brito, a Tabela 10 mostra as respostas obtidas.

Tabela 10: Respostas Obtidas na Apresentação do Questionário do Anexo I às Crianças Voluntárias da Escola Municipal Professora Genoveva Sicuro de Brito.

	sem predição		com predição	
	BOM	RUIM	MELHOR	PIOR
Como foi escrever a história?	31	3	25	9
	SIM	NÃO	MAIS	MENOS
Você cansou?	1	33	16	18
Você gostou?	29	5	26	8

O grupo de voluntários da Escola de Ensino Especial Tia Vivian Marçal não respondeu ao questionário em questão, por indicação da pedagoga responsável, a fim de evitar problemas com ansiedade.

CAPÍTULO 5

DISCUSSÃO

São analisados e comentados, individualmente, neste capítulo, os resultados obtidos na pesquisa realizada.

5.1 ANÁLISE DOS RESULTADOS

Os resultados são analisados em tópicos que se referem às etapas de projeto, implementação e teste necessárias ao desenvolvimento da técnica apresentada.

5.1.1 *Corpus*

No desenvolvimento deste trabalho, algumas decisões tomadas buscaram apresentar resultados com qualidades não diretamente relacionadas ao seu objetivo principal. A opção pela seleção manual de textos, adversamente ao automatismo de busca utilizado pelos trabalhos semelhantes de estruturação de *corpus* (PARDO & RINO, 2003, SARDINHA, 2004), requerendo criteriosa busca dos textos, os quais, selecionados pelo seu conteúdo, necessitaram leitura individual, assegurou ao usuário a qualidade gráfica das palavras apresentadas pela predição.

Esta leitura integral dos textos também fez com que publicações com excessivas figuras de linguagem, que podem confundir a produção textual de uma criança em alfabetização ou de usuários com déficit cognitivo, fossem eliminadas, assim como trabalhos com uso de gírias. Assim, o *corpus* deste trabalho diferencia-se dos demais disponíveis para estudos lingüísticos, podendo ser aplicado na sala de aula no processo de alfabetização sem que venha interferir de forma negativa neste processo.

Da mesma forma, o processamento do *corpus* etiquetado, quando verificadas as etiquetas e definida a classificação correta, nos casos em que o etiquetador utilizado apontava para mais de uma, como citado no Capítulo de Metodologia deste trabalho, destaca a qualidade desta estrutura.

Tais atividades demandaram um tempo maior do que o reservado para tais tarefas, no entanto, adicionou à base de textos do aplicativo características gramaticais mais corretas, oferecendo ao usuário palavras corretamente grafadas e classificadas dentro da melhor forma possível.

O *corpus* final obtido e que será disponibilizado na Internet de forma aberta e gratuita, apresenta qualidade gramatical e se aproxima da realidade verbal de crianças alfabetizadas. Salienta-se que alguns *corpora* pertencem a editoras ou grupos de pesquisa, sendo de acesso restrito (SARDINHA, 2000, RECSKI, 2005). A organização em pastas e sub-pastas, e tendo cada texto um arquivo com cabeçalho, permite que o mesmo seja aplicado com facilidade em estudos lingüísticos da língua portuguesa dentro do âmbito infantil.

5.1.2 Teclado Virtual Livre

O uso do Teclado Virtual Livre veio a enriquecer este trabalho por fazer parte do projeto AMPLISOFT que tem apresentado sucesso de uso, oferecendo ao seu público alvo um software livre, com código fonte aberto e acesso irrestrito (JORDAN, MATIAS & NOHAMA, 2006).

Assim, utilizando o Teclado Virtual Livre como interface para a técnica de predição desenvolvida neste trabalho, é possível disponibilizar uma nova versão do aplicativo com nova técnica de predição de palavras, como software livre, também, com código fonte aberto e gratuito, disponível na Internet, tornando-o, assim, mais acessível e passível de alterações conforme necessidade e desejo do usuário final.

Em contrapartida, esta pesquisa enriquece o projeto AMPLISOFT e seu aplicativo simulador de teclado apresentando resultados mais favoráveis ao seu uso por pessoas com necessidades especiais. Este fato é observado nos números atingidos com o uso da nova predição de palavras, com a qual se obteve redução do número de cliques variando entre 21 % e 35%, enquanto que com o uso da predição estatística simples, disponível até então para o aplicativo, esta redução foi de 15% (MATIAS, JORDAN & NOHAMA, 2004).

Ainda, esta pesquisa atingiu resultados favoráveis quanto ao parâmetro tempo de execução. No início das atividades, esperava-se obter acréscimo para tal parâmetro, como relatado no trabalho de Matias, Jordan & Nohama, (2004), no qual se refere a um acréscimo inicial de tempo de execução dos testes, reduzido após dois dias de treinamento dos voluntários. Já este trabalho obteve uma média de redução até 78% para este parâmetro,

evidenciando ainda mais a redução da dificuldade e dos movimentos envolvidos para a produção de um texto.

5.1.3 Modelos Ocultos de Markov

A decisão pelo uso dos parâmetros markovianos para a predição das palavras baseou-se na literatura (NAKAMURA *et al.*, 1990, KUHN, NIEMANN & SCHUKAT-TALAMAZZINI, 1994, CLARKSON, 1999, GASPERIN & LIMA, 2001, GUSTAVII & PETERSSON, 2003, ZIMMERMANN & BUNKE, 2004, LAW & CHAN, 2006, HUNNICUTT, NOZADZE & CHIKOIDZE, 2006); no entanto, requereria elevado esforço computacional se fossem empregados algoritmos usuais (RABINER, 1989) para o processamento da predição, devido ao fato de que o número de símbolos observáveis é elevado. Embora o *corpus* empregado neste trabalho seja classificado como pequeno, a quantidade de símbolos observados é de 7.066 palavras diferentes entre si. Considerando que o número de estados, isto é, de classes gramaticais etiquetadas no *corpus*, foi igual a 254, isto implicaria em uma matriz de observação (Matriz B), de dimensão (254 X 7.066), significando que seriam necessárias 1.794.764 operações de multiplicação, ou, pelo menos, de leitura e comparação de cada elemento para evitar desperdício de esforços computacionais com multiplicações por aqueles que contiverem valores nulos de probabilidades de observação.

A fim de mitigar o esforço computacional, dividiu-se a matriz de observação em duas, acelerando a recuperação e o processamento das informações envolvidas, respeitando as considerações abordadas no algoritmo de Viterbi.

A opção pelos Modelos Ocultos de Markov permitiu que o desenvolvimento de uma técnica de rápido processamento e boa taxa de acerto, mesmo quando comparada a técnicas mais elaboradas, como descreve o item 5.1.7 deste trabalho, fosse alcançado.

Os HMMs têm sido frequentemente utilizados nos processos de reconhecimento de escrita e fala em conjunto a um modelo lingüístico (ZIMMERMANN & BUNKE, 2004). Neste trabalho, aliou-se as técnicas desenvolvidas em tais reconhecimentos a um *corpus* em português, etiquetado e criteriosamente estruturado, para predizer palavras em um simulador de teclado, de forma precisa e exigindo pouco processamento computacional. Considerando as classes gramaticais como estados markovianos e dividindo a matriz de observações em duas matrizes, fez com que as probabilidades de seqüências das classes gramaticais em uma

sentença imperem sobre as das palavras, resultando em uma técnica gramaticalmente mais correta.

As matrizes e estruturas resultantes e utilizadas no processo inicial de instalação do aplicativo apresentaram tamanho compatível à capacidade de processamento e armazenagem dos computadores mais simples presentes nas escolas visitadas.

Comparativamente à técnica desenvolvida por Nakamura *et al.* (1990), a qual baseia-se em redes mistas e classificação do *corpus* por classes gramaticais, assim como este trabalho, sua taxa de acerto atingiu 90,7% de acerto, evidenciando a redução de esforço envolvido no preparo de um texto e ultrapassando os 86,9% obtidos por Nakamura *et al.*, (1990). A técnica de Nakamura *et al.*, (1990), apesar da semelhança na classificação das palavras, é mais elaborada, exigindo maior volume de processamento computacional, diminuindo a velocidade de resposta do aplicativo.

Os resultados obtidos para os parâmetros analisados (tempo de execução das tarefas e número de cliques) mostraram a vantagem do uso dos postulados markovianos no aplicativo Teclado Virtual Livre, o qual, ao ser utilizado com predição estatística simples alcançou 15% na redução do número de cliques e redução de tempo somente após dois dias de treinamento (MATIAS, JORDAN & NOHAMA, 2004).

5.1.4 Algoritmo de Predição

Por não computar toda a Matriz de Observação, o algoritmo atingiu o objetivo de ser eficiente mesmo quando processado em máquina com menor capacidade de processamento (Pentium 200 MHz). O uso de uma matriz auxiliar, denominada Matriz de Busca, confere ao trabalho uma diferença aos encontrados em literaturas de desenvolvimentos baseados em Viterbi, e permitiu reduzir os numerosos cruzamentos entre estados (Matriz A) e entre estados e observações (Matriz B), posto que sejam computados apenas os parâmetros não nulos. Ainda a indexação dos parâmetros tornou a busca das informações, sempre que necessária no decorrer do algoritmo, rápida, sem exigir muita memória para armazenamento de grandes planilhas.

Todo o aplicativo, com arquivos de instalação, código fonte, dados iniciais e objetos de interface ocupam cerca de 30 MB, tamanho portátil aos computadores atualmente utilizados nas escolas e de uso pessoal.

O tamanho do aplicativo e seu bom desempenho devem-se, também, ao fato dele ter sido desenvolvido na linguagem de programação *Microsoft Visual C++ v6.0*[®].

Por ter sua base de textos (*corpus*) fixa, as alterações lexicais exigem alterações no código fonte. No entanto, como o algoritmo de predição em conjunto com o Teclado Virtual Livre será disponibilizado na Internet sob regime de Software Livre, o trabalho de alteração será facilitado pelos comentários feitos em todo o código.

5.1.5 Testes Iniciais de Funcionamento do Teclado Virtual Livre com a Nova Técnica de Predição

Fez-se uma avaliação parcial de todo o desenvolvimento do *software* e de seu uso em conjunto ao aplicativo Teclado Virtual Livre, a fim de que problemas de compatibilidade, desempenho e funcionamento do aplicativo não viessem a atrapalhar os testes ou insatisfazer aos voluntários.

Os testes realizados, para verificação da capacidade de reconhecimento e resgate de trechos do próprio *corpus*, com uso do algoritmo, mostraram que, pela redução da quantidade de cliques, a predição ocorre. Foi observada uma redução média de 47% no número de cliques necessários para a digitação dos textos em questão.

No teste de coerência, com a digitação do texto “As Letrinhas”, obteve-se redução de 21% no número de cliques, enquanto que o tempo apresentou um acréscimo de 20%. Este acréscimo ocorre pelo tempo gasto na procura pela palavra desejada na lista de predição a cada letra digitada.

Com estes resultados, determinou-se que os testes com voluntários fossem realizados, pois o aplicativo e a técnica de predição comportavam-se como esperado.

5.1.6 Teste Individual

No gráfico da Figura 19, encontrado na página 66, constata-se redução tanto na quantidade de cliques e o incremento do tempo quando do uso da predição para a digitação do texto.

A criança que relatou ter se cansado menos ao digitar o texto com predição, mesmo tendo levado mais tempo para fazê-lo, obteve cerca de 31% na redução do número de cliques e um aumento de aproximadamente 12% no tempo. Relata-se que o aumento de tempo menor

que o indicado nos testes de fidelidade e coerência aconteceu devido ao fato desta criança já estar familiarizada com o uso do Teclado Virtual Livre.

Esse teste foi realizado com um voluntário já usuário do Teclado Virtual Livre, porém com a predição de palavras baseada em estatística simples (MATIAS & NOHAMA, 2003), para que se pudesse avaliar, de forma impessoal, o funcionamento do conjunto formado pelo aplicativo e pela técnica de predição baseada em Markov. Apesar de não se encontrar em literatura teste semelhante (feito com somente um voluntário), nota-se que os resultados obtidos são os esperados, ou seja, um ganho na aplicação de uma nova técnica, destacado aqui pela redução dos parâmetros (número de cliques e tempo de execução) envolvidos.

5.1.7 Testes para Obtenção de Parâmetros Passíveis de Comparação a Outra Técnica de Predição

Utilizou-se como fonte para comparação os dados obtidos e apresentados no trabalho de Nakamura *et al.* (1990), o qual apresenta uma técnica mista para a predição de palavras.

Para isso, seria necessário conhecer o percentual de acerto da técnica, ou seja, quando um texto qualquer é digitado com uso do Teclado Virtual Livre com predição de palavras baseada em HMM, quantas próximas palavras são adivinhadas e sugeridas pela lista de predição do aplicativo. Caso a próxima palavra não fosse indicada pela lista de predição, iniciava-se sua digitação e quando ela fosse sugerida, caso fosse, era computado o número de cliques necessário à sua aparição na lista.

Como se observa na Tabela 5, página 67, o número de palavras sugeridas pela técnica de predição desenvolvida neste trabalho foi de 253 palavras, sendo 75 delas sugeridas sem a necessidade da seleção de nenhuma letra, ou seja, foram sugeridas pelo algoritmo de predição e não do completar, como descreve a Metodologia deste trabalho. Desta forma, determina-se que 90,7% das palavras desejadas foram indicadas pelo aplicativo em algum estágio de digitação, enquanto que 26,9% foram indicadas automaticamente. Ainda, destaca-se que mais 68 palavras apareceram na lista de predição com somente a digitação da primeira letra da palavra desejada, o que corresponde a 24,4%. Logo, se somados os percentuais de acerto com nenhum e um clique, ultrapassa-se 50% das palavras de um texto de 279 palavras.

Nakamura *et al.* (1990), com sua técnica mista para o reconhecimento das palavras, alcançaram uma taxa de acerto de 85,5% com sua técnica sendo treinada com 512 sentenças

(cerca de 12.300 palavras) e de 86,9 %, quando treinada com 1024 sentenças (aproximadamente 24.600 palavras).

Apesar da forma como se manipulam os parâmetros da predição ser diferente, pois o trabalho de Nakamura *et al.* (1990) refere-se a uma técnica que usa redes neurais, um dos objetivos das técnicas de predição é diminuir o trabalho envolvido na produção de textos sejam eles lidos (reconhecimento de voz) ou escritos (simuladores de teclados). Desta forma, tomou-se como válida a comparação dos valores obtidos com o uso das técnicas, pelos seus objetivos comuns.

Denota-se que a técnica desenvolvida por Nakamura *et al.* (1990), apesar de permitir um auto-aprendizado da máquina para novas palavras, exige um esforço computacional muito maior que o da técnica aqui apresentada.

5.1.8 Testes de Aplicabilidade

Os testes de aplicabilidade do Teclado Virtual Livre em conjunto com a técnica de predição baseada em Modelos de Markov apresentaram resultados positivos, tanto na esperada redução do número de cliques quanto na aceitação do aplicativo pelas crianças.

Dá-se continuidade a este item analisando-se os grupos separadamente, tomando como grupos de controle os parâmetros tempo e número de cliques obtidos na digitação sem a predição.

5.1.8.1 Grupo Formado pela 4ª série A da Escola Municipal Professora Genoveva Sicuro de Brito

Da turma A da 4ª série, somente onze crianças realizaram os testes, por atenderem aos critérios de inclusão estabelecidos e descritos no capítulo da Metodologia. Dos onze voluntários, somente oito conjuntos de dados puderam ser aproveitados. Como se observa na Tabela 6, localizada na página 68, os resultados obtidos pelos voluntários 3, 5 e 7, apresentaram problemas descritos na seqüência.

O Voluntário 3 clicou na tecla *esc* do teclado convencional, fechando o Teclado Virtual Livre, quando já estava finalizando a primeira etapa do teste. Com isso, todos os seus dados foram perdidos. O mesmo voluntário pediu para começar novamente mas, por ter

ficado desanimado, digitou o texto parcialmente, somente as 14 palavras iniciais e não desejou que fosse cronometrado seu tempo, pois seria diferente de seus colegas. A pedagoga responsável pela turma relatou ser tal criança hiperativa e apresentar problemas comportamentais em todas as aulas de informática.

O Voluntário 5 não conseguiu realizar a segunda etapa do teste, por demorar na primeira etapa. Já o Voluntário 7 precisou ausentar-se da sala de aula quando acabara de digitar a 15ª palavra do texto.

Na Figura 20, da página 68, é possível observar a variação do tempo gasto pelos voluntários para a digitação do texto “A Lagarta”, sem e com a predição de palavras. Nota-se que quatro voluntários necessitaram de mais tempo para preparar o texto utilizando a predição, enquanto os quatro restantes apresentaram redução neste parâmetro.

A maior discrepância entre valores ocorreu com o Voluntário 11, o qual reduziu em 35,3% o tempo de digitação com o uso da predição de palavras, e o maior acréscimo de tempo foi apresentado pelo Voluntário 2, sendo 14,4% maior seu tempo de digitação com a predição. A redução média de tempo deste grupo foi de 17,8% enquanto que o acréscimo apresentou valor médio igual a 8,2%.

O desvio padrão calculado para o parâmetro tempo deste grupo foi de 17,1%, mostrando a grande discrepância entre os valores obtidos.

A Figura 21, página 69, ilustra a redução do número de cliques apresentada pelos oito voluntários na digitação do texto com a predição de palavras. O valor médio desta redução é de 26,3%, sendo a maior economia de trabalho apresentada pelo Voluntário 11, com 42,9% menos cliques. O desvio padrão apresentou valor igual a 8,4%. Este valor mostra que a variação do parâmetro número de cliques é menor que a do tempo de execução, para este grupo, o que é esperado, pois o conhecimento de informática dos alunos é diferenciado. Há crianças que possuem computadores domésticos e, por isso, assimilam mais rápido novos aplicativos, enquanto outras necessitam de mais tempo para uma mesma tarefa.

5.1.8.2 Grupo Formado pela 4ª série B da Escola Municipal Professora Genoveva Sicuro de Brito

Este grupo constituiu-se de 11 crianças as quais atenderam aos critérios de inclusão deste trabalho, sendo que somente o conjunto de dados do Voluntário 1 não pode ser aproveitado para as análises comparativas. Este aluno foi descrito pela pedagoga responsável

como sendo uma criança com problemas comportamentais e de aprendizado. O aluno declarou-se cansado e não conseguiu finalizar a segunda etapa do teste.

A Figura 22, da página 70, ilustra o gráfico comparativo do parâmetro tempo, no qual se observa que 80% dos voluntários apresentaram redução no tempo gasto para a digitação do texto em questão, sendo 26,4% a maior redução e 16,4% o maior acréscimo de tempo deste grupo de voluntários. O valor médio de redução foi de 16%.

O desvio padrão do parâmetro tempo determinado sobre os valores obtidos com o grupo da 4ªB tem valor 16,1%. No gráfico da Figura 23, ilustrado na página 70, é possível identificar a redução do número de cliques por todos os voluntários. O valor médio desta redução é de 21%, sendo o maior valor individual apresentado pelo Voluntário 8, igual a 43,3%. Sobre estes dados, determinou-se o desvio padrão com valor igual a 13,0%, novamente mostrando que os valores desviam menos que para o parâmetro tempo.

5.1.8.3 Grupo Formado pela 4ª série C da Escola Municipal Professora Genoveva Sicuro de Brito

Desta última turma de 4ª série da escola em questão, somente 12 crianças realizaram os testes, sendo que um dos conjuntos de dados (Voluntário 9), teve de ser desconsiderado na análise comparativa. A criança voluntária apresenta problemas de ansiedade, sendo acompanhada pela psicóloga da escola.

Para o parâmetro tempo declarado na Tabela 8, localizada na página 71, elaborou-se o gráfico da Figura 24, também na página 71, na qual se observa que quatro voluntários apresentaram redução de tempo, um gastou a mesma quantidade de segundos para digitar o texto com e sem a predição e os seis restantes tiveram acréscimo no parâmetro. O acréscimo médio foi de 9,7% e a redução média 12,3%, sendo os máximos valores iguais a 16,8% e 22,2%, respectivamente.

O desvio padrão do tempo para este grupo foi de 12,2%.

Na Figura 25, da página 72, observa-se que, a exemplo dos outros dois grupos, 100% dos voluntários obtiveram redução no número de cliques, sendo a máxima redução igual a 41,4% (Voluntário 10), a mínima 4,9% (Voluntário 5) e a redução média igual a 23,2%.

O desvio padrão para este parâmetro apresentou valor igual a 12,7%. Este grupo apresentou valores próximos para os desvios padrões, mostrando que grande parte deste grupo apresentou dificuldades maiores com o aplicativo, dado também identificado pela maior

média de tempo gasto 907 s com a predição e 903 s sem, contra, respectivamente, 807 e 714 s do grupo da 4ª B e, 654 e 691 s gastos em média pelos alunos da turma A.

5.1.8.4 Grupo Formado por Alunos da Escola de Ensino Especial Tia Vivian Marçal

Sob orientação da pedagoga Ilse Marçal, dez voluntários foram selecionados para a realização dos testes. Por motivos como ausência nas sessões de testes e ansiedade, três dos dez voluntários não finalizaram as etapas propostas, sendo formado um grupo com sete conjuntos de resultados. Este grupo formou-se por adolescentes, todos alfabetizados e com paralisia cerebral.

Os resultados expostos na Tabela 9 são analisados no gráfico da Figura 26, da página 73, no qual se observa a notável diferença nos valores do parâmetro tempo, sendo a maior diferença entre o tempo despendido para a digitação do texto sem e com a predição igual a 3313 s para o Voluntário 6, e ocorrendo um acréscimo de tempo de 880 s (Voluntário 4). Esta discrepância entre valores eleva o desvio padrão do parâmetro, fazendo-o atingir a marca de 82%. Entre os valores esperados e obtidos, o desvio padrão do parâmetro tempo de execução, para este grupo, apresentou o valor mais divergente. No entanto, como o grupo apresentava condições motoras, cognitivas e emocionais bastante diferenciadas, sendo necessário para alguns alunos até dividir cada etapa em duas, ou seja, digitação de parte do texto num dia e o restante em outro, este elevado valor de desvio padrão torna-se compreensível e aceitável.

Como os comprometimentos motores e cognitivos dos voluntários são, também, muito diferentes e não é objetivo deste trabalho estudá-las, este fator (desvio padrão) não é considerado problema para a pesquisa, apesar de seu alto valor.

Observa-se, ainda, que 86% dos voluntários reduziram o tempo para digitação do texto, sendo a média desta redução igual a 78%. Somente um voluntário necessitou de maior tempo para digitar o texto com o uso da predição, sendo gastos 880 s a mais.

Na Figura 27, localizada na página 73, ilustra-se o gráfico que mostra o comportamento do parâmetro número de cliques. Nele, observa-se que este parâmetro sofreu pouca alteração, sendo que todos os voluntários reduziram o número de cliques e a média deste parâmetro apresentou, aproximadamente, o valor de 35%. O desvio padrão, também, apresentou-se baixo, igual a 11,2%.

Destaca-se que para a digitação do texto sem a predição de palavras foi necessária a divisão do texto em duas partes para três voluntários, o que não foi necessário para a digitação

com a predição, evidenciando a diminuição da dificuldade para desempenhar esta tarefa quando comparada àquela.

Comparativamente à literatura correspondente, os valores obtidos na redução de cliques em todos os testes realizados superaram os obtidos por Fonseca & Carvalho (2002). Seus autores relatam redução média de 17% e valor máximo 34,1%, contra a média de 23,5% e máximo de 43,3% do grupo de voluntários da Escola Professora Genoveva Sicuro de Brito e média de 34,6% e máxima de 51,3% do grupo da Escola Tia Vivian Marçal. Ainda em comparação ao trabalho de Fonseca & Carvalho (2002), este parâmetro apresentou progresso, sendo que tais pesquisadores alcançaram, com seu teclado dinâmico com predição de palavras, redução média de 10,4%, considerando-se duas sessões de testes para treinamento do aplicativo. Ainda se deve considerar que o trabalho de Fonseca & Carvalho (2002) não tem agregada a análise gramatical, pois se baseia no histórico inicialmente composto por uma lista de palavras a qual é incrementada no decorrer do uso do aplicativo. Destaca-se que esta limitação inicial não oferece ao usuário opções de novas palavras, não despertando sua curiosidade e criatividade. Outro problema da técnica utilizada por Fonseca & Carvalho (2002) é a possibilidade de ocorrências de erros gramaticais (como concordância verbal e nominal) e de ortografias, por não considerarem as classes gramaticais das palavras e por aceitar e adicionar à sua lista de palavras base qualquer seqüência de caracteres que o usuário digite.

5.1.9 Respostas Dadas ao Questionário

O questionário presente no Anexo IV objetivou uma visualização da facilitação proporcionada à produção de texto, pois, como era esperado aumento do tempo causado pela busca das palavras na lista de predição, isso poderia representar um maior desgaste físico, sendo necessária uma resposta descritiva sobre a aceitação da técnica. Também não se encontrou em literatura teste semelhante a este. A realização deste teste teve base no pedido das pedagogas de instituições de ensino especial que temem o desagrado dos seus alunos na implementação de um novo aplicativo ou nova técnica para a produção textual ou aprendizado.

As respostas dadas pelos 34 alunos e resumidas na Tabela 10, da página 74, mostraram que 74% das crianças preferiram digitar o texto utilizando a predição de palavras. As crianças que relataram preferir digitar todas as letras a buscar na lista a palavra desejada eram crianças

que conheciam o teclado convencional e apresentavam facilidade no processo de seleção das letras no Teclado Virtual Livre.

As crianças que relataram cansaço ou tiveram problemas com o aplicativo declaram não terem gostado do mesmo.

As respostas negativas dadas ao questionário, nas quais os voluntários demonstraram alguma insatisfação com o aplicativo ou com a técnica foram, em sua totalidade, originárias de crianças que tiveram problemas com o aplicativo, como a criança que perdeu tudo o que havia digitado, ou por crianças com problemas comportamentais declarados por suas pedagogas responsáveis. Isto indica que, com treinamento e maior dedicação no ensino do aplicativo, estas más impressões poderão ser superadas.

De forma geral, pode-se afirmar que o aplicativo teve boa aceitação, visto o fato do reduzido contato que os voluntários tiveram com o aplicativo antes do seu uso.

5.2 TRABALHOS FUTUROS

Com base na pesquisa desenvolvida nesta dissertação, diversos trabalhos podem ser realizados, de modo a ampliar a aplicabilidade da ferramenta de produção textual.

A ampliação do número de palavras do *corpus* com enfoque infantil, com a seleção e adição de mais textos, poderá melhorar ainda mais os resultados obtidos nesta pesquisa, com maiores reduções do número de cliques, contribuindo com os ideais do projeto AMPLISOFT, bem como um aprofundamento ainda maior às classes gramaticais e suas inflexões poderá tornar a lista de palavras sugeridas mais precisas quanto ao gênero e número das classes.

Aplicando-se ao ensino, seja da educação especial ou não, podem ser feitos estudos de reforço educacional da língua portuguesa, colaboração do aplicativo no processo de alfabetização, entre outros processos didáticos.

Algumas sugestões para trabalhos futuros relacionados à modelagem usando Markov, são:

1. utilização de uma modelagem baseada na mistura de sentenças (*sentence-level mixture*): HMMs de n-gramas fornecem um *framework* que pode ser utilizado para representar uma variedade de modelos de linguagem disponíveis na literatura, assim uma possível variação do modelo proposto seria a mistura de sentenças, na qual modela-se diferentes tipos de sentenças separadamente;
2. avaliação do uso de trigramas a partir de HMMs de 3ª ordem;

3. medição do desempenho da predição de grupos de palavras ao invés de uma simples palavra, técnica denominada aglomeração (*clustering*), a qual permite a predição de grupos de palavras;
4. avaliação do impacto do uso de *caching* - palavras observadas recentemente são mais prováveis de ocorrer novamente;
5. avaliação detalhada da relação precisão X (*versus*) cobertura do método proposto utilizando *corpus* de diferentes tamanhos;
6. avaliação do uso de uma técnica de aprendizado incremental que permita aumentar a cobertura do método proposto mantendo a precisão, à medida que este é utilizado. Na literatura existem métodos de aprendizado incremental dedicados a HMM;
7. utilizar o algoritmo de Baum Welch para treinamento e prever as modificações necessárias para futuro aprendizado incremental.

Ainda, a adição de outras técnicas, como redes neurais, tornando o trabalho baseado em redes mistas de tomadas de decisão podem fazer com que palavras não presentes na base de textos sejam “aprendidas” pelo aplicativo, como no trabalho de Nakamura *et al.* (1990).

Em outras aplicações, além das vislumbradas pelo projeto AMPLISOFT, indica-se que o uso de *corpora* com diferentes enfoques pode permitir que outros tipos de usuários usufruam dos benefícios da predição de palavras. Por exemplo, textos médicos como base facilitariam o preenchimento de prontuários, além de padronizar os termos utilizados.

Da mesma forma, inserindo no mercado de trabalho uma pessoa com limitações motoras nos membros superiores, serviços de tele-atendimento poderiam ser realizados, desde que o *corpus* fosse composto de relatos de atendimentos anteriores, como um histórico.

Sugerem-se, também, estudos envolvendo a análise fisiológica do uso de tais emuladores comparado ao teclado convencional para pessoas que usam frequentemente o computador, sendo atingidos por problemas, como Lesões por Esforços Repetitivos (LER) (<http://www.neuro.med.br/LER.htm>), cooperando com os estudos da ergonomia das interfaces homem-máquina.

CAPÍTULO 6

CONCLUSÕES

Pessoas ativas socialmente são pessoas mais saudáveis, com elevada auto-estima e, conseqüentemente, seu ambiente familiar torna-se mais agradável. Além disso, pessoas mais saudáveis oneram menos os cofres públicos e podem participar de ambientes antes difíceis, como escolas e mercado de trabalho. Com base nesses fatos, técnicas vêm sendo desenvolvidas a fim de ampliar a inclusão social. Emuladores de teclado com facilitação para a produção textual permitem que deficientes verbais possam se comunicar quando o aplicativo é adicionado a um sintetizador de voz. A ampliação na produção de textos pode inserir no mercado de trabalho pessoas sem ou com limitações nos membros superiores, se tais emuladores forem associados a apontadores alternativos, como os de cabeça. Enfim, uma técnica de predição associada a um emulador de teclado pode ter aplicações diferentes, para usuários diferentes, não sendo uma aplicação mais importante do que a outra.

Focados nesse contexto, as principais contribuições técnico-científicas desta pesquisa podem ser resumidas no desenvolvimento do *corpus* e da Matriz B.

Os resultados obtidos indicam que maior quantidade e qualidade de textos também são obtidas, pois toda a predição considera regras gramaticais e o seu desenvolvimento baseia-se em um *corpus* criteriosamente estruturado. A elaboração do *corpus* através de criteriosa seleção dos textos fez com que ele fosse composto por um número de palavras que o caracteriza como pequeno (SARDINHA, 2004); no entanto, os resultados evidenciaram que a qualidade da predição não foi prejudicada por este fato. A qualidade do *corpus* e sua especialidade, resultantes do processamento do mesmo, contribuem para o desenvolvimento de estudos lingüísticos. A elaboração da lista de pares de palavras x (*versus*) classe gramatical do *corpus* estruturou a base de dados para a determinação dos parâmetros markovianos. A disponibilização de todo o desenvolvimento da técnica de predição em sítio na Internet faz com que o aplicativo seja de acesso irrestrito aos seus usuários. A limitação lingüística apontada pela utilização de um *corpus* pode ser ultrapassada pela abertura do código fonte, totalmente comentado, disponibilizado em Internet sob regime Livre e Gratuito. Se for do desejo de algum usuário alterar a base de textos, basta ter um novo *corpus*, tratá-lo com as rotinas desenvolvidas neste trabalho e atualizar o código fonte da técnica de predição disponível.

O trabalho realizado proporciona uma maior facilidade na produção de textos, gerando comunicação e permitindo que textos maiores possam ser preparados sem maiores dificuldades aos usuários e sem maiores esforços computacionais caracterizados por lentidão no processamento. A determinação dos parâmetros markovianos e o levantamento das ações esperadas do algoritmo de predição indicaram a necessidade do desenvolvimento de uma técnica diferenciada, com o uso de uma matriz auxiliar (Matriz de Busca), o que concedeu ao algoritmo de predição velocidade de processamento, permitindo que o aplicativo possa ser utilizado com menor esforço computacional, ou seja, por computadores menos velozes. O desenvolvimento desta matriz auxiliar designou a este trabalho uma característica inovadora e contribui cientificamente à pesquisas com uso de Modelos de Markov, promovendo uma forma diferenciada de manipular os parâmetros markovianos, nem sempre necessária quando se tem máquinas de rápido processamento, mas vital quando essa realidade não é encontrada. Os testes do protocolo desenvolvido e adaptado ao Teclado Virtual Livre, realizados com voluntários alfabetizados, crianças e jovens, mostraram que, apesar do incremento do parâmetro tempo para alguns usuários, estes apontaram, por intermédio do questionário, que o uso da predição é menos cansativo e mostraram-se mais contentes ao utilizá-la. Essa aceitação pode facilitar a interação homem-máquina e auxiliar na comunicação escrita de pessoas com dificuldades. A análise estatística dos resultados permite visualizar a redução da quantidade de cliques, a qual indica que a quantidade de movimentos envolvidos na digitação foi reduzida e que ocorre o uso das palavras pertencentes à lista de predição, as quais consideram regras gramaticais; logo, o texto produzido, por considerar tais regras, apresenta qualidade.

Desta forma, com base na literatura citada neste trabalho, importância social das inclusões digital e social e nos resultados positivos obtidos nesta pesquisa, aponta-se a importância de agregar aos sistemas de comunicação um acelerador de uso eficiente, permitindo maior autonomia às pessoas consideradas não capazes por não se expressarem, seja de forma oral ou escrita, ou o fazerem de forma lenta e dependente de outra pessoa.

APÊNDICE I

CABEÇALHO DOS TEXTOS QUE COMPÕEM O *CORPUS*

Este Anexo traz os cabeçalhos contendo os dados básicos dos textos constituintes do *corpus* que foi a base do desenvolvimento da técnica de predição desenvolvida nesta dissertação.

CABEÇALHOS EM ORDEM ALFABÉTICA

<título A Barata Arrogante>

<fonte http://sitededicadas.uol.com.br/conto_leitor8b.htm>

<autor A.Jorge Silva>

<língua portuguesa>

<tipo de conto>

<título A Boneca Zanolha>

<fonte <http://www.contos.poesias.nom.br/abonecazarolha/abonecazarolha.htm>>

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de infanto-juvenil>

<título A Discussão dos Talheres>

<fonte <http://www.contos.poesias.nom.br/adiscussaodostalheres/> >

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de infanto-juvenil>

<título A Fábrica da Laranja>

<fonte <http://www.contos.poesias.nom.br/afabricadelaranja/afabricadelaranja.htm>>

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de infantil>

<título A Fruta>

<fonte http://sitededicas.uol.com.br/hi2_p0.htm>

<autor Alberto Filho>

<língua portuguesa>

<tipo de texto infantil>

<título a lebre e o cão de caça>

<fonte <http://sitededicas.uol.com.br/fabula23a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título A Menina e a Boneca>

<fonte <http://www.contos.poesias.nom.br/ameninaeaboneca/ameninaeaboneca.htm>>

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de infanto-juvenil>

<título A Farra dos Mortos>

<fonte <http://www.contos.poesias.nom.br/afarradosmortos/afarradosmortos.htm>>

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de conto>

<título A Missa dos Mortos>

<fonte <http://sitededicas.uol.com.br/folk17.htm>>

<autor desconhecido>

<língua portuguesa>

<tipo de lenda>

<título a mula>

<fonte <http://sitededicas.uol.com.br/fabula17a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título a mulher e sua galinha>

<fonte <http://sitededicas.uol.com.br/fabula9a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título a raposa e as uvas>

<fonte <http://sitededicas.uol.com.br/fabula30a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título A Última Noite de Natal>

<fonte http://www.releituras.com/graciramos_natal.asp>

<autor Graciliano Ramos>

<língua portuguesa>

<tipo de infanto juvenil>

<título A Vassoura Deprimida>

<fonte <http://www.contos.poesias.nom.br/avassouradeprimida/> >

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de infanto-juvenil>

<título Aladim>

<fonte <http://virtualbooks.terra.com.br/>>

<autor desconhecido>

<língua portuguesa>

<tipo de contos populares>

<título A Escolha dos Livros Infantis>

<fonte <http://www.graudez.com.br/litinf/trabalhos/editor1.htm>>

<autor Cristiane Madanêlo De Oliveira>

<língua portuguesa>

<tipo de texto artigo>

<título a árvore mágica>

<fonte http://sitededicas.uol.com.br/hi4_p0.htm>

<autor Alberto Filho>

<língua portuguesa>

<tipo de texto infantil>

<título a árvore e os machados>

<fonte <http://sitededicas.uol.com.br/fabula15a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título as lebres e as rãs>

<fonte <http://sitededicas.uol.com.br/fabula11a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título Boitatá>

<fonte <http://www.arteducacao.pro.br/Cultura/lendas>>

<autor desconhecido>

<língua portuguesa>

<tipo de lenda>

<título Branca de Neve>

<fonte <http://virtualbooks.terra.com.br/>>

<autor Irmãos Grimm>

<língua portuguesa>

<tipo de conto>

<título A Menina dos Brincos de Ouro>

<fonte <http://sitededicadas.uol.com.br/ct01a.htm>>

<autor desconhecido>

<língua portuguesa>

<tipo de contos populares>

<título Caipora>

<fonte <http://www.arteducacao.pro.br/Cultura/lendas.htm#Caipora>>

<autor desconhecido>

<língua portuguesa>

<tipo de lenda>

<título Cantiga dos Pastores>

<fonte http://www.releituras.com/aprado_cantiga.asp>

<autor Adélia Prado>

<língua portuguesa>

<tipo de texto infanto-juvenil>

<título Carta a um Pintassilgo>

<fonte <http://www.contos.poesias.nom.br/cartaaumpintassilgo/>>

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de infantil>

<título Chapeuzinho vermelho>

<fonte http://sitededicadas.uol.com.br/hti_cv01.htm>

<autor Irmãos Grimm>

<língua portuguesa>

<tipo de conto>

<título Cinderela>

<fonte <http://virtualbooks.terra.com.br/>>

<autor Irmãos Grimm>

<língua portuguesa>

<tipo de conto>

<título Filhotinhos da Rua>

<fonte <http://www.contos.poesias.nom.br/filhotinhosdarua/filhotinhosdarua.htm>>

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de infantil>

<título a formiga e a pomba>

<fonte <http://sitededicas.uol.com.br/fabula14a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de texto fábula>

<título a galinha e os ovos de ouro>

<fonte <http://sitededicas.uol.com.br/fabula13a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título Histórias de Bem-te-vi>

<fonte http://www.releituras.com/cmeireles_bemtevi.asp>

<autor Cecília Meireles>

<língua portuguesa>

<tipo de texto infanto-juvenil>

<título Joãozinho e Maria>

<fonte <http://virtualbooks.terra.com.br/>>

<autor Ruth rocha>

<língua portuguesa>

<tipo de conto>

<título Lá Vem o Chuchu>

<fonte <http://www.universoespirita.org.br/DIAGRAMACAO%20EM%20251201/> >

<autor Majô/Esmeralda>

<língua portuguesa>

<tipo de texto infantil>

<título A Lenda das Lapas>

<fonte <http://www.usinadeletras.com.br/exibelotexto.phtml?cod=2005&cat> >

<autor José J. Serpa>

<língua portuguesa>

<tipo de lenda>

<título Linha da Vida>

<fonte http://www2.uol.com.br/cultvox/livros_gratis/porta_palavra.pdf>

<autor Humberto Bley Menezes>

<língua portuguesa>

<tipo de crônicas>

<título Lobisomem>

<fonte <http://sitededicas.uol.com.br/folk03.htm>>

<autor desconhecido>

<língua portuguesa>

<tipo de lenda>

<título A mãe d'água>

<fonte <http://sitededicas.uol.com.br/folk09.htm>>

<autor desconhecido>

<língua portuguesa>

<tipo de lenda>

<título mandioca>

<fonte http://www.universoespirita.org.br/05_01_2002/Mandioca.htm>

<autor Majô/Esmeralda>

<língua portuguesa>

<tipo de texto infantil>

<título Matinta Perêra>

<fonte <http://sitededicas.uol.com.br/folk07.htm>>

<autor desconhecido>

<língua portuguesa>

<tipo de lenda>

<título Menção Honrosa>

<fonte http://www2.uol.com.br/cultvox/livros_gratis/porta_palavra.pdf>

<autor Humberto Bley Menezes>

<língua portuguesa>

<tipo de crônicas>

<título A Menina dos Brincos de Ouro>

<fonte <http://sitededicas.uol.com.br/ct01a.htm>>

<autor desconhecido>

<língua portuguesa>

<tipo de contos populares>

<título O menino que viu uma coisa>

<fonte http://sitededicas.uol.com.br/hi8_p1.htm>

<autor Alberto Filho>

<língua portuguesa>

<tipo de texto infantil>

<título Minha Vó Fez pra mim>

<fonte <http://www.usinadeletras.com.br/exibelotexto.phtml?cod=2052&cat>>

<autor Maria da Graça Almeida>

<língua portuguesa>

<tipo de infanto juvenil>

<título o mistério do velho casarão>

<fonte http://sitededicas.uol.com.br/ct9_p1.htm>

<autor Alberto Filho>
<língua portuguesa>
<tipo de texto infantil>

<título O monstro e o pé de galinha>
<fonte http://sitededicadas.uol.com.br/hpe_p0a.htm>
<autor Alberto Filho>
<língua portuguesa>
<tipo de texto juvenil>

<título Mula sem cabeça>
<fonte <http://sitededicadas.uol.com.br/folk10.htm>>
<autor desconhecido>
<língua portuguesa>
<tipo de lenda>

<título A Mulher da Meia-noite>
<fonte <http://sitededicadas.uol.com.br/folk13.htm>>
<autor desconhecido>
<língua portuguesa>
<tipo de lenda>

<título Natal da Ilha do Nanja>
<fonte http://www.releituras.com/cmeireles_natal.asp>
<autor Cecília Meireles>
<língua portuguesa>
<tipo de texto infanto-juvenil>

<título Natal no Chile>
<fonte http://www2.uol.com.br/cultvox/livros_gratis/porta_palavra.pdf>
<autor Humberto Bley Menezes>
<língua portuguesa>
<tipo de crônicas>

<título O Negrinho do Pastoreio>
<fonte <http://sitededicas.uol.com.br/folk08.htm>>
<autor desconhecido>
<língua portuguesa>
<tipo de lenda>

<título o asno, a raposa e o leão>
<fonte <http://sitededicas.uol.com.br/fabula6a.htm>>
<autor Esopo>
<língua portuguesa>
<tipo de fábula>

<título o asno e o velho pastor>
<fonte <http://sitededicas.uol.com.br/fabula12a.htm>>
<autor Esopo>
<língua portuguesa>
<tipo de fábula>

<título o asno em pele de leão>
<fonte <http://sitededicas.uol.com.br/fabula24a.htm>>
<autor Esopo>
<língua portuguesa>
<tipo de fábula>

<título O Bicho Folharal>
<fonte <http://sitededicas.uol.com.br/ct02a.htm>>
<autor desconhecido>
<língua portuguesa>
<tipo de contos populares>

<título o boi e a rã>
<fonte <http://sitededicas.uol.com.br/fabula21a.htm>>
<autor Esopo>
<língua portuguesa>

<tipo de fábula>

<título O Buraco>

<fonte <http://www.contos.poesias.nom.br/oburaco/oburaco.htm>>

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de conto>

<título o cachorro e sua sombra>

<fonte <http://sitededicas.uol.com.br/fabula2a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título O Cão, o Lobo e o Mestre Macaco>

<fonte <http://www.contos.poesias.nom.br/ocaoelobo/ocaoelobo.htm>>

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de infantil>

<título o cão raivoso>

<fonte <http://sitededicas.uol.com.br/fabula27a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título o carvalho e os juncos>

<fonte <http://sitededicas.uol.com.br/fabula16a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título o cavalo e o seu tratador>

<fonte <http://sitededicas.uol.com.br/fabula5a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título o cego e o filhote de lobo>

<fonte <http://sitededicas.uol.com.br/fabula20a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título o cervo doente>

<fonte <http://sitededicas.uol.com.br/fabula4a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título o corvo e o jarro>

<fonte <http://sitededicas.uol.com.br/fabula28a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título o filhote de cervo e sua mãe>

<fonte <http://sitededicas.uol.com.br/fabula8a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título o galo de briga e a águia>

<fonte <http://sitededicas.uol.com.br/fabula22a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título o galo e a pedra preciosa>

<fonte <http://sitededicas.uol.com.br/fabula7a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título o gato e o galo>

<fonte <http://sitededicas.uol.com.br/fabula1a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título O Gato e o Rato>

<fonte <http://www.contos.poesias.nom.br/ogatoeorato/ogatoeorato.htm>>

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de infantil>

<título o ladrão e o cão de guarda>

<fonte <http://sitededicas.uol.com.br/fabula18a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título o leão apaixonado>

<fonte <http://sitededicas.uol.com.br/fabula25a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título o leão e o rato>

<fonte <http://sitededicas.uol.com.br/fabula3a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título o leão e os três touros>

<fonte <http://sitededicas.uol.com.br/fabula19a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título o leão, o urso e a raposa>

<fonte <http://sitededicas.uol.com.br/fabula29a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título o lobo e a garça>

<fonte <http://sitededicas.uol.com.br/fabula10a.htm>>

<autor Esopo>

<língua portuguesa>

<título o lobo e a ovelha>

<fonte <http://sitededicas.uol.com.br/fabula26a.htm>>

<autor Esopo>

<língua portuguesa>

<tipo de fábula>

<título O Menino e A Flor>

<fonte <http://www.contos.poesias.nom.br/omeninoeafior/omeninoeafior.htm>>

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de infantil>

<título O Menino e o Padre>

<fonte <http://sitededicas.uol.com.br/ct04a.htm>>

<autor desconhecido>

<língua portuguesa>

<tipo de contos populares>

<título O Menino, o Burro e o Cachorro>

<fonte <http://sitededicas.uol.com.br/ct05a.htm>>

<autor desconhecido>

<língua portuguesa>

<tipo de contos populares>

<título O Morto>

<fonte <http://www.contos.poesias.nom.br/omorto/omorto.htm>>

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de conto>

<título O Papai Noel>

<fonte <http://www.contos.poesias.nom.br/opapainoel/opapainoel.htm>>

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de infantil>

<título O Patinho Feio>

<fonte http://virtualbooks.terra.com.br/freebook/infantis/o_patinho_feio.htm>

<autor Hans Christian Andersen>

<língua portuguesa>

<tipo de infantil>

<título O rei que era dono do mundo>

<fonte http://sitededicas.uol.com.br/hi6_p1.htm>

<autor Alberto Filho>

<língua portuguesa>

<tipo de texto infantil>

<título O Sapo e a Flor>

<fonte <http://www.contos.poesias.nom.br/osapoeaflor/osapoeaflor.htm>>

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de infantil>

<título O sonho da Lua>

<fonte <http://www.contos.poesias.nom.br/osonhodalu/osonhodalu.htm>>

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de infantil>

<título O Ursinho Chorão>

<fonte http://virtualbooks.terra.com.br/osmelhoresautores/O_Ursinho_chorao.htm>

<autor José Guimarães>

<língua portuguesa>

<tipo de infantil>

<título O Velho Ambicioso>

<fonte <http://sitededicas.uol.com.br/ct06.htm>>

<autor desconhecido>

<língua portuguesa>

<tipo de contos populares>

<título Ora, um Carrinho>

<fonte <http://www.contos.poesias.nom.br/oraumcarrinho/oraumcarrinho.htm>>

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de infantil>

<título Os Compadres Corcundas>

<fonte <http://sitededicas.uol.com.br/ct07a.htm>>

<autor desconhecido>

<língua portuguesa>

<tipo de contos populares>

<título Papa Figo>

<fonte <http://sitededicadas.uol.com.br/folk04.htm>>

<autor desconhecido>

<língua portuguesa>

<tipo de lenda>

<título Passeata>

<fonte http://www2.uol.com.br/cultvox/livros_gratis/porta_palavra.pdf>

<autor Humberto Bley Menezes>

<língua portuguesa>

<tipo de crônicas>

<título Pequeno Polegar>

<fonte <http://virtualbooks.terra.com.br/>>

<autor desconhecido>

<língua portuguesa>

<tipo de conto>

<título Pinóquio>

<fonte <http://virtualbooks.terra.com.br/>>

<autor Carlo Collodi>

<língua portuguesa>

<tipo de conto>

<título Psiu>

<fonte <http://www.usinadeletras.com.br/exibelotexto.phtml?cod=2055&cat> >

<autor Maria da Graça Almeida>

<língua portuguesa>

<tipo de infanto juvenil>

<título Romãozinho>

<fonte <http://sitededicadas.uol.com.br/folk16.htm>>

<autor desconhecido>

<língua portuguesa>

<tipo de lenda>

<título Rosa Amarela>

<fonte <http://www.contos.poesias.nom.br/cantigaderoda2/cantigaderoda2.htm>>

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de cantiga de roda>

<título Saci Pererê>

<fonte <http://sitededicas.uol.com.br/folk01.htm>>

<autor desconhecido>

<língua portuguesa>

<tipo de lenda>

<título Triseza no Fundo do Mar>

<fonte <http://www.contos.poesias.nom.br/tristezanofundodomar/>>

<autor Maria Hilda de J. Alão>

<língua portuguesa>

<tipo de infantil>

<título Um Lápis Encantado>

<fonte http://sitededicas.uol.com.br/conto_leitor10b.htm>

<autor Edna Araújo>

<língua portuguesa>

<tipo de infantil>

<título Vitória Régia>

<fonte <http://sitededicas.uol.com.br/folk11.htm>>

<autor desconhecido>

<língua portuguesa>

<tipo de lenda>

APÊNDICE II

ROTINAS DESENVOLVIDAS PARA A TÉCNICA DE PREDIÇÃO

São apresentadas os códigos das rotinas elaboradas a fim de auxiliar o desenvolvimento do algoritmo de predição. Tais rotinas foram desenvolvidas com auxílio do *Microsoft Visual Basic for Application*[®] (no ambiente do aplicativo Word[®] ou Excel[®]).

No Word:

```
Sub CocatenaTxt ()
' CocatenaTxt Macro
' Macro gravada 27/6/2006 por Monica Jordan
Set dlgCaminho = Application.FileDialog(msoFileDialogFolderPicker)
dlgCaminho.Show
Documents.Add
myName = Dir("*.TXT")
While myName <> ""
    With Selection
        .InsertFile FileName:=myName, ConfirmConversions:=False
        .InsertParagraphAfter
        .InsertBreak Type:=wdSectionBreakNextPage
        .Collapse Direction:=wdCollapseEnd
    End With
    myName = Dir()
Wend
End Sub
```

No Excel:

```
Private Sub Workbook_Open()
Dim Sinal As Boolean
Sinal = True
For Each ws In Worksheets
    If ws.Name = "Contagem das Classes" Then Sinal = False
Next ws
If Sinal Then FrequenciaDeClasses
CarregaIndice
End Sub
'-----
'Variáveis globais
Const NumPalavrasPorClasse As Integer = 827 'carregar o número máx de palavras por
classe observado no corpus
Const SaidaParaPlanilha As Boolean = False 'Ativa/Desativa geração de planilha
Const SaidaParaTxt As Boolean = True 'Ativa/Desativa geração de ficheiro
texto
Const ColunasporLinha As Integer = 100 'definição para formatar arquivo txt
Const nElemPorLinha As Integer = 6

Type TipoListaDeEstados
NomeDoEstado As String
```

```

    ContagemOuPosicao As Integer
End Type
Type PalFreq
    Palavra As String
    Ocorrencia As Integer
    Probabilidade As Double
End Type
Type TipoMatrizB
    Pares(NumPalavrasPorClasse) As PalFreq
End Type

Dim N As Integer
Dim Indice() As String
Dim MatrizAuxB() As TipoMatrizB

'*****
Sub CarregaIndice()
' Macro gravada em 7/8/2006 por Monica Jordan & Paulo C. Pereira
' Atalho do teclado: Ctrl+q
    Sheets("Contagem das Classes").Activate
    Range("A1").Activate
    N = -1
    Do While ActiveCell.Value <> ""
        N = N + 1
        ReDim Preserve Indice(N)
        Indice(N) = ActiveCell.Text
        ActiveCell.Offset(1).Activate
    Loop
    Range("A1").Activate
End Sub

'*****
Sub FrequenciaDeClasses()
' Macro gravada em 7/8/2006 por Monica Jordan & Paulo C. Pereira
' Atalho do teclado: Ctrl+w
' Se acusar erro, aumentar o pressuposto número de estados no corpus...
' isto é, aumentar o valor entre parentesis na linha abaixo
    Dim ListaDeEstados(499) As TipoListaDeEstados
    Dim Localizado As Boolean
    Dim i As Integer
    Dim ff As Long
    Dim Arquivo As String
    Dim LinhaDeTexto As String

    Arquivo = SelArquivo("concatenados.txt")
    ff = FreeFile
    N = 0
    Open Arquivo For Input As *ff
    Do While Not EOF(ff)
        Line Input *ff, LinhaDeTexto
        If LinhaDeTexto Like "[*]" Then
            Localizado = False
            For i = 0 To N
                If ListaDeEstados(i).NomeDoEstado = TagsNaLinha(LinhaDeTexto) Then
                    Localizado = True
                    ListaDeEstados(i).ContagemOuPosicao = ListaDeEstados(i).ContagemOuPosicao
+ 1
                End If
            Next i
            If Not Localizado Then
                ListaDeEstados(N).NomeDoEstado = TagsNaLinha(LinhaDeTexto)
                ListaDeEstados(N).ContagemOuPosicao = 1
                N = N + 1
            End If
        End If
    Loop
    Close *ff
    Sheets.Add

```

```

ActiveSheet.Name = "Contagem das Classes"
Range("A1").Activate
For i = 0 To N - 1
    ActiveCell.Value = ListaDeEstados(i).NomeDoEstado
    ActiveCell.Offset(0, 1).Value = ListaDeEstados(i).ContagemOuPosicao
    ActiveCell.Offset(1).Activate
Next i
Columns("A:B").Sort Key1:=Range("A1"), Order1:=xlAscending, Key2:=Range("B1") _
, Order2:=xlAscending, Header:=xlGuess, OrderCustom:=1, MatchCase:= _
False, Orientation:=xlTopToBottom, DataOption1:=xlSortNormal, DataOption2 _
:=xlSortNormal
Columns("A:B").AutoFit
Range("A1").Select
End Sub

'*****
Function SelArquivo(NomedoArquivo As String) As String
    Dim result As Integer
    With Application.FileDialog(msoFileDialogFilePicker)
        .Title = "Selecione o Arquivo"
        .Filters.Add "Ficheiro texto", "*.txt", 1
        .AllowMultiSelect = False
        .InitialFileName = "E:\_MONICA\FINAL\" & NomedoArquivo
        result = .Show
    If (result <> 0) Then
        SelArquivo = Trim(.SelectedItems.Item(1))
    End If
    End With
End Function

'*****
Function AchaNaLinhaPelaDireita(LinhaDeTexto As String, oQue As String) As Integer
    Dim Procurado As Integer
    For Procurado = Len(LinhaDeTexto) - Len(oQue) + 1 To 1 Step -1
        If Mid(LinhaDeTexto, Procurado, Len(oQue)) = oQue Then
            AchaNaLinhaPelaDireita = Procurado
            Exit Function
        End If
    Next Procurado
End Function

'*****
Function AchaNaLinhaPelaEsquerda(LinhaDeTexto As String, oQue As String) As Integer
    Dim Procurado As Integer
    For Procurado = 1 To Len(LinhaDeTexto) - Len(oQue) + 1
        If Mid(LinhaDeTexto, Procurado, Len(oQue)) = oQue Then
            AchaNaLinhaPelaEsquerda = Procurado
            Exit Function
        End If
    Next Procurado
End Function

'*****
Function TagsNaLinha(LinhaDeTexto As String) As String
    TagsNaLinha = Trim(Mid(LinhaDeTexto, AchaNaLinhaPelaDireita(LinhaDeTexto, "]" ) +
2))
End Function

'*****
Function PalavraNaLinha(LinhaDeTexto As String) As String
    PalavraNaLinha = Trim(Left(LinhaDeTexto, AchaNaLinhaPelaEsquerda(LinhaDeTexto, "
") - 1))
End Function

'*****
Sub CriaVetorPi()
' Macro gravada em 10/8/2006 por Monica Jordan & Paulo C. Pereira
' Atalho do teclado: Ctrl+Shift+P

```

```

Dim VetorPi() As Single, PseudoPi() As Long
Dim i As Integer
Dim ff As Long, Somatorio As Long
Dim LinhaAtual As String
Dim LinhaAnterior As String
Dim TagAtual As String, LinhaDePi As String
Dim Arquivo As String, TxtVetorPi As String
'Computa PSEUDO VETOR PI:
If N <= 0 Then Exit Sub 'só para checar que há "estados"
ReDim PseudoPi(N)
For i = 0 To N
    PseudoPi(i) = 0
Next i
Arquivo = SelArquivo("concatenados.txt")
ff = FreeFile
LinhaAnterior = "."
Open Arquivo For Input As *ff
Do While Not EOF(ff)
    Line Input *ff, LinhaAtual
    If (Not LinhaAnterior Like "]*") And (LinhaAtual Like "]*") Then
        TagAtual = TagsNaLinha(LinhaAtual)
        For i = 0 To N
            If Indice(i) = TagAtual Then
                PseudoPi(i) = PseudoPi(i) + 1
            End If
        Next i
    End If
    LinhaAnterior = LinhaAtual
Loop
Close *ff
'Computa VETOR PI e monta string para saida em ficheiro:
ReDim VetorPi(N)
TxtVetorPi = "{"
For i = 0 To N
    Somatorio = Somatorio + PseudoPi(i)
Next i
LinhaDePi = ""
For i = 0 To N
    VetorPi(i) = PseudoPi(i) / Somatorio
    If Len(LinhaDePi) > ColunasporLinha Then
        LinhaDePi = LinhaDePi & Chr(10)
        TxtVetorPi = TxtVetorPi & LinhaDePi
        LinhaDePi = ""
    End If
    If i < N Then
        LinhaDePi = LinhaDePi & VetorPi(i) & ","
    Else
        LinhaDePi = LinhaDePi & VetorPi(i)
    End If
Next i
TxtVetorPi = TxtVetorPi & LinhaDePi & "}"
'Transpõe valores para planilhas
If SaidaParaPlanilha Then
    'Contagem:
    Sheets.Add
    ActiveSheet.Name = "ContagemPi"
    Range("A1").Activate
    For i = 0 To N
        ActiveCell.Offset(i).Value = Indice(i)
    Next i
    For i = 0 To N
        ActiveCell.Offset(i, 1).Value = PseudoPi(i)
    Next i
    'Vetor Pi:
    Sheets.Add
    ActiveSheet.Name = "VetorPI"
    Range("A1").Activate
    For i = 0 To N

```

```

        ActiveCell.Offset(i).Value = Indice(i)
        ActiveCell.Offset(i, 1).Value = VetorPi(i)
    Next i
End If
'Escreve ficheiro texto:
If SaidaParaTxt Then
    ff = FreeFile
    Open "VetorPi.txt" For Output As *ff ' Abre o arquivo para saída.
    Print *ff, TxtVetorPi
    Close *ff ' Fecha o arquivo.
End If
End Sub

'*****
Sub CriaMatrizA()
' Macro gravada em 8/8/2006 por Monica Jordan & Paulo C. Pereira
' Atalho do teclado: Ctrl+Shift+A
Dim PseudoMatrizA() As Single
Dim PrimeiraVez As Boolean
Dim i As Integer, j As Integer, k As Integer
Dim MeioN As Integer
Dim ff As Long, SomaDaLinha As Long, numeracao As Long
Dim LinhaDeTexto As String, LinhaDeA As String
Dim Origem As String, Destino As String
Dim Arquivo As String, TxtMatrizA As String
If N < 0 Then Exit Sub 'só para checar que há "estados"
'Computa PSEUDO MATRIZ A:
numeracao = 0
ReDim PseudoMatrizA(N, N)
For i = 0 To N
    For j = 0 To N
        PseudoMatrizA(i, j) = 0
    Next j
Next i
Arquivo = SelArquivo("concatenados.txt")
ff = FreeFile
Open Arquivo For Input As *ff
PrimeiraVez = True
Do While Not EOF(ff)
    numeracao = numeracao + 1
    If PrimeiraVez Then
        Line Input *ff, LinhaDeTexto
        Destino = TagsNaLinha(LinhaDeTexto)
        For k = 0 To N
            If Indice(k) = Destino Then j = k
        Next k
        PrimeiraVez = False
    End If
    Line Input *ff, LinhaDeTexto
    If LinhaDeTexto Like "*" Then
        Origem = Destino
        Destino = TagsNaLinha(LinhaDeTexto)
        i = j
        For k = 0 To N
            If Indice(k) = Destino Then j = k
        Next k
        PseudoMatrizA(i, j) = PseudoMatrizA(i, j) + 1
    Else
        PrimeiraVez = True
    End If
Loop
Close *ff
'Computa MATRIZ A e monta string para saída em ficheiro:
ReDim MatrizA(N, N)
TxtMatrizA = "{"
LinhaDeA = ""
For i = 0 To N
    SomaDaLinha = 0

```

```

For j = 0 To N
    SomaDaLinha = SomaDaLinha + PseudoMatrizA(i, j)
Next j
For j = 0 To N
    If SomaDaLinha > 0 Then
        MatrizA(i, j) = PseudoMatrizA(i, j) / SomaDaLinha
    Else
        MatrizA(i, j) = 0
    End If
    If Len(LinhaDeA) > ColunasporLinha Then
        LinhaDeA = LinhaDeA & Chr(10)
        TxtMatrizA = TxtMatrizA & LinhaDeA
        LinhaDeA = ""
    End If
    Select Case j
        Case 0: LinhaDeA = LinhaDeA & "{" & MatrizA(i, j) & ","
        Case 1 To N - 1: LinhaDeA = LinhaDeA & MatrizA(i, j) & ","
        Case N And i < N: LinhaDeA = LinhaDeA & MatrizA(i, j) & "},"
        Case N And i = N: LinhaDeA = LinhaDeA & MatrizA(i, j) & "}"
    End Select
Next j
Next i
TxtMatrizA = TxtMatrizA & LinhaDeA & "}"
'Transpõe valores para planilhas
If SaidaParaPlanilha = True Then
    'Contagens:
    MeioN = N / 2
    Sheets.Add
    ActiveSheet.Name = "1ContagemA"
    Range("A1").Activate
    For k = 0 To N
        ActiveCell.Offset(k + 1).Value = Indice(k)
    Next k
    For k = 0 To MeioN
        ActiveCell.Offset(0, k + 1).Value = Indice(k)
    Next k
    Range("B2").Activate
    For i = 0 To N
        For j = 0 To MeioN
            ActiveCell.Offset(i, j).Value = PseudoMatrizA(i, j)
        Next j
    Next i
    Range("A1").Activate
    Sheets.Add
    ActiveSheet.Name = "2ContagemA"
    Range("A1").Activate
    For k = 0 To N
        ActiveCell.Offset(k + 1).Value = Indice(k)
    Next k
    For k = MeioN + 1 To N
        ActiveCell.Offset(0, k - MeioN).Value = Indice(k)
    Next k
    Range("B2").Activate
    For i = 0 To N
        For j = MeioN + 1 To N
            ActiveCell.Offset(i, j - MeioN - 1).Value = PseudoMatrizA(i, j)
        Next j
    Next i
    Range("A1").Activate
    'MATRIZ A:
    Sheets.Add
    Range("A1").Activate
    ActiveSheet.Name = "1MatrizA"
    Range("A1").Activate
    For k = 0 To N
        ActiveCell.Offset(k + 1).Value = Indice(k)
    Next k
    For k = 0 To MeioN

```

```

        ActiveCell.Offset(0, k + 1).Value = Indice(k)
    Next k
    Range("B2").Activate
    For i = 0 To N
        For j = 0 To MeioN
            ActiveCell.Offset(i, j).Value = MatrizA(i, j)
        Next j
    Next i
    Range("A1").Activate
    Sheets.Add
    ActiveSheet.Name = "2MatrizA"
    Range("A1").Activate
    For k = 0 To N
        ActiveCell.Offset(k + 1).Value = Indice(k)
    Next k
    For k = MeioN + 1 To N
        ActiveCell.Offset(0, k - MeioN).Value = Indice(xk)
    Next k
    Range("B2").Activate
    For i = 0 To N
        For j = MeioN + 1 To N
            ActiveCell.Offset(i, j - MeioN - 1).Value = MatrizA(i, j)
        Next j
    Next i
    Range("A1").Activate
End If
'Escreve ficheiro texto:
If SaidaParaTxt = True Then
    ff = FreeFile
    Open "_MatrizA.txt" For Output As *ff ' Abre o arquivo para saída.
    Print *ff, TxtMatrizA
    Close *ff ' Fecha o arquivo.
End If
End Sub

'*****
Sub Cria_MatrizB1_MatrizDeBusca()
' Macro gravada em 9/9/2006 por Monica Jordan & Paulo C. Pereira
' Atalho do teclado: Ctrl+Shift+B
Dim Achou As Boolean
Dim i As Integer, j As Integer, Onde As Integer, Valor As Integer, x As Integer
Dim nCasos As Integer, dimensao2 As Integer
Dim ff As Long, TotalDePalavras As Long
Dim Probabilidade As Double
Dim LinhaDeTexto As String, TagAtual As String
Dim Arquivo As String, TxtMatriz As String, TxtAux As String, LinhaAux As String
Dim MatrizB1() As String, MatrizB2() As String, Aux() As String, Lista() As
String

If N < 0 Then Exit Sub 'só para checar que há "estados"
ReDim MatrizB2(N) 'redefine a matriz conforme classes.
ReDim MatrizB1(N, 1) 'Idem, com coluna0 p/ n. palavras, colunai p/ as palavras
For i = 0 To N
    MatrizB1(i, 0) = "-1"
Next i
dimensao2 = 0
TotalDePalavras = 0
Arquivo = SelArquivo("concatenados.txt")
ff = FreeFile
Open Arquivo For Input As *ff
Do While Not EOF(ff)
    Line Input *ff, LinhaDeTexto
    If LinhaDeTexto Like "*]*" Then
        TagAtual = TagsNaLinha(LinhaDeTexto)
        For i = 0 To N
            If TagAtual = Indice(i) Then
                Achou = False
                TotalDePalavras = TotalDePalavras + 1
            End If
        Next i
    End If

```

```

Aux() = Split(MatrizB2(i), ",")
For j = 0 To UBound(Aux())
  If LCase(PalavraNaLinha(LinhaDeTexto)) = LCase(Aux(j)) Then
    Achou = True
    Onde = j
    If PalavraNaLinha(LinhaDeTexto) <> Aux(j) Then 'maiuscula x
minúsculas, vence minúsculas
      Aux(j) = UCase(Aux(j))
      MatrizB2(1) = Join(Aux(), ",")
    End If
  End If
Next j
If Not Achou Then
  nCasos = CInt(MatrizB1(i, 0)) + 1 'incrementa contador de palavras por
classe
  MatrizB1(i, 0) = CStr(nCasos)
  If nCasos > dimensao2 Then dimensao2 = nCasos 'computa max. de palavras
por classe
  If MatrizB2(i) <> "" Then
    MatrizB1(i, 1) = MatrizB1(i, 1) + ","
    MatrizB2(i) = MatrizB2(i) + ","
  End If
  MatrizB2(i) = MatrizB2(i) + PalavraNaLinha(LinhaDeTexto)
  MatrizB1(i, 1) = MatrizB1(i, 1) + "1"
Else
  Aux() = Split(MatrizB1(i, 1), ",")
  Valor = CInt(Aux(Onde)) + 1
  Aux(Onde) = CStr(Valor)
  MatrizB1(i, 1) = Join(Aux(), ",")
End If
End If
Next i
End If
Loop
Close *ff
TxtMatriz = "[" + CStr(N + 1) + "]"[" + CStr(dimensao2 + 1) + "]={ + Chr(10)
For i = 0 To N
  Aux() = Split(MatrizB1(i, 1), ",")
  TxtAux = ""
  For j = 0 To UBound(Aux())
    Probabilidade = CInt(Aux(j)) / TotalDePalavras
    TxtAux = TxtAux + CStr(Probabilidade)
    If UBound(Aux()) > 0 And j < UBound(Aux()) Then
      TxtAux = TxtAux + ","
      If j > 0 And ((j + 1) / nElemPorLinha - Int((j + 1) / nElemPorLinha)) = 0
Then
        TxtAux = TxtAux + Chr(10)
      End If
    End If
  Next j
  TxtMatriz = TxtMatriz + "{" + MatrizB1(i, 0) + "," + TxtAux + "}," + Chr(10)
Next i
TxtMatriz = Left(TxtMatriz, Len(TxtMatriz) - 2) + Chr(10) + "};"
'Escreve ficheiro texto:
If SaidaParaTxt Then
  ff = FreeFile
  Open "_MatrizB1.txt" For Output As *ff ' Abre o arquivo para saída.
  Print *ff, TxtMatriz
  Close *ff ' Fecha o arquivo.
End If
TxtMatriz = "{" + Chr(10)
For i = 0 To N
  Aux() = Split(MatrizB2(i), ",")
  TxtAux = ""
  LinhaAux = ""
  For j = 0 To UBound(Aux())
    LinhaAux = LinhaAux + Chr(34) + Aux(j) + Chr(34)
    If UBound(Aux()) > 0 And j < UBound(Aux()) Then

```

```

        LinhaAux = LinhaAux + ","
        If Len(LinhaAux) > ColunasporLinha Then
            TxtAux = TxtAux + LinhaAux + Chr(10)
            LinhaAux = ""
        End If
    Next j
    TxtMatriz = TxtMatriz + "{" + TxtAux + LinhaAux + "}," + Chr(10)
Next i
TxtMatriz = Left(TxtMatriz, Len(TxtMatriz) - 2) + Chr(10) + ";";
TxtMatriz = Replace(TxtMatriz, "=", " ") 'substitui os sinais de igual inseridos
pelo etiquetador por espaços
'Escreve ficheiro texto:
If SaidaParaTxt Then
    ff = FreeFile
    Open "_MatrizB2.txt" For Output As *ff ' Abre o arquivo para saída.
    Print *ff, TxtMatriz
    Close *ff ' Fecha o arquivo.
End If
'Montagem da lista das diferentes palavras...
TxtAux = ","
For i = 0 To N
    Aux() = Split(LCase(MatrizB2(i)), ",")
    For j = 0 To UBound(Aux())
        If Not TxtAux Like "*" + Aux(j) + ",*" Then
            TxtAux = TxtAux + Aux(j) + ","
        End If
    Next j
Next i
TxtAux = Mid(TxtAux, 2, Len(TxtAux) - 2)
Lista() = Split(TxtAux, ",")
'Montagem da MatrizDeBusca...
TxtAux = ""
dimensao2 = 0
For x = 0 To UBound(Lista())
    LinhaAux = ""
    nCasos = -1
    For i = 0 To N
        Aux() = Split(LCase(MatrizB2(i)), ",")
        For j = 0 To UBound(Aux())
            If Lista(x) = Aux(j) Then
                LinhaAux = LinhaAux + "," + CStr(i) + "," + CStr(j)
                nCasos = nCasos + 1
                If nCasos > dimensao2 Then dimensao2 = nCasos
            End If
        Next j
    Next i
    TxtAux = TxtAux + "{" + Chr(34) + Lista(x) + Chr(34) + "," + CStr(nCasos) +
LinhaAux + "}"
    If x < UBound(Lista()) Then
        TxtAux = TxtAux + "," + Chr(10)
    Else
        TxtAux = TxtAux + Chr(10) + ";";
    End If
Next x
TxtMatriz = "[" + CStr(UBound(Lista()) + 1) + "]" + CStr(dimensao2 + 1) + "]={"
+ Chr(10) +
    Replace(TxtAux, "=", " ") 'substitui os sinais de igual inseridos pelo
etiquetador por espaços
'Escreve ficheiro texto:
If SaidaParaTxt Then
    ff = FreeFile
    Open "_MatrizDeBusca.txt" For Output As *ff ' Abre o arquivo para saída.
    Print *ff, TxtMatriz
    Close *ff ' Fecha o arquivo.
End If
End Sub

```


APÊNDICE III

QUESTIONÁRIO ENTREGUE ÀS CRIANÇAS VOLUNTARIAS APÓS A

DIGITAÇÃO DO TEXTO “A LAGARTA”

Após digitação do texto sem a predição:

1. Como foi escrever essa historinha?



CHATO



BOM

2. Você cansou?



SIM



NÃO

3. Você gostou?



NÃO



SIM

Após digitação do texto com a predição:

1. E agora, como foi escrever a historinha?



RUIM



MELHOR

2. Você cansou menos que da primeira vez?



NÃO



SIM

3. Você gostou mais?



NÃO



SIM

APÊNDICE IV

CÓDIGO FONTE DO ALGORITMO DE PREDIÇÃO

```
/*
Teclado Virtual Livre
Copyright (C) 2005 Percy Nohama <percy@ppgia.pucpr.br>

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
*/

// TecladoVirtualDlg.cpp : implementation file
//
#include "stdafx.h"
#include "TecladoVirtual.h"
#include "TecladoVirtualDlg.h"
#include "OpcaoDlg.h"
#include "UsuarioDlg.h"

// Sintetizador de voz
#include <mmsystem.h>
#include <initguid.h>
#include <objerror.h>

#ifdef DIRECTSOUND
#include <dsound.h>
#endif // DIRECTSOUND

#include "c:\speech.h"

ITTSCentral *pITTSCentral;
IAudioMultiMediaDevice *pIAMM;

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
////
//***** MATRIZEz E VARIÁVEIS DECLARADAS PARA A PREDIÇÃO POR HMM*****
const double VetorPi[NumeroDeClasses] =
//Cola-se aqui a Matriz Pi criada pelas rotinas Visual Basic do Excel

const double MatrizA[NumeroDeClasses][NumeroDeClasses] =
//Cola-se aqui a Matriz A criada pelas rotinas Visual Basic do Excel

struct NProvaveisPalavras{
    int NPalavras;
    double Probabilidade [maxNumPalavrasPorClasse];
};
```

```

const struct NProvaveisPalavras MatrizB1[NumeroDeClasses] =
//Cola-se aqui a Matriz B1 criada pelas rotinas Visual Basic do Excel

struct ClPoPrB
{
    int indClasse;
    int PosiB;
};

const struct MatrizInvertida
{
    char Palavra [maiorPalavraDoCorpus];
    int nClasses;
    ClPoPrB parB [numClassesDaPalavra];
} MatrizDeBusca[numPalavrasDiferentes] =
//Cola-se aqui a Matriz de Busca criada pelas rotinas Visual Basic do Excel

////////////////////////////////////
///// Se houve necessidade de alteração do tamanho das listas as seguir, alterar
o valor di fimDaLista TecladoVirtualDlg.h

struct LinhaPred
{
    int LinMtzBusca;
    double Resultado;
};

struct ListaDePredicao
{
    struct LinhaPred lstPred[fimDaLista+1];
    int classe[numClassesDaPalavra];
    double delta[numClassesDaPalavra];
} mtzPred[nPalavrasNoParagrafo+1];

int indPred=0;

////////////////////////////////////
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();
    // Dialog Data
    //{{AFX_DATA(CAboutDlg)
    enum { IDD = IDD_ABOUTBOX };
    //}}AFX_DATA

    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CAboutDlg)
protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
    //}}AFX_VIRTUAL

// Implementation
protected:
    //{{AFX_MSG(CAboutDlg)
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
    //{{AFX_DATA_INIT(CAboutDlg)
    //}}AFX_DATA_INIT
}
//*****

```

```

****
void CAboutDlg::DoDataExchange (CDataExchange* pDX)
{
    //*****
    // Associa o ID dos componentes gráficos com a variável *
    //*****
    CDialog::DoDataExchange (pDX);
    //{{AFX_DATA_MAP(CAboutDlg)
    //}}AFX_DATA_MAP
}
BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
    //{{AFX_MSG_MAP(CAboutDlg)
    ON_WM_DESTROY()
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CTecladoVirtualDlg dialog
CTecladoVirtualDlg::CTecladoVirtualDlg(CWnd* pParent /*=NULL*/)
: CDialog(CTecladoVirtualDlg::IDD, pParent)
{
    //{{AFX_DATA_INIT(CTecladoVirtualDlg)
    m_vedit1 = _T("");
    m_veditContaClique = 0;
    m_veditTempoClique = 0;
    //}}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon in Win32
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}
//*****
****
void CTecladoVirtualDlg::DoDataExchange (CDataExchange* pDX)
{
    //*****
    // Associa o ID dos componentes gráficos com a variável *
    //*****
    CDialog::DoDataExchange (pDX);
    //{{AFX_DATA_MAP(CTecladoVirtualDlg)
    DDX_Control(pDX, IDC_BUTTON44, m_cButtonTempoClique);
    DDX_Control(pDX, IDC_BUTTON10, m_cButtonContaClique);
    DDX_Control(pDX, IDC_STATIC5, m_cStaticContaClique);
    DDX_Control(pDX, IDC_STATIC4, m_cStaticTempoClique);
    DDX_Control(pDX, IDC_EDIT3, m_cEditTempoClique);
    DDX_Control(pDX, IDC_EDIT4, m_cEditContaClique);
    DDX_Text(pDX, IDC_EDIT3, m_veditContaClique);
    DDX_Text(pDX, IDC_EDIT4, m_veditTempoClique);
    DDX_Control(pDX, IDC_STATIC3, m_focus);
    DDX_Control(pDX, IDC_BUTTON8, m_b8);
    DDX_Control(pDX, IDC_BUTTON9, m_b9);
    DDX_Control(pDX, IDC_BUTTON7, m_b7);
    DDX_Control(pDX, IDC_LIST3, m_clist3);
    DDX_Control(pDX, IDC_LIST1, m_clist1);
    DDX_Control(pDX, IDC_EDIT1, m_cedit1);
    DDX_Control(pDX, IDCANCEL, m_bfechar);
    DDX_Control(pDX, IDC_BUTTON1, m_b1);
    DDX_Control(pDX, IDC_BUTTON2, m_b2);
    DDX_Control(pDX, IDC_BUTTON3, m_b3);
    DDX_Control(pDX, IDC_BUTTON4, m_b4);
    DDX_Control(pDX, IDC_BUTTON5, m_b5);
    DDX_Control(pDX, IDC_BUTTON11, m_b11);
    DDX_Control(pDX, IDC_BUTTON12, m_b12);
    DDX_Control(pDX, IDC_BUTTON13, m_b13);
    DDX_Control(pDX, IDC_BUTTON14, m_b14);
    DDX_Control(pDX, IDC_BUTTON15, m_b15);
    DDX_Control(pDX, IDC_BUTTON16, m_b16);
    DDX_Control(pDX, IDC_BUTTON17, m_b17);
    DDX_Control(pDX, IDC_BUTTON18, m_b18);
    DDX_Control(pDX, IDC_BUTTON19, m_b19);
    DDX_Control(pDX, IDC_BUTTON20, m_b20);
}

```

```

DDX_Control(pDX, IDC_BUTTON21, m_b21);
DDX_Control(pDX, IDC_BUTTON22, m_b22);
DDX_Control(pDX, IDC_BUTTON23, m_b23);
DDX_Control(pDX, IDC_BUTTON24, m_b24);
DDX_Control(pDX, IDC_BUTTON25, m_b25);
DDX_Control(pDX, IDC_BUTTON26, m_b26);
DDX_Control(pDX, IDC_BUTTON27, m_b27);
DDX_Control(pDX, IDC_BUTTON28, m_b28);
DDX_Control(pDX, IDC_BUTTON29, m_b29);
DDX_Control(pDX, IDC_BUTTON30, m_b30);
DDX_Control(pDX, IDC_BUTTON31, m_b31);
DDX_Control(pDX, IDC_BUTTON32, m_b32);
DDX_Control(pDX, IDC_BUTTON33, m_b33);
DDX_Control(pDX, IDC_BUTTON34, m_b34);
DDX_Control(pDX, IDC_BUTTON35, m_b35);
DDX_Control(pDX, IDC_BUTTON36, m_b36);
DDX_Control(pDX, IDC_BUTTON37, m_b37);
DDX_Control(pDX, IDC_BUTTON38, m_b38);
DDX_Control(pDX, IDC_BUTTON39, m_b39);
DDX_Control(pDX, IDC_BUTTON40, m_b40);
DDX_Control(pDX, IDC_BUTTON41, m_b41);
DDX_Control(pDX, IDC_BUTTON42, m_b42);
DDX_Control(pDX, IDC_BUTTON43, m_b43);
DDX_Text(pDX, IDC_EDIT1, m_vedit1);
//}}AFX_DATA_MAP
}
//*****
****
BEGIN_MESSAGE_MAP(CTecladoVirtualDlg, CDialog)
//*****
//* Associa o ID dos componentes gráficos com a função *
//*****
//{{AFX_MSG_MAP(CTecladoVirtualDlg)
ON_WM_SYSCOMMAND()
ON_WM_PAINT()
ON_WM_QUERYDRAGICON()
ON_BN_CLICKED(IDC_BUTTON10, OnContaClique)
ON_BN_CLICKED(IDC_BUTTON44, OnTempoClique)
ON_LBN_SELCHANGE(IDC_LIST1, OnHistorico)
ON_LBN_SELCHANGE(IDC_LIST3, OnBotao)
ON_BN_CLICKED(IDC_BUTTON1, OnBotao)
ON_BN_CLICKED(IDC_BUTTON2, OnBotao)
ON_BN_CLICKED(IDC_BUTTON3, OnBotao)
ON_BN_CLICKED(IDC_BUTTON4, OnBotao)
ON_BN_CLICKED(IDC_BUTTON5, OnBotao)
ON_BN_CLICKED(IDC_BUTTON7, OnBotao)
ON_BN_CLICKED(IDC_BUTTON8, Outras)
ON_BN_CLICKED(IDC_BUTTON9, Outras)
ON_BN_CLICKED(IDC_BUTTON11, OnBotao)
ON_BN_CLICKED(IDC_BUTTON12, OnBotao)
ON_BN_CLICKED(IDC_BUTTON13, OnBotao)
ON_BN_CLICKED(IDC_BUTTON14, OnBotao)
ON_BN_CLICKED(IDC_BUTTON15, OnBotao)
ON_BN_CLICKED(IDC_BUTTON16, OnBotao)
ON_BN_CLICKED(IDC_BUTTON17, OnBotao)
ON_BN_CLICKED(IDC_BUTTON18, OnBotao)
ON_BN_CLICKED(IDC_BUTTON19, OnBotao)
ON_BN_CLICKED(IDC_BUTTON20, OnBotao)
ON_BN_CLICKED(IDC_BUTTON21, OnBotao)
ON_BN_CLICKED(IDC_BUTTON22, OnBotao)
ON_BN_CLICKED(IDC_BUTTON23, OnBotao)
ON_BN_CLICKED(IDC_BUTTON24, OnBotao)
ON_BN_CLICKED(IDC_BUTTON25, OnBotao)
ON_BN_CLICKED(IDC_BUTTON26, OnBotao)
ON_BN_CLICKED(IDC_BUTTON27, OnBotao)
ON_BN_CLICKED(IDC_BUTTON28, OnBotao)
ON_BN_CLICKED(IDC_BUTTON29, OnBotao)
ON_BN_CLICKED(IDC_BUTTON30, OnBotao)

```

```

ON_BN_CLICKED(IDC_BUTTON31, OnBotao)
ON_BN_CLICKED(IDC_BUTTON32, OnBotao)
ON_BN_CLICKED(IDC_BUTTON33, OnBotao)
ON_BN_CLICKED(IDC_BUTTON34, OnBotao)
ON_BN_CLICKED(IDC_BUTTON35, OnBotao)
ON_BN_CLICKED(IDC_BUTTON36, OnBotao)
ON_BN_CLICKED(IDC_BUTTON37, OnBotao)
ON_BN_CLICKED(IDC_BUTTON38, OnBotao)
ON_BN_CLICKED(IDC_BUTTON39, OnBotao)
ON_BN_CLICKED(IDC_BUTTON40, OnBotao)
ON_BN_CLICKED(IDC_BUTTON41, OnBotao)
ON_BN_CLICKED(IDC_BUTTON42, OnBotao)
ON_BN_CLICKED(IDC_BUTTON43, OnBotao)
ON_BN_CLICKED(IDC_BUTTON43, OnBotao)
ON_LBN_SELCHANGE(IDCANCEL, OnBotao)
ON_WM_TIMER()
ON_WM_DESTROY()
//}}AFX_MSG_MAP
END_MESSAGE_MAP()
////////////////////////////////////
// CTecladoVirtualDlg message handlers
//*****
****
BOOL CTecladoVirtualDlg::OnInitDialog()
{
    CDialog::OnInitDialog();
    // Add "About..." menu item to system menu.
    // IDM_ABOUTBOX must be in the system command range.
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);
    CMenu* pSysMenu = GetSystemMenu(false);
    if (pSysMenu != NULL)
    {
        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }
    // Set the icon for this dialog. The framework does this automatically
    // when the application's main window is not a dialog
    SetIcon(m_hIcon, true); // Set big icon
    SetIcon(m_hIcon, false); // Set small icon

    //*****
    // Inicio do Teclado *
    //*****

    /** Conta Clique
    altContaClique=50;

    //Cálculo do VetorPi
    Predicao();

    //Identifica usuário
    ConfUsuarios("");

    //Redimensiona a tela
    CRect rect,rect2;
    GetWindowRect( rect );
    GetDesktopWindow()->GetWindowRect( &rect2);

    int topo=(rect2.Height()/2)-(rect.Height()/2);
    int esq=(rect2.Width()/2)-(rect.Width()/2);

```

```

::SetWindowPos(m_hWnd,HWND_TOPMOST,esq,topo,rect.Width(),rect.Height(),SWP_SHOWWIND
OW);
//Abre as configurações
m_dir=1;
h=10,v=10;
if(config.varredura==true)SetTimer(1,config.velo*100,0);
return true; // return true unless you set the focus to a control
}
//*****
****
void CTecladoVirtualDlg::OnDestroy()
{
//*****
// sai do aplicativo *
//*****
CDialog::OnDestroy();
// Fecha o sintetizador
SintetizadorFechar();
// Salva a configuração
ConfSalvar();
}
//*****
****
void CTecladoVirtualDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
if ((nID & 0xFFFF) == IDM_ABOUTBOX)
{
CAboutDlg dlgAbout;
dlgAbout.DoModal();
}
else
{
CDialog::OnSysCommand(nID, lParam);
}
}
// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.
//*****
****
void CTecladoVirtualDlg::OnPaint()
{
if (IsIconic())
{
CPaintDC dc(this); // device context for painting
SendMessage(WM_ICONERASEBKGND, (LPARAM) dc.GetSafeHdc(), 0);
// Center icon in client rectangle
int cxIcon = GetSystemMetrics(SM_CXICON);
int cyIcon = GetSystemMetrics(SM_CYICON);
CRect rect;
GetClientRect(&rect);
int x = (rect.Width() - cxIcon + 1) / 2;
int y = (rect.Height() - cyIcon + 1) / 2;
// Draw the icon
dc.DrawIcon(x, y, m_hIcon);
}
else
{
CDialog::OnPaint();
}
}
// The system calls this to obtain the cursor to display while the user drags
// the minimized window.
HCURSOR CTecladoVirtualDlg::OnQueryDragIcon()
{
return (HCURSOR) m_hIcon;
}
//*****
****

```

```

****
BOOL CTecladoVirtualDlg::PreTranslateMessage(MSG* pMsg)
{
    //*****
    // Captura o clique do teclado *
    //*****
    // Exibe a ajuda
    if(config.varredura==false && pMsg->message == WM_KEYDOWN && pMsg->wParam==
VK_F1){
        WinExec("hh.exe \"./AjudaTeclado.chm\" ,SW_SHOW);
    }
    // Toque no teclado
    if(config.varredura==true){
        if (pMsg->message == WM_CHAR && m_dir==2) {
            KillTimer(1);
            m_dir=0;
            m_clist1.SetFocus();
            m_dir=1;
            SetTimer(1,config.velo*100,0);
        }
        // Aciona o clique do mouse
        if(v!=15){
            mouse_event(MOUSEEVENTF_LEFTDOWN, 0, 0, 0, 0);
            mouse_event(MOUSEEVENTF_LEFTUP, 0, 0, 0, 0);
        }
        else OnBotao();
        h=10;
        v=10;
        CRect r;
        GetWindowRect(r);
        SetCursorPos(r.left,r.top);
        m_clist1.SetFocus();
    }
    else if (pMsg->message == WM_CHAR && m_dir==1) {
        // Altera o sentido da varredura
        m_dir=2;
    }
    }
    return CDialog::PreTranslateMessage(pMsg);
}
//*****
****
void CTecladoVirtualDlg::Interface(HWND hParentWnd, CRect &rect)
{
    //*****
    // Organiza as teclas na tela *
    //*****
    int d=2; // Recuo direito
    int j=2; // Recuo de topo
    int m=39; // Largura do botao
    int n=25; // Altura do botao
    int e=32;
    int x=2;
    int alt=250;
    int lar=590;
    // Título da janela
    SetWindowText("Teclado Virtual Livre - Usuário:"+m_usuario);
    // Tamanho das letras
    LOGFONT lf;memset(&lf, 0, sizeof(lf));lf.lfHeight = -11;
    LOGFONT lf2;memset(&lf2, 0, sizeof(lf2));lf2.lfHeight = -16;
    // Ajusta o tamanho das listas
    if (config.tamanho==1){
        d=2; j=2; m=29; n=19; e=24; x=2;
        lf.lfHeight =-9;
        lf2.lfHeight =-11;
        if(config.historico==1 ){
            // (NULL,recuo esq,topo,largura,alt)
            m_clist1.SetWindowPos(NULL,2,2,320,61,NULL);
            m_clist3.SetWindowPos(NULL,322,2,113,142,NULL);
        }
    }
}

```

```

    m_cedit1.SetWindowPos (NULL, 2, 65, 320, 18, NULL);
    j=87;
    alt=190+altContaClique;
    lar=442;
}
else {
    //(NULL,recuo esq,topo,largura,alt)
    m_clist1.SetWindowPos (NULL, 2, 2, 0, 0, NULL);
    m_clist3.SetWindowPos (NULL, 322, 2, 113, 73, NULL);
    m_cedit1.SetWindowPos (NULL, 2, 2, 320, 18, NULL);
    j=22;
    alt=125+altContaClique;
    lar=442;
}
}
if (config.tamanho==2) {
    d=2; j=2; m=39; n=25; e=32; x=2;
    lf.lfHeight = -11;
    lf2.lfHeight = -16;
    if(config.historico==1) {
        //(NULL,recuo esq,topo,largura,alt)
        m_clist1.SetWindowPos (NULL, 2, 2, 430, 83, NULL);
        m_clist3.SetWindowPos (NULL, 435, 2, 148, 193, NULL);
        m_cedit1.SetWindowPos (NULL, 2, 89, 430, 25, NULL);
        j=119;
        alt=250+altContaClique;
        lar=590;
    }
    else {
        //(NULL,recuo esq,topo,largura,alt)
        m_clist1.SetWindowPos (NULL, 2, 2, 0, 0, NULL);
        m_clist3.SetWindowPos (NULL, 435, 2, 148, 100, NULL);
        m_cedit1.SetWindowPos (NULL, 2, 2, 430, 25, NULL);
        j=29;
        alt=160+altContaClique;
        lar=590;
    }
}
if (config.tamanho==3) {
    d=2; j=2; m=49; n=31; e=40; x=2;
    lf.lfHeight = -13;
    lf2.lfHeight = -21;
    if(config.historico==1) {
        //(NULL,recuo esq,topo,largura,alt)
        m_clist1.SetWindowPos (NULL, 2, 2, 540, 105, NULL);
        m_clist3.SetWindowPos (NULL, 548, 2, 183, 244, NULL);
        m_cedit1.SetWindowPos (NULL, 2, 110, 540, 32, NULL);
        j=151;
        alt=310+altContaClique;
        lar=738;
    }
    else {
        //(NULL,recuo esq,topo,largura,alt)
        m_clist1.SetWindowPos (NULL, 2, 2, 540, 0, NULL);
        m_clist3.SetWindowPos (NULL, 548, 2, 183, 131, NULL);
        m_cedit1.SetWindowPos (NULL, 2, 2, 540, 32, NULL);
        j=37;
        alt=195+altContaClique;
        lar=738;
    }
}
}
// Conta Clique -- Ordena na tela
m_cStaticTempoClique.SetWindowPos (NULL, 20, alt-60, 50, 40, NULL);
m_cEditContaClique.SetWindowPos (NULL, 100, alt-60, 50, 20, NULL);
m_cStaticContaClique.SetWindowPos (NULL, 160, alt-60, 50, 40, NULL);
m_cEditTempoClique.SetWindowPos (NULL, 220, alt-60, 50, 20, NULL);
m_cButtonTempoClique.SetWindowPos (NULL, 300, alt-60, 50, 20, NULL);
m_cButtonContaClique.SetWindowPos (NULL, 360, alt-60, 50, 20, NULL);

```

```

// ordena as letras do teclado
if (config.formato=="abcde"){
    FormatoABC();
}
else
    FormatoQWE();
    strcpy(lf2.lfFaceName, "Times New Roman");
    strcpy(lf.lfFaceName, "Arial");
    CFont *fonte2 = new CFont();
    fonte2->CreateFontIndirect(&lf2);
    CFont *fonte = new CFont();
    fonte->CreateFontIndirect(&lf);
// Adiciona as fontes
m_clist3.SetFont(fonte2); m_clist1.SetFont(fonte2);m_cedit1.SetFont(fonte2);
m_b1.SetFont(GetFont());m_b2.SetFont(fonte);m_b3.SetFont(fonte);
m_b4.SetFont(fonte);m_b5.SetFont(fonte);m_b7.SetFont(fonte);
m_b8.SetFont(fonte);m_b9.SetFont(fonte);m_b11.SetFont(fonte);
m_b12.SetFont(fonte);m_b13.SetFont(fonte);m_b14.SetFont(fonte);
m_b15.SetFont(fonte);m_b16.SetFont(fonte);m_b17.SetFont(fonte);
m_b18.SetFont(fonte);m_b19.SetFont(fonte);m_b20.SetFont(fonte);
m_b21.SetFont(fonte);m_b22.SetFont(fonte);m_b23.SetFont(fonte);
m_b24.SetFont(fonte);m_b25.SetFont(fonte);m_b26.SetFont(fonte);
m_b27.SetFont(fonte);m_b28.SetFont(fonte);m_b29.SetFont(fonte);
m_b30.SetFont(fonte);m_b31.SetFont(fonte);m_b32.SetFont(fonte);
m_b33.SetFont(fonte);m_b34.SetFont(fonte);m_b35.SetFont(fonte);
m_b36.SetFont(fonte);m_b37.SetFont(fonte);m_b38.SetFont(fonte);
m_b39.SetFont(fonte);m_b40.SetFont(fonte);m_b41.SetFont(fonte);
m_b42.SetFont(fonte);m_b43.SetFont(fonte);m_bfechar.SetFont(fonte);
// Ajusta a posição dos botões
// Move a janela
: :MoveWindow(hParentWnd,rect.left,rect.top,lar+5,alt+5,true);
// : :MoveWindow(hParentWnd,esq ,topo,lar ,alt , true);
// Primeira fila
j+=x;
m_b1.SetWindowPos(NULL,d,j-2,m+e,n,NULL);
m_b2.SetWindowPos(NULL,d+e+m,j-2,m+e,n,NULL);
m_b3.SetWindowPos(NULL,d+2*(m+e),j-2,m+e,n,NULL);
m_b4.SetWindowPos(NULL,d+3*(m+e),j-2,m+e,n,NULL);
m_b5.SetWindowPos(NULL,d+4*(m+e),j-2,m+e,n,NULL);
// Segunda fila
j+=n+x;
m_b11.SetWindowPos(NULL,d,j,m,n,NULL);
m_b12.SetWindowPos(NULL,d+m,j,m,n,NULL);
m_b13.SetWindowPos(NULL,d+2*m,j,m,n,NULL);
m_b14.SetWindowPos(NULL,d+3*m,j,m,n,NULL);
m_b15.SetWindowPos(NULL,d+4*m,j,m,n,NULL);
m_b16.SetWindowPos(NULL,d+5*m,j,m,n,NULL);
m_b17.SetWindowPos(NULL,d+6*m,j,m,n,NULL);
m_b18.SetWindowPos(NULL,d+7*m,j,m,n,NULL);
m_b19.SetWindowPos(NULL,d+8*m,j,m,n,NULL);
m_b20.SetWindowPos(NULL,d+9*m,j,m,n,NULL);
m_b41.SetWindowPos(NULL,d+10*m,j,m,n,NULL);
// 3a.
m_b21.SetWindowPos(NULL,d,j+n,m,n,NULL);
m_b22.SetWindowPos(NULL,d+m,j+n,m,n,NULL);
m_b23.SetWindowPos(NULL,d+2*m,j+n,m,n,NULL);
m_b24.SetWindowPos(NULL,d+3*m,j+n,m,n,NULL);
m_b25.SetWindowPos(NULL,d+4*m,j+n,m,n,NULL);
m_b26.SetWindowPos(NULL,d+5*m,j+n,m,n,NULL);
m_b27.SetWindowPos(NULL,d+6*m,j+n,m,n,NULL);
m_b28.SetWindowPos(NULL,d+7*m,j+n,m,n,NULL);
m_b29.SetWindowPos(NULL,d+8*m,j+n,m,n,NULL);
m_b30.SetWindowPos(NULL,d+9*m,j+n,m,n,NULL);
m_b42.SetWindowPos(NULL,d+10*m,j+n,m,n,NULL);
// 4a.
m_b31.SetWindowPos(NULL,d,j+2*n,m,n,NULL);
m_b32.SetWindowPos(NULL,d+m,j+2*n,m,n,NULL);
m_b33.SetWindowPos(NULL,d+m*2,j+2*n,m,n,NULL);

```

```

m_b34.SetWindowPos (NULL, d+m*3, j+2*n, m, n, NULL);
m_b35.SetWindowPos (NULL, d+m*4, j+2*n, m, n, NULL);
m_b36.SetWindowPos (NULL, d+m*5, j+2*n, m, n, NULL);
m_b37.SetWindowPos (NULL, d+m*6, j+2*n, m, n, NULL);
m_b38.SetWindowPos (NULL, d+m*7, j+2*n, m, n, NULL);
m_b39.SetWindowPos (NULL, d+m*8, j+2*n, m, n, NULL);
m_b40.SetWindowPos (NULL, d+m*9, j+2*n, m, n, NULL);
m_b43.SetWindowPos (NULL, d+10*m, j+2*n, m, n, NULL);
m_b9.SetWindowPos (NULL, d+11*m, j+2*n, m, n, NULL);
m_b8.SetWindowPos (NULL, d+12*m, j+2*n, m, n, NULL);
m_b7.SetWindowPos (NULL, d+13*m, j+2*n, m, n, NULL);
m_bfechar.SetWindowPos (NULL, d+14*m, j+2*n, m, n, NULL);
}
//*****
****
void CTecladoVirtualDlg::ConfUsuarios (CString usuario)
{
//*****
// Abre as configurações do config.ini *
//*****
// Abre a janela de usuários
char buf[1000];
::GetModuleFileName (AfxGetApp ()->m_hInstance, buf, sizeof (buf));
diretorio=buf;
diretorio.Delete (diretorio.ReverseFind ('\\')+1, diretorio.GetLength ()-
diretorio.ReverseFind ('\\')-1);
HWND hWnd;
CRect rect;
// Se o usuario não for selecionado
if (usuario.GetLength ()==0) {
CUusuarioDlg a;
a.DoModal ();
if (a.m_caminho=="")
m_usuario="Padrão";
else
m_usuario=a.m_caminho;
SetWindowText ("Teclado Virtual Livre - Usuário:"+m_usuario);
hWnd=::GetActiveWindow ();
GetWindowRect (&rect);
}
else {
m_usuario=usuario;
HWND hChildWnd;
hChildWnd=::GetActiveWindow ();
hWnd=::GetParent (hChildWnd);
GetWindowRect (&rect);
}
// Abre as configurações do usuário
CFile arquivo;
CString tex;
CString caminho=m_usuario;
caminho=".\\\\"+caminho;
caminho+="\\config.ini";
TCHAR temp[500] = {0};
GetPrivateProfileString ("Configurações do Teclado
Virtual", "autoclick", "0", temp, 500, caminho);
config.autoclick=atoi (temp);
GetPrivateProfileString ("Configurações do Teclado Virtual", "delay", "4",
temp, 500, caminho);
config.delay=atoi (temp);
GetPrivateProfileString ("Configurações do Teclado Virtual", "formato",
"abcde", temp, 500, caminho);
config.formato=temp;
GetPrivateProfileString ("Configurações do Teclado Virtual", "externo",
"0", temp, 500, caminho);
config.externo=atoi (temp);
GetPrivateProfileString ("Configurações do Teclado
Virtual", "maiuscula", "0", temp, 500, caminho);

```

```

    config.maiuscula=atoi(temp);
    GetPrivateProfileString("Configurações do Teclado Virtual","tamanho", "2"
, temp, 500, caminho);
    config.tamanho=atoi(temp);
    GetPrivateProfileString("Configurações do Teclado Virtual","historico","1"
, temp, 500, caminho);
    config.historico=atoi(temp);
    //Organiza a posição das teclas na tela
    Interface(hWnd, rect);
    GetPrivateProfileString("Configurações do Teclado Virtual","som"
, "0", temp, 500, caminho);
    config.som=atoi(temp);
    GetPrivateProfileString("Configurações do Teclado Virtual","voz"
, "0", temp, 500, caminho);
    config.masculino=atoi(temp);
    GetPrivateProfileString("Configurações do Teclado
Virtual", "varredura", "0", temp, 500, caminho);
    config.varredura=atoi(temp);
    GetPrivateProfileString("Configurações do Teclado Virtual", "velo"
, "10", temp, 500, caminho);
    config.velo=atoi(temp);
    m_dir=1;
    h=10, v=10;
    if(config.varredura==1) SetTimer(1, config.velo*100, 0);
;
}
//*****
****
void CTecladoVirtualDlg::ConfSalvar()
{
    //*****
    // Salva as configurações do arquivo *
    //*****
    // Escreve as variáveis no arquivo
    TCHAR temp[500] = {0};
    CString temp2, caminho;
    caminho=diretorio;
    caminho+=m_usuario;
    caminho+="\\config.ini";
    if(config.autoclick==1) strcpy(temp, "1"); else strcpy(temp, "0");
    WritePrivateProfileString("Configurações do Teclado
Virtual", "autoclick", temp, caminho);
    temp2.Format("%d", config.delay);
    strcpy(temp, temp2);
    WritePrivateProfileString("Configurações do Teclado Virtual", "delay"
, temp, caminho);
    strcpy(temp, config.formato);
    WritePrivateProfileString("Configurações do Teclado Virtual", "formato"
, temp, caminho);
    if(config.externo==1) strcpy(temp, "1"); else strcpy(temp, "0");
    WritePrivateProfileString("Configurações do Teclado Virtual", "externo"
, temp, caminho);
    if(config.maiuscula==1) strcpy(temp, "1"); else strcpy(temp, "0");
    WritePrivateProfileString("Configurações do Teclado
Virtual", "maiuscula", temp, caminho);
    if(config.historico==0) strcpy(temp, "0"); else strcpy(temp, "1");
    WritePrivateProfileString("Configurações do Teclado
Virtual", "historico", temp, caminho);
    temp2.Format("%d", config.tamanho);
    strcpy(temp, temp2);
    WritePrivateProfileString("Configurações do Teclado Virtual", "tamanho"
, temp, caminho);
    temp2.Format("%d", config.som);
    strcpy(temp, temp2);
    WritePrivateProfileString("Configurações do Teclado Virtual", "som"
, temp, caminho);
    if(config.masculino==1) strcpy(temp, "1"); else strcpy(temp, "0");
    WritePrivateProfileString("Configurações do Teclado Virtual", "voz"

```

```

,temp,caminho);
    if(config.varredura==1)strcpy(temp,"1"); else strcpy(temp,"0");
    WritePrivateProfileString("Configurações do Teclado Virtual","varredura"
,temp,caminho);
    temp2.Format("%d",config.velo);
    strcpy(temp,temp2);
    WritePrivateProfileString("Configurações do Teclado Virtual","velo"
,temp,caminho);
}
//*****
****
void CTecladoVirtualDlg::OnHistorico()
{
    //*****
    // Clica sobre a lista de histórico *
    //*****
    // Retira da temp e adiciona na área da frase
    CString temp;
    char temp2[256];
    char valor[256];
    UpdateData(false);
    m_clist1.GetText(m_clist1.GetCurSel(),temp);
    _tcscpy(temp2,temp);
    int x=temp.ReverseFind('\\');
    if(x<0)x=15;
    int tam=temp.GetLength();
    x+=2;
    int p=0;
    for(int i=x;i<=tam;i++){
        valor[p]=temp2[i];
        p++;
    }
    LPTSTR temp4 = new TCHAR[tam];
    _tcscpy(temp4,valor);
    // Atualiza o tempo
    m_vedit1+=temp4;
    m_vedit1+="";
    UpdateData(false);
    m_cedit1.SetFocus();
    m_cedit1.SetSel(m_vedit1.GetLength(),m_vedit1.GetLength(),true);
}
//*****
****
void CTecladoVirtualDlg::OnImprimir()
{
    //*****
    // * Evento de clique no botão imprimir *
    //*****
    // Pega o texto do histórico linha por linha
    CString texto,temp;
    for(int i=0;i<=m_clist1.GetCount()-1;i++){
        m_clist1.GetText(i,temp);
    // linha com tempo
        if(temp.Find("(")==0){
            temp.Delete(0,temp.Find(")")+2);
            temp.TrimRight();
        }
    // linha sem tempo
        else{
            int b=temp.ReverseFind('\\n');
            int c=temp.GetLength();
            if(temp.ReverseFind('\\n')==temp.GetLength()-1)
                texto.Delete(temp.GetLength()-1,1);
            temp.TrimLeft();
        }
        texto+=temp;
        texto+="\\n";
    }
}

```

```

    }
    // Adiciona o texto
    texto+=m_vedit1;
    // Prepara o texto para a impressão
    CDC m_dc;
    CPrintDialog printDlg(false);
    // Janela da impressora
    if (printDlg.DoModal() == IDCANCEL) return;
    m_dc.Attach(printDlg.GetPrinterDC());
    m_dc.m_bPrinting = true;
    DOCINFO di;
    ::ZeroMemory (&di, sizeof (DOCINFO));
    di.cbSize = sizeof (DOCINFO);
    // Título do documento.
    di.lpszDocName = "Teclado Virtual";
    // Começa a imprimir
    if(m_dc.StartDoc( &di) > 0){
    // Começa a página
    m_dc.StartPage();
    m_dc.SetMapMode(MM_HIMETRIC);
    // Adiciona o texto
    CRect rectTexto;
    rectTexto.left=1000;
    rectTexto.top=-1000;
    rectTexto.right=18000;
    rectTexto.bottom =-3000;
    m_dc.DrawText(texto, &rectTexto,DT_NOPREFIX |
DT_WORDBREAK);
    // Fim da página e do documento
    m_dc.EndPage();
    m_dc.EndDoc();
    }
    // Ocorreu um erro, cancela impressão
    else m_dc.AbortDoc();
    // Acaba a impressão
    m_dc.Detach();
}
//*****
****
void CTecladoVirtualDlg::FormatoABC()
{
    //*****
    // ORDENA AS TECLAS SEGUINDO O ALFABETO - ABCDE *
    //*****
    if(config.maiuscula==1){
        m_b11.SetWindowText("A"); m_b12.SetWindowText("B"); m_b13.SetWindowText("C");
        m_b14.SetWindowText("D"); m_b15.SetWindowText("E"); m_b16.SetWindowText("F");
        m_b17.SetWindowText("G"); m_b18.SetWindowText("H"); m_b19.SetWindowText("I");
        m_b20.SetWindowText("J"); m_b41.SetWindowText(",");

        m_b21.SetWindowText("K"); m_b22.SetWindowText("L"); m_b23.SetWindowText("M");
        m_b24.SetWindowText("N"); m_b25.SetWindowText("O"); m_b26.SetWindowText("P");
        m_b27.SetWindowText("Q"); m_b28.SetWindowText("R"); m_b29.SetWindowText("S");
        m_b30.SetWindowText("^"); m_b42.SetWindowText(".");

        m_b31.SetWindowText("T"); m_b32.SetWindowText("U"); m_b33.SetWindowText("V");
        m_b34.SetWindowText("W"); m_b35.SetWindowText("X"); m_b36.SetWindowText("Y");
        m_b37.SetWindowText("Z"); m_b38.SetWindowText("`"); m_b39.SetWindowText("~");
        m_b40.SetWindowText("`"); m_b43.SetWindowText("[CAPS]");
    }else{
        m_b11.SetWindowText("a"); m_b12.SetWindowText("b"); m_b13.SetWindowText("c");
        m_b14.SetWindowText("d"); m_b15.SetWindowText("e"); m_b16.SetWindowText("f");
        m_b17.SetWindowText("g"); m_b18.SetWindowText("h"); m_b19.SetWindowText("i");
        m_b20.SetWindowText("j"); m_b41.SetWindowText(",");

        m_b21.SetWindowText("k"); m_b22.SetWindowText("l"); m_b23.SetWindowText("m");
        m_b24.SetWindowText("n"); m_b25.SetWindowText("o"); m_b26.SetWindowText("p");
        m_b27.SetWindowText("q"); m_b28.SetWindowText("r"); m_b29.SetWindowText("s");
    }
}

```

```

    m_b30.SetWindowText("^"); m_b42.SetWindowText(".");

    m_b31.SetWindowText("t"); m_b32.SetWindowText("u"); m_b33.SetWindowText("v");
    m_b34.SetWindowText("w"); m_b35.SetWindowText("x"); m_b36.SetWindowText("y");
    m_b37.SetWindowText("z"); m_b38.SetWindowText("`"); m_b39.SetWindowText("~");
    m_b40.SetWindowText("`"); m_b43.SetWindowText("[CAPS]");
}
//*****
****

void CTecladoVirtualDlg::FormatoQWE()
{
    //*****
    // ORDENA AS TECLAS SEGUINDO O TECLADO - QWERT *
    //*****
    if(config.maiuscula==1){
        m_b11.SetWindowText("Q"); m_b12.SetWindowText("W"); m_b13.SetWindowText("E");
        m_b14.SetWindowText("R"); m_b15.SetWindowText("T"); m_b16.SetWindowText("Y");
        m_b17.SetWindowText("U"); m_b18.SetWindowText("I"); m_b19.SetWindowText("O");
        m_b20.SetWindowText("P"); m_b41.SetWindowText(",");

        m_b21.SetWindowText("A"); m_b22.SetWindowText("S"); m_b23.SetWindowText("D");
        m_b24.SetWindowText("F"); m_b25.SetWindowText("G"); m_b26.SetWindowText("H");
        m_b27.SetWindowText("J"); m_b28.SetWindowText("K"); m_b29.SetWindowText("L");
        m_b30.SetWindowText("^"); m_b42.SetWindowText(".");

        m_b31.SetWindowText("Z"); m_b32.SetWindowText("X"); m_b33.SetWindowText("C");
        m_b34.SetWindowText("V"); m_b35.SetWindowText("B"); m_b36.SetWindowText("N");
        m_b37.SetWindowText("M"); m_b38.SetWindowText("`"); m_b39.SetWindowText("~");
        m_b40.SetWindowText("`"); m_b43.SetWindowText("[CAPS]");
    }else{
        m_b11.SetWindowText("q"); m_b12.SetWindowText("w"); m_b13.SetWindowText("e");
        m_b14.SetWindowText("r"); m_b15.SetWindowText("t"); m_b16.SetWindowText("y");
        m_b17.SetWindowText("u"); m_b18.SetWindowText("i"); m_b19.SetWindowText("o");
        m_b20.SetWindowText("p"); m_b41.SetWindowText(",");

        m_b21.SetWindowText("a"); m_b22.SetWindowText("s"); m_b23.SetWindowText("d");
        m_b24.SetWindowText("f"); m_b25.SetWindowText("g"); m_b26.SetWindowText("h");
        m_b27.SetWindowText("j"); m_b28.SetWindowText("k"); m_b29.SetWindowText("l");
        m_b30.SetWindowText("^"); m_b42.SetWindowText(".");

        m_b31.SetWindowText("z"); m_b32.SetWindowText("x"); m_b33.SetWindowText("c");
        m_b34.SetWindowText("v"); m_b35.SetWindowText("b"); m_b36.SetWindowText("n");
        m_b37.SetWindowText("m"); m_b38.SetWindowText("`"); m_b39.SetWindowText("~");
        m_b40.SetWindowText("`"); m_b43.SetWindowText("[CAPS]");
    }
}
//*****
****

void CTecladoVirtualDlg::OnTimer(UINT nIDEvent)
{
    //*****
    // Temporizador da varredura *
    //*****
    CRect rect;
    GetWindowRect(&rect);
    m_clist1.SetFocus();
    if(v==16)v=10;//volta ao chegar ao fim
    CRect r;
    int lar=0;
    int alt=0;
    // Leva a varredura horizontal linha por linha
    if(m_dir==1){
        v++;
        CString a;
        a.Format("%d",v);

```

```

// primeira linha
if(v==11){
    CRect r;
    m_b1.GetWindowRect( r);
    SetCursorPos(r.left+r.Width()-4,r.top+r.Height()-4);
    m_focus.SetWindowPos(NULL,0,0,0,0,NULL);
    m_clist1.SetFocus();

m_b1.SetCor( RGB(255,255,0) );m_b2.SetCor( RGB(255,255,0) );m_b3.SetCor( RGB(255,255,0) )
;
    m_b4.SetCor( RGB(255,255,0) );m_b5.SetCor( RGB(255,255,0) );
}
// 2a
if(v==12){
    CRect r;
    m_b11.GetWindowRect( r);
    SetCursorPos(r.left+r.Width()-4,r.top+r.Height()-4);

m_b1.SetCor( GetSysColor( COLOR_BTNFACE ) );m_b2.SetCor( GetSysColor( COLOR_BTNFACE ) );

m_b3.SetCor( GetSysColor( COLOR_BTNFACE ) );m_b4.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b5.SetCor( GetSysColor( COLOR_BTNFACE ) );m_b11.SetCor( RGB(255,255,0) );
    m_b12.SetCor( RGB(255,255,0) );m_b13.SetCor( RGB(255,255,0) );
    m_b14.SetCor( RGB(255,255,0) );m_b15.SetCor( RGB(255,255,0) );
    m_b16.SetCor( RGB(255,255,0) );m_b17.SetCor( RGB(255,255,0) );
    m_b18.SetCor( RGB(255,255,0) );m_b19.SetCor( RGB(255,255,0) );
    m_b20.SetCor( RGB(255,255,0) );m_b41.SetCor( RGB(255,255,0) );
}
// 3a
if(v==13){
    CRect r;
    m_b21.GetWindowRect( r);
    SetCursorPos(r.left+r.Width()-4,r.top+r.Height()-4);

    m_b11.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b12.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b13.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b14.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b15.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b16.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b17.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b18.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b19.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b20.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b41.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b21.SetCor( RGB(255,255,0) );m_b22.SetCor( RGB(255,255,0) );
    m_b23.SetCor( RGB(255,255,0) );m_b24.SetCor( RGB(255,255,0) );
    m_b25.SetCor( RGB(255,255,0) );m_b26.SetCor( RGB(255,255,0) );
    m_b27.SetCor( RGB(255,255,0) );m_b28.SetCor( RGB(255,255,0) );
    m_b29.SetCor( RGB(255,255,0) );m_b30.SetCor( RGB(255,255,0) );
    m_b42.SetCor( RGB(255,255,0) );
}
// 4a
if(v==14){
    CRect r;
    m_b31.GetWindowRect( r);
    SetCursorPos(r.left+r.Width()-4,r.top+r.Height()-4);
    m_b21.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b22.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b23.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b24.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b25.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b26.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b27.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b28.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b29.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b30.SetCor( GetSysColor( COLOR_BTNFACE ) );
    m_b42.SetCor( GetSysColor( COLOR_BTNFACE ) );
}

```

```

m_b31.SetCor( RGB(255,255,0) );m_b32.SetCor( RGB(255,255,0) );m_b33.SetCor( RGB(255,255,
0) );
m_b34.SetCor( RGB(255,255,0) );m_b35.SetCor( RGB(255,255,0) );m_b36.SetCor( RGB(255,255,
0) );
m_b37.SetCor( RGB(255,255,0) );m_b38.SetCor( RGB(255,255,0) );m_b39.SetCor( RGB(255,255,
0) );
m_b40.SetCor( RGB(255,255,0) );m_b43.SetCor( RGB(255,255,0) );m_b8.SetCor( RGB(255,255,0
) );
m_b9.SetCor( RGB(255,255,0) );m_b7.SetCor( RGB(255,255,0) );m_bfechar.SetCor( RGB(255,25
5,0) );
}
//5a
if(v==15){
    CRect r;
    m_clist3.GetWindowRect( r);
    m_focus.SetWindowPos( NULL,r.left-rect.left-
6,0,r.Width()+4,r.Height()+4,NULL);

m_b31.SetCor( GetSysColor( COLOR_BTNFACE) );m_b32.SetCor( GetSysColor( COLOR_BTNFACE) );m
_b33.SetCor( GetSysColor( COLOR_BTNFACE) );

m_b34.SetCor( GetSysColor( COLOR_BTNFACE) );m_b35.SetCor( GetSysColor( COLOR_BTNFACE) );m
_b36.SetCor( GetSysColor( COLOR_BTNFACE) );

m_b37.SetCor( GetSysColor( COLOR_BTNFACE) );m_b38.SetCor( GetSysColor( COLOR_BTNFACE) );m
_b39.SetCor( GetSysColor( COLOR_BTNFACE) );

m_b40.SetCor( GetSysColor( COLOR_BTNFACE) );m_b43.SetCor( GetSysColor( COLOR_BTNFACE) );m
_b8.SetCor( GetSysColor( COLOR_BTNFACE) );

m_b9.SetCor( GetSysColor( COLOR_BTNFACE) );m_b7.SetCor( GetSysColor( COLOR_BTNFACE) );m_b
fechar.SetCor( GetSysColor( COLOR_BTNFACE) );
    SetCursorPos( r.left+r.Width()-4,r.top+r.Height()-4);
}
}
// Varredura Vertical
if(m_dir==2){ //Corre botão por botão
    h++;
    if(v==11){

m_b2.SetCor( GetSysColor( COLOR_BTNFACE) );m_b3.SetCor( GetSysColor( COLOR_BTNFACE) );

m_b4.SetCor( GetSysColor( COLOR_BTNFACE) );m_b5.SetCor( GetSysColor( COLOR_BTNFACE) );
    m_b1.GetWindowRect( r);
    if(h==16)h=11;//volta ao chegar ao fim
    }
    if(v==12){

m_b12.SetCor( GetSysColor( COLOR_BTNFACE) );m_b13.SetCor( GetSysColor( COLOR_BTNFACE) );

m_b14.SetCor( GetSysColor( COLOR_BTNFACE) );m_b15.SetCor( GetSysColor( COLOR_BTNFACE) );

m_b16.SetCor( GetSysColor( COLOR_BTNFACE) );m_b17.SetCor( GetSysColor( COLOR_BTNFACE) );

m_b18.SetCor( GetSysColor( COLOR_BTNFACE) );m_b19.SetCor( GetSysColor( COLOR_BTNFACE) );

m_b20.SetCor( GetSysColor( COLOR_BTNFACE) );m_b41.SetCor( GetSysColor( COLOR_BTNFACE) );
    m_b11.GetWindowRect( r);
    if(h==22)h=11;//volta ao chegar ao fim
    }
    if(v==13){

m_b22.SetCor( GetSysColor( COLOR_BTNFACE) );m_b23.SetCor( GetSysColor( COLOR_BTNFACE) );

m_b24.SetCor( GetSysColor( COLOR_BTNFACE) );m_b25.SetCor( GetSysColor( COLOR_BTNFACE) );

m_b26.SetCor( GetSysColor( COLOR_BTNFACE) );m_b27.SetCor( GetSysColor( COLOR_BTNFACE) );

```

```

m_b28.SetCor(GetSysColor(COLOR_BTNFACE));m_b29.SetCor(GetSysColor(COLOR_BTNFACE));
m_b30.SetCor(GetSysColor(COLOR_BTNFACE));m_b42.SetCor(GetSysColor(COLOR_BTNFACE));
m_b21.GetWindowRect( r);
    if(h==22)h=11;//volta ao chegar ao fim
    }
    if(v==14){

m_b32.SetCor(GetSysColor(COLOR_BTNFACE));m_b33.SetCor(GetSysColor(COLOR_BTNFACE));
m_b34.SetCor(GetSysColor(COLOR_BTNFACE));m_b35.SetCor(GetSysColor(COLOR_BTNFACE));
m_b36.SetCor(GetSysColor(COLOR_BTNFACE));m_b37.SetCor(GetSysColor(COLOR_BTNFACE));
m_b38.SetCor(GetSysColor(COLOR_BTNFACE));m_b39.SetCor(GetSysColor(COLOR_BTNFACE));
m_b40.SetCor(GetSysColor(COLOR_BTNFACE));m_b43.SetCor(GetSysColor(COLOR_BTNFACE));

m_b8.SetCor(GetSysColor(COLOR_BTNFACE));m_b9.SetCor(GetSysColor(COLOR_BTNFACE));

m_b7.SetCor(GetSysColor(COLOR_BTNFACE));m_bfechar.SetCor(GetSysColor(COLOR_BTNFACE)
);
    m_b31.GetWindowRect( r);
    if(h==28)h=11;//volta ao chegar ao fim
    }
    alt=r.top;
// Corre a Lista
    if(v==15){
        CRect r;
        m_clist3.GetWindowRect(r);
        int cont=m_clist3.GetCount();
        m_clist3.SetCurSel(h-11);
        if(h==21 ||h==(cont+11)){
            h=11;
            m_clist3.SetCurSel(0);
        }
    }
//verifica a distancia p/ prox cursor
    if(v!=15){

if(h==11)m_b11.GetWindowRect(r);if(h==12)m_b12.GetWindowRect(r);if(h==13)m_b13.GetW
indowRect(r);

if(h==14)m_b14.GetWindowRect(r);if(h==15)m_b15.GetWindowRect(r);if(h==16)m_b16.GetW
indowRect(r);

if(h==17)m_b17.GetWindowRect(r);if(h==18)m_b18.GetWindowRect(r);if(h==19)m_b19.GetW
indowRect(r);

if(h==20)m_b20.GetWindowRect(r);if(h==21)m_b41.GetWindowRect(r);if(h==22)m_b9.GetWi
ndowRect(r);

if(h==23)m_b8.GetWindowRect(r);if(h==24)m_b7.GetWindowRect(r);if(h==25)m_bfechar.Ge
tWindowRect(r);
        //botao maior
        if(v==11){

if(h==11)m_b1.GetWindowRect(r);if(h==12)m_b2.GetWindowRect(r);if(h==13)m_b3.GetWind
owRect(r);
            if(h==14)m_b4.GetWindowRect(r); if(h==15)m_b5.GetWindowRect(r);
        }
        lar=r.left;
        SetCursorPos(lar+r.Width()-4,alt+r.Height()-4);
    }
}
CDialog::OnTimer(nIDEvent);
}

```

```

//*****
****
bool CTecladoVirtualDlg::SintetizadorAbrir(CString frase)
{
    // Fecha o sintetizador
    SintetizadorFechar();
    TTSMODEINFO TTSModeInfo;
    TTSMODEINFO ttsmiTemp;
    PITTSFIND pITTSFind;
    // inicializa o OLE
    if (FAILED(CoInitialize(NULL))) return false;
    // Cria o objeto Multimedia Audio Destination
    if (FAILED(CoCreateInstance(CLSID_MMAudioDest, NULL, CLSCTX_ALL,
IID_IAudioMultiMediaDevice, (void**) &pIAMM)))
        return false;
    pIAMM->DeviceNumSet(WAVE_MAPPER);
    // Cria o enumerador do Text-To-Speech
    if (FAILED(CoCreateInstance(CLSID_TTSEnumerator, NULL, CLSCTX_ALL, IID_ITTSFind,
(void**) &pITTSFind))){
        pIAMM->Release();
        return false;
    }
    ZeroMemory(&ttsmiTemp, sizeof(ttsmiTemp));
    // Define o idioma (português)
    ttsmiTemp.language.LanguageID = MAKELANGID(LANG_PORTUGUESE,
SUBLANG_PORTUGUESE_BRAZILIAN);
    // Define o tipo de voz Masculina ou Feminina
    if(config.masculino==1)ttsmiTemp.wGender = GENDER_FEMALE;
    else ttsmiTemp.wGender = GENDER_MALE;
    ttsmiTemp.wAge = TTSAGE_ADULT;
    // Encontra o engine baseado nas configurações, acima
    if (FAILED(pITTSFind->Find(&ttsmiTemp, NULL, &TTSModeInfo))){
        pIAMM->Release();
        pITTSFind->Release();
        return false;
    }
    // Seleciona o modo Text-To-Speech
    if (FAILED(pITTSFind->Select(TTSModeInfo.gModeID, &pITTSCentral, (LPUNKNOWN)
pIAMM))){
        pIAMM->Release();
        pITTSFind->Release();
        return false;
    }
    pITTSFind->Release();
    SDATA sdata;
    // Faz a montagem do texto
    int tam = frase.GetLength();
    sdata.dwSize=tam+1;
    sdata.pData = (CHAR *) malloc( sdata.dwSize);
    strcpy((CHAR*)sdata.pData, frase);
    // Envia o texto para o engine
    if (FAILED(pITTSCentral->TextData(CHARSET_TEXT, 0, sdata, NULL,
IID_ITTSBufNotifySinkW))){
        return false;
    }
    // libera
    free(sdata.pData);
    // Retorna true, para não enviar a mensagem de erro
    return true;
}
//*****
****
void CTecladoVirtualDlg::SintetizadorFechar()
{
    //*****
    // Fecha o sintetizador *
    //*****
    if ( pIAMM ) {

```

```

    pIAMM->Release();
    pIAMM = NULL;

}

if ( pITTSCentral ) {
    pITTSCentral->Release();
    pITTSCentral = NULL;
}
}

//*****
****
void CTecladoVirtualDlg::Enviar(char letra)
{
    //*****
    // Envia o caractere para o outro aplicativo *
    //*****
    // Pega o estado da letra
    int estado = HIBYTE(VkKeyScan(letra));
    // Pressiona Shift, control ou menu, se necessário
    if (estado & 0x0001)keybd_event (VK_SHIFT,0,0,0);
    if (estado & 0x0002)keybd_event (VK_CONTROL,0,0,0);
    if (estado & 0x0004)keybd_event (VK_MENU,0,0,0);
    // Envia o evento de pressionar a tecla ao aplicativo
    keybd_event (VkKeyScan(letra),MapVirtualKey (letra, 0),0,0);
    // Solta a tecla
    keybd_event (VkKeyScan(letra),MapVirtualKey (letra, 0),KEYEVENTF_KEYUP,0);
    // Solta a tecla
    if (estado & 0x0004)keybd_event (VK_MENU,0,KEYEVENTF_KEYUP,0);
    if (estado & 0x0002)keybd_event (VK_CONTROL,0,KEYEVENTF_KEYUP,0);
    if (estado & 0x0001)keybd_event (VK_SHIFT,0,KEYEVENTF_KEYUP,0);
}
//*****
****
void CTecladoVirtualDlg::OnBotao()
{
    //*****
    // Cada tecla clicada executa a função *
    //*****
    CString letra,temp;
    m_b11.GetWindowText (letra);
    // Botão Lista de predição
    if(config.historico==1 && config.tamanho==2) {
        m_clist1.SetWindowPos (NULL,2,2,430,83,NULL);
        m_clist3.SetWindowPos (NULL,435,2,148,193,NULL);
        m_cedit1.SetWindowPos (NULL,2,89,430,25,NULL);
    }
    else if(config.historico==1 && config.tamanho==1) {
        m_clist1.SetWindowPos (NULL,2,2,320,61,NULL);
        m_clist3.SetWindowPos (NULL,322,2,113,142,NULL);
        m_cedit1.SetWindowPos (NULL,2,65,320,18,NULL);
    }
    else if(config.historico==0 && config.tamanho==2) {
        m_clist1.SetWindowPos (NULL,2,2,0,0,NULL);
        m_clist3.SetWindowPos (NULL,435,2,148,100,NULL);
        m_cedit1.SetWindowPos (NULL,2,2,430,25,NULL);
    }
    else if(config.historico==0 && config.tamanho==1) {
        m_clist1.SetWindowPos (NULL,2,2,0,0,NULL);
        m_clist3.SetWindowPos (NULL,322,2,113,73,NULL);
        m_cedit1.SetWindowPos (NULL,2,2,320,18,NULL);
    }
}

// Botões de funções
// Tela opções
if(m_b7.m_cima==1) { //associado a tecla Opções
    if(config.varredura==1) {
        m_dir=0;
        m_b18.SetFocus ();
    }
}
}

```

```

//      MessageBox("");
      KillTimer(1);
    }
    COpcaoDlg opcao;
    opcao.DoModal();
    }//abre a janela opcao
if(m_b8.m_cima==1)Outras(); //associado a tecla Num.
if(m_b9.m_cima==1)Outras(); //associado a tecla Espec.
if(m_bfechar.m_cima==1){ //associado a tecla Fechar
    ConfSalvar(); exit(0);
}
// Botões do teclado exibindo letras
if (letra=="a" || letra=="A" || letra=="q" || letra=="Q" ){
    if (m_b11.m_cima==1){m_vedit1+=m_b11.Letra();Completar();}
    if (m_b12.m_cima==1){m_vedit1+=m_b12.Letra();Completar();}
    if (m_b13.m_cima==1){m_vedit1+=m_b13.Letra();Completar();}
    if (m_b14.m_cima==1){m_vedit1+=m_b14.Letra();Completar();}
    if (m_b15.m_cima==1){m_vedit1+=m_b15.Letra();Completar();}
    if (m_b16.m_cima==1){m_vedit1+=m_b16.Letra();Completar();}
    if (m_b17.m_cima==1){m_vedit1+=m_b17.Letra();Completar();}
    if (m_b18.m_cima==1){m_vedit1+=m_b18.Letra();Completar();}
    if (m_b19.m_cima==1){m_vedit1+=m_b19.Letra();Completar();}
    if (m_b20.m_cima==1){m_vedit1+=m_b20.Letra();Completar();}
    if (m_b21.m_cima==1){m_vedit1+=m_b21.Letra();Completar();}
    if (m_b22.m_cima==1){m_vedit1+=m_b22.Letra();Completar();}
    if (m_b23.m_cima==1){m_vedit1+=m_b23.Letra();Completar();}
    if (m_b24.m_cima==1){m_vedit1+=m_b24.Letra();Completar();}
    if (m_b25.m_cima==1){m_vedit1+=m_b25.Letra();Completar();}
    if (m_b26.m_cima==1){m_vedit1+=m_b26.Letra();Completar();}
    if (m_b27.m_cima==1){m_vedit1+=m_b27.Letra();Completar();}
    if (m_b28.m_cima==1){m_vedit1+=m_b28.Letra();Completar();}
    if (m_b29.m_cima==1){m_vedit1+=m_b29.Letra();Completar();}
    if (m_b31.m_cima==1){m_vedit1+=m_b31.Letra();Completar();}
    if (m_b32.m_cima==1){m_vedit1+=m_b32.Letra();Completar();}
    if (m_b33.m_cima==1){m_vedit1+=m_b33.Letra();Completar();}
    if (m_b34.m_cima==1){m_vedit1+=m_b34.Letra();Completar();}
    if (m_b35.m_cima==1){m_vedit1+=m_b35.Letra();Completar();}
    if (m_b36.m_cima==1){m_vedit1+=m_b36.Letra();Completar();}
    if (m_b37.m_cima==1){m_vedit1+=m_b37.Letra();Completar();}
// Sugere palavras na lista que completam o que está sendo digitado ou predizem a
inicial se houve ponto ou virgula
    if (m_b41.m_cima==1 ||m_b42.m_cima==1){ // se ponto final ou vírgula, use
Pi x B
        m_vedit1.TrimRight(); // se apertou antes OnFrente, limpa espaço
        if (m_b41.m_cima==1)m_vedit1+=m_b41.Letra(); // escreve vírgula
        if (m_b42.m_cima==1)m_vedit1+=m_b42.Letra(); // escreve ponto
        m_vedit1+=" ";
        Predicao();
    }
    //else if (m_vedit1!="")Completar();

    if (m_b30.m_cima==1 && config.acento=="^")m_vedit1+="^";
    else if (m_b30.m_cima==1 && config.acento!="^")config.acento="^";
    if (m_b38.m_cima==1 && config.acento=="´")m_vedit1+="´";
    else if (m_b38.m_cima==1 && config.acento!="´")config.acento="´";
    if (m_b39.m_cima==1 && config.acento=="~")m_vedit1+="~";
    else if (m_b39.m_cima==1 && config.acento!="~")config.acento="~";
    if (m_b40.m_cima==1 && config.acento=="`")m_vedit1+="`";
    else if (m_b40.m_cima==1 && config.acento!="`")config.acento="`";

// Letras maiúsculas ou minúsculas
    if (m_b43.m_cima==1){
        if(config.maiuscula==1)config.maiuscula=0;
        else config.maiuscula=1;
        CString letra;m_b11.GetWindowText(letra);
        letra.MakeLower();
        if(letra=="a")FormatoABC();

```

```

        else FormatoQWE();
    }
}
/** Conta Clique
    m_veditContaClique++;
    UpdateData (false);
}
// Botões do teclado exibindo caracteres especiais
if (letra=="[ESC]") {
    int cont=0;
    char aplic[256];
    HWND hWnd = ::FindWindow (NULL, NULL);
    while (hWnd && cont<=1) {
        ::GetWindowText (hWnd, aplic, 255);
        if (aplic [0] != 0 && ::IsWindowVisible (hWnd)) {
            ::SetForegroundWindow(hWnd);
            cont++;
        }
        hWnd = ::GetNextWindow (hWnd, GW_HWNDNEXT);
    }
}
// Envia os comandos ao aplicativo externo
if (m_b11.m_cima==1) keybd_event (VK_ESCAPE,0,0,0);
if (m_b12.m_cima==1) keybd_event (VK_UP,0,0,0);
if (m_b13.m_cima==1) keybd_event (VK_TAB,0,0,0);
if (m_b14.m_cima==1) keybd_event (VK_F1,0,0,0);
if (m_b15.m_cima==1) keybd_event (VK_F2,0,0,0);
if (m_b16.m_cima==1) keybd_event (VK_F3,0,0,0);
if (m_b17.m_cima==1) keybd_event (VK_F4,0,0,0);
if (m_b18.m_cima==1) keybd_event (VK_F5,0,0,0);
if (m_b19.m_cima==1) keybd_event (VK_F6,0,0,0);
if (m_b20.m_cima==1) keybd_event (VK_F7,0,0,0);
if (m_b21.m_cima==1) keybd_event (VK_F8,0,0,0);
if (m_b21.m_cima==1) keybd_event (VK_LEFT,0,0,0);
if (m_b22.m_cima==1) m_vedit1+=m_b22.Letra();
if (m_b23.m_cima==1) keybd_event (VK_RIGHT,0,0,0);
if (m_b26.m_cima==1) m_vedit1+="&";
if (m_b27.m_cima==1) m_vedit1+=m_b27.Letra();
if (m_b28.m_cima==1) m_vedit1+=m_b28.Letra();
if (m_b29.m_cima==1) keybd_event (VK_INSERT,0,0,0);
if (m_b30.m_cima==1) keybd_event (VK_HOME,0,0,0);
if (m_b42.m_cima==1) keybd_event (VK_PRIOR,0,0,0);
if (m_b31.m_cima==1) m_vedit1+=m_b31.Letra();
if (m_b32.m_cima==1) keybd_event (VK_DOWN,0,0,0);
if (m_b33.m_cima==1) m_vedit1+=m_b33.Letra();
if (m_b36.m_cima==1) m_vedit1+=m_b36.Letra();
if (m_b37.m_cima==1) m_vedit1+=m_b37.Letra();
if (m_b38.m_cima==1) m_vedit1+=m_b38.Letra();
if (m_b39.m_cima==1) keybd_event (VK_DELETE,0,0,0);
if (m_b40.m_cima==1) keybd_event (VK_END,0,0,0);
if (m_b43.m_cima==1) keybd_event (VK_NEXT,0,0,0);
// Sugere palavras na lista que completam o que está sendo digitado ou predizem
a inicial
if (m_b24.m_cima==1||m_b25.m_cima==1||m_b34.m_cima==1||m_b35.m_cima==1) {
// se houve pontuação, use Pi x B
    m_vedit1.TrimRight(); // se apertou anteriormente OnFrente,
remove o espaço
    if (m_b24.m_cima==1) m_vedit1+=m_b24.Letra(); // escreve ":"
    if (m_b25.m_cima==1) m_vedit1+=m_b25.Letra(); // escreve "!"
    if (m_b34.m_cima==1) m_vedit1+=m_b34.Letra(); // escreve ";"
    if (m_b35.m_cima==1) m_vedit1+=m_b35.Letra(); // escreve "?"
    m_vedit1+=" ";
    Predicao();
}
else if (m_vedit1!="") Completar();
}
// Botões do teclado exibindo a calculadora
if (letra=="Calc.") {
    if (m_b11.m_cima==1) {::WinExec("calc", SW_SHOW);config.externo=true;}
    if (m_b12.m_cima==1) m_vedit1+=m_b12.Letra();
}

```

```

    if (m_b13.m_cima==1)m_veditl+=m_b13.Letra();
    if (m_b14.m_cima==1)m_veditl+=m_b14.Letra();
    if (m_b15.m_cima==1)m_veditl+=m_b15.Letra();
    if (m_b16.m_cima==1)m_veditl+=m_b16.Letra();
    if (m_b17.m_cima==1)m_veditl+=m_b17.Letra();
    if (m_b18.m_cima==1)m_veditl+=m_b18.Letra();
    if (m_b19.m_cima==1)m_veditl+=m_b19.Letra();
    if (m_b20.m_cima==1)m_veditl+=m_b20.Letra();
    if (m_b41.m_cima==1)m_veditl+=m_b41.Letra();
    if (m_b21.m_cima==1){::WinExec("notepad", SW_SHOW);config.externo=true;}
    if (m_b22.m_cima==1)m_veditl+=m_b22.Letra();
    if (m_b23.m_cima==1)m_veditl+=m_b23.Letra();
    if (m_b24.m_cima==1)m_veditl+=m_b24.Letra();
    if (m_b25.m_cima==1)m_veditl+=m_b25.Letra();
    if (m_b26.m_cima==1)m_veditl+=m_b26.Letra();
    if (m_b27.m_cima==1)m_veditl+=m_b27.Letra();
    if (m_b28.m_cima==1)m_veditl+=m_b28.Letra();
    if (m_b29.m_cima==1)m_veditl+=m_b29.Letra();
    if (m_b30.m_cima==1)m_veditl+=m_b30.Letra();
    if (m_b42.m_cima==1)m_veditl+=m_b42.Letra();
    if (m_b31.m_cima==1){::system("start iexplore");config.externo=true;}
    if (m_b32.m_cima==1){::system("start winword");config.externo=true;}
    if (m_b33.m_cima==1){::system("start excel");config.externo=true;}
    if (m_b34.m_cima==1){WinExec("hh.exe ""./AjudaTeclado.chm" ,SW_SHOW);}
    if (m_b35.m_cima==1)OnImprimir();
    if (m_b36.m_cima==1)m_veditl+=m_b36.Letra();
    if (m_b37.m_cima==1)m_veditl+=m_b37.Letra();
    if (m_b38.m_cima==1)m_veditl+=m_b38.Letra();
    if (m_b39.m_cima==1)m_veditl+=m_b39.Letra();
    if (m_b40.m_cima==1)m_veditl+=m_b40.Letra();
    if (m_b43.m_cima==1)m_veditl+=m_b43.Letra();
}

// Botões superiores
if(m_b1.m_cima==1)OnLimpar();
if(m_b2.m_cima==1)OnVoltar();
if(m_b3.m_cima==1)OnTras();
if(m_b4.m_cima==1)OnFrente();
if(m_b5.m_cima==1)OnEnter();

//Lista de Predição
if(m_clist3.m_cima==1)OnLista();

// Leva cursor à ultima letra
m_ceditl.SetFocus();
m_ceditl.SetSel(m_veditl.GetLength(),m_veditl.GetLength(),true);

}
//*****
****
void CTecladoVirtualDlg::OnLimpar()
{
    //*****
    // Botão limpar *
    //*****
    // Limpa a área de texto
    m_veditl.Empty();
    indPred=0;
    Predicao();
}
//*****
****
void CTecladoVirtualDlg::OnVoltar()
{
    //*****
    // Botão voltar (retorna uma palavra) *
    //*****

```

```

// Apaga os caracteres da frase até achar um espaço, ou seja, apaga última palavra
UpdateData (false);
m_vedit1.TrimRight();
UpdateData (false);
int x=m_vedit1.ReverseFind(' ')+1;
if(x>0){
    m_vedit1.Delete(x,m_vedit1.GetLength()-x);
    indPred--;
    Predicao();
}
else OnLimpar();
}
//*****
****
void CTecladoVirtualDlg::OnTras()
{
    //*****
    // Volta um caractere *
    //*****
    int cont=0;
    if(!m_vedit1.IsEmpty()){
        if(m_vedit1.Right(1)==" ")indPred--;
        m_vedit1.Delete(m_vedit1.GetLength()-1,1); // Retira o último caractere
        UpdateData (false);
        if (m_vedit1.IsEmpty()){
            Predicao(); //se ficar vazio, Iniciar novamente
        }
        else Completar(); // Pesquisa as palavras na lista
    }
    else { // Se estiver vazia a frase e habilitado o uso externo,
        if(config.externo==1){ //acha o ultimo processo e faz o backspace
            HWND hWnd = ::FindWindow (NULL, NULL);
            char aplic[256];
            while (hWnd && cont<=1){
                ::GetWindowText (hWnd, aplic, 255);
                if (aplic [0] != 0 && ::IsWindowVisible (hWnd)){
                    ::SetForegroundWindow(hWnd);
                    cont++;
                }
                hWnd = ::GetNextWindow (hWnd, GW_HWNDNEXT);
            }
            keybd_event (VK_BACK,0,0,0);
        }
    }
}
//*****
****
void CTecladoVirtualDlg::OnFrente()
{
    //*****
    // Adiciona apenas um espaço *
    //*****
    char aplic[256];
    if(m_vedit1.GetLength()<0){ // Se estiver vazia a frase e habilitado o uso
externo, então
        // acha o ultimo processo e avança no aplicativo
        if(config.externo==1){
            HWND hWnd = ::FindWindow (NULL, NULL);
            int cont=0;
            while (hWnd && cont<=1){
                ::GetWindowText (hWnd, aplic, 255);
                if (aplic [0] != 0 && ::IsWindowVisible (hWnd)){
                    ::SetForegroundWindow(hWnd);
                    cont++;
                }
                hWnd = ::GetNextWindow (hWnd, GW_HWNDNEXT);
            }
            keybd_event (VK_SPACE,0,0,0);
        }
    }
}

```

```

    }
}
else if(m_vedit1.GetLength(>0){
    UpdateData(false);
    bool incrementa=true, inicio=ChecaInicio();
    CString palavra=m_vedit1;
    if (m_vedit1.Right(1)==" ") incrementa=false;
    palavra.Delete(0,palavra.ReverseFind(' ')+1);
    // Adiciona o espaço
    m_vedit1.TrimRight();
    m_vedit1+=" ";
    UpdateData(false);
    // Quando o usuário digita uma palavra, pesquisa na matriz de busca
    int i,j,k,m;
    double Valor=0;
    bool achou=false, emPi=false;
    for(i=0; i<=fimDaLista; i++) mtzPred[indPred].lstPred[i].Resultado=0; //Limpa
lista atual
    if(inicio){ //Use Pi x B se estiver em Pi:
        for(i=0; i<numPalavrasDiferentes; i++){ //Para cada palavra da matriz de
busca...
            if(palavra.CompareNoCase(MatrizDeBusca[i].Palavra)==0){ //ache o que
corresponder à palavra digitada...
                for(j=0; j<numClassesDaPalavra; j++){ //para cada classe da palavra
encontrada...
                    mtzPred[indPred].classe[j]=MatrizDeBusca[i].parB[j].indClasse;
                    mtzPred[indPred].delta[j]=VetorPi[mtzPred[indPred].classe[j]]*
MatrizB1[mtzPred[indPred].classe[j]].Probabilidade[MatrizDeBusca[i].parB[j].PosiB];
                    if(mtzPred[indPred].delta[j]>0) emPi=true;
                }
            }
        }
    }
    if(!emPi){
        for(i=0; i<numClassesDaPalavra; i++) mtzPred[indPred].delta[i]=0; //limpa
deltas atuais
        for(i=0; i<NumeroDeClasses; i++){ //para cada Pi...
            m=-1;
            for(j=0; j<numClassesDaPalavra; j++){ //para cada delta atual
                if(VetorPi[i]>mtzPred[indPred].delta[j]) m=j;
                else break; //sai do loop
            }
            if(m>=0){ //se Pi maior que algum delta atual, encaixa Pi na listagem
delta atual
                for(j=0; j<m; j++){
                    mtzPred[indPred].delta[j]=mtzPred[indPred].delta[j+1];
                    mtzPred[indPred].classe[j]=mtzPred[indPred].classe[j+1];
                }
                mtzPred[indPred].delta[m]=VetorPi[i];
                mtzPred[indPred].classe[m]=i;
            }
        }
    }
}
else{ //Viterbi
    for(i=0; i<numPalavrasDiferentes; i++){ //para cada palavra da matriz de
busca...
        if(palavra.CompareNoCase(MatrizDeBusca[i].Palavra)==0){ //ache o que
corresponder à palavra digitada...
            achou=true;
            for(j=0; j<numClassesDaPalavra; j++){ //Para cada classe da palavra
encontrada...
                mtzPred[indPred].delta[j]=0;
                mtzPred[indPred].classe[j]=MatrizDeBusca[i].parB[j].indClasse;
                for(k=0; k<numClassesDaPalavra; k++){ //use cada delta anterior...
                    if(mtzPred[indPred-1].delta[k]*MatrizA[mtzPred[indPred-1].classe[k]]
[mtzPred[indPred].classe[j]]>mtzPred[indPred].delta[j]){
                        mtzPred[indPred].delta[j]=mtzPred[indPred-1].delta[k]*

```

```

        MatrizA[mtzPred[indPred-
1].classe[k]][mtzPred[indPred].classe[j]];
    }
}
    mtzPred[indPred].delta[j]=mtzPred[indPred].delta[j]*

MatrizB1[mtzPred[indPred].classe[j]].Probabilidade[MatrizDeBusca[i].parB[j].PosiB];
    }
}
    }
    if(!achou){
        for(i=0;i<numClassesDaPalavra;i++)mtzPred[indPred].delta[i]=0; //limpa
deltas atuais
        for(i=0;i<NumeroDeClasses;i++){ //para cada classe de destino na matriz
A...
            for(j=0;j<numClassesDaPalavra;j++){ //para classe de origem no delta
anterior...
                Valor=mtzPred[indPred-1].delta[j]*MatrizA[mtzPred[indPred-
1].classe[j]][i];
                //Classificador:
                m=-1;
                for(k=0;k<numClassesDaPalavra;k++){ //Se Valor > que algum delta atual,
encaixe a classe...
                    if(Valor>mtzPred[indPred].delta[k])m=k;
                    else break; //sai do loop
                }
                if(m>=0){ //se delta(n-1) * A (todas) maior que algum delta atual,
encaixa valor...
                    for(k=0;k<m;k++){
                        mtzPred[indPred].delta[k]=mtzPred[indPred].delta[k+1];
                        mtzPred[indPred].classe[k]=mtzPred[indPred].classe[k+1];
                    }
                    mtzPred[indPred].delta[m]=Valor;
                    mtzPred[indPred].classe[m]=i;
                }
            }
        }
    }
}
    if(incrementa==true){
        indPred++;
        Predicao();
    }
}
}
//*****
****
void CTecladoVirtualDlg::OnEnter()
{
    //*****
    // Botão enter clicado *
    //*****
    // Aciona o sintetizador
    UpdateData(false);
    if(config.som>0){
        if(SintetizadorAbrir(m_vedit1)==false){
            MessageBox("Não foi possível iniciar o sintetizador.", "ERRO", MB_ICONERROR);
        }
    }
    int tam=0;
    tam = m_vedit1.GetLength();
    int cont=0;
    char aplic[256];
    // Acha o ultimo processo
    if(config.externo==1){
        HWND hWnd = ::FindWindow (NULL, NULL);
        while (hWnd && cont<=1){
            ::GetWindowText (hWnd, aplic, 255);

```

```

    if (aplic [0] != 0 && ::IsWindowVisible (hWnd)){
        ::SetForegroundWindow(hWnd);
        cont++;
    }
    hWnd = ::GetNextWindow (hWnd, GW_HWNDNEXT);
}
}
// Envia ao outro aplicativo
if(config.externo==1 && tam>0){
    char *texto=new char[tam];
    strcpy(texto,m_vedit1);
    for(int i=0;i<tam;i++){
        char letra=texto[i];
        if (letra=='á') {Enviar('´');Enviar('a');}
        else if (letra=='à') {Enviar('`');Enviar('a');}
        else if (letra=='ã') {Enviar('~');Enviar('a');}
        else if (letra=='â') {Enviar('^');Enviar('a');}
        else if (letra=='Á') {Enviar('´');Enviar('A');}
        else if (letra=='À') {Enviar('`');Enviar('A');}
        else if (letra=='Ã') {Enviar('~');Enviar('A');}
        else if (letra=='Â') {Enviar('^');Enviar('A');}

        else if (letra=='é') {Enviar('´');Enviar('e');}
        else if (letra=='è') {Enviar('`');Enviar('e');}
        else if (letra=='ê') {Enviar('^');Enviar('e');}
        else if (letra=='ë') {Enviar('`');Enviar('E');}
        else if (letra=='È') {Enviar('`');Enviar('E');}
        else if (letra=='Ê') {Enviar('^');Enviar('E');}

        else if (letra=='í') {Enviar('´');Enviar('i');}
        else if (letra=='ì') {Enviar('`');Enviar('i');}
        else if (letra=='î') {Enviar('^');Enviar('i');}
        else if (letra=='Î') {Enviar('´');Enviar('I');}
        else if (letra=='Ì') {Enviar('`');Enviar('I');}
        else if (letra=='Ê') {Enviar('^');Enviar('I');}

        else if (letra=='ó') {Enviar('´');Enviar('o');}
        else if (letra=='ò') {Enviar('`');Enviar('o');}
        else if (letra=='ô') {Enviar('^');Enviar('o');}
        else if (letra=='õ') {Enviar('~');Enviar('o');}
        else if (letra=='Ó') {Enviar('´');Enviar('O');}
        else if (letra=='Ò') {Enviar('`');Enviar('O');}
        else if (letra=='Ô') {Enviar('^');Enviar('O');}
        else if (letra=='Õ') {Enviar('~');Enviar('O');}

        else if (letra=='ú') {Enviar('´');Enviar('u');}
        else if (letra=='ù') {Enviar('`');Enviar('u');}
        else if (letra=='û') {Enviar('^');Enviar('u');}
        else if (letra=='Û') {Enviar('´');Enviar('U');}
        else if (letra=='Ù') {Enviar('`');Enviar('U');}
        else if (letra=='Û') {Enviar('^');Enviar('U');}
        else Enviar(letra);
    }
    ::Sleep(500);
    HWND hWnd = ::FindWindow (NULL,NULL);
    hWnd = ::FindWindow (NULL,NULL);
    ::SetForegroundWindow(hWnd);
}
// Se a frase estiver vazia e o botão enter pressionado
if(config.externo==1 && tam==0){
    keybd_event (VK_RETURN,0,0,0);
}
// Se tiver digitado algo ao p. enter
if(tam>0){
// Adiciona o tempo e a frase no historico
char time[10];
_strtime(time);
CString texto=m_vedit1,texto2;

```

```

    CString temp=" ";
    temp+=time;
    temp+=") ";
// Pula linha
    while(texto.GetLength()>44) {
        texto2=texto;
        texto.Delete(44,texto.GetLength()-44);
        texto2.Delete(0,44);
        temp+=texto;
        m_clist1.AddString(temp);
        texto=texto2;
        temp="          ";
    }
    temp+=texto;
    m_clist1.AddString(temp);
// Rola a lista de histórico quando as 4 linhas já estiverem cheias
    if(m_clist1.GetCount()>4)m_clist1.DeleteString(0);
    m_clist1.SetCurSel(m_clist1.GetCount()-1);
    m_vedit1.Empty(); //a atualização dos controles do diálogo será em
CarregaLista(0)
// Faz a predicao
    indPred=0;
    Predicao();
}
} //*****
****
void CTecladoVirtualDlg::Outras()
{
    //*****
    //  exhibe os botões de funções  *
    //*****
    CString texto,texto1,texto2;
    m_b9.GetWindowText(texto1);
    m_b8.GetWindowText(texto2);
    if(m_b9.m_cima==1){m_b9.GetWindowText(texto);}
    if(m_b8.m_cima==1){m_b8.GetWindowText(texto);}
// Exibe no teclado os caracteres especiais
    if(texto=="Espec.") {
        m_b11.SetWindowText("[ESC]");           m_b12.SetWindowText("^");
m_b13.SetWindowText("[TAB]");
        m_b14.SetWindowText("[F1]");           m_b15.SetWindowText("[F2]");
m_b16.SetWindowText("[F3]");
        m_b17.SetWindowText("[F4]");           m_b18.SetWindowText("[F5]");
m_b19.SetWindowText("[F6]");
        m_b20.SetWindowText("[F7]"); m_b41.SetWindowText("[F8]");

        m_b21.SetWindowText("<-"); m_b22.SetWindowText("@"); m_b23.SetWindowText(">");
        m_b24.SetWindowText(";"); m_b25.SetWindowText("!"); m_b26.SetWindowText("&&");
        m_b27.SetWindowText("*"); m_b28.SetWindowText("|");
m_b29.SetWindowText("[Ins]");
        m_b30.SetWindowText("[Home]"); m_b42.SetWindowText("[P.Up]");

        m_b31.SetWindowText("'"); m_b32.SetWindowText("v");
m_b33.SetWindowText("\\");
        m_b34.SetWindowText(";"); m_b35.SetWindowText("?"); m_b36.SetWindowText("#");
        m_b37.SetWindowText("$"); m_b38.SetWindowText("%");
m_b39.SetWindowText("[Del]");
        m_b40.SetWindowText("[End]"); m_b43.SetWindowText("[Down]");
    }
// Exibe no teclado os números
    if(texto=="Num.") {
        config.numero=true;

        m_b11.SetWindowText("Calc."); m_b12.SetWindowText("1");
m_b13.SetWindowText("2");
        m_b14.SetWindowText("3"); m_b15.SetWindowText("4"); m_b16.SetWindowText("5");
        m_b17.SetWindowText("6"); m_b18.SetWindowText("7"); m_b19.SetWindowText("8");

```

```

    m_b20.SetWindowText("9"); m_b41.SetWindowText("0");

    m_b21.SetWindowText("Notas"); m_b22.SetWindowText("+"); m_b23.SetWindowText("-");
    m_b24.SetWindowText("*"); m_b25.SetWindowText("/"); m_b26.SetWindowText("%");
    m_b27.SetWindowText("="); m_b28.SetWindowText(""); m_b29.SetWindowText("");
    m_b30.SetWindowText(""); m_b42.SetWindowText("");

    m_b31.SetWindowText("Internet"); m_b32.SetWindowText("Word");
    m_b33.SetWindowText("Excel");
    m_b34.SetWindowText("Ajuda"); m_b35.SetWindowText("Imprimir");
    m_b36.SetWindowText("(");
    m_b37.SetWindowText(")"); m_b38.SetWindowText("["); m_b39.SetWindowText("]");
    m_b40.SetWindowText("<"); m_b43.SetWindowText(">");
}
// Exibe no teclado as letras
if(texto=="Letras"){
    if(config.formato=="abcde") FormatoABC();
    else if(config.formato=="qwert") FormatoQWE();
}
// Desabilita os botões sem uso externo
if(config.externo==false && texto=="Espec."){
    m_b11.EnableWindow(false); m_b12.EnableWindow(false); m_b13.EnableWindow(false);
    m_b14.EnableWindow(false); m_b15.EnableWindow(false); m_b16.EnableWindow(false);
    m_b17.EnableWindow(false); m_b18.EnableWindow(false); m_b19.EnableWindow(false);
    m_b20.EnableWindow(false); m_b41.EnableWindow(false);
    m_b21.EnableWindow(false); m_b23.EnableWindow(false);
    m_b32.EnableWindow(false);

    m_b29.EnableWindow(false); m_b30.EnableWindow(false);
    m_b39.EnableWindow(false); m_b40.EnableWindow(false);
    m_b42.EnableWindow(false); m_b43.EnableWindow(false);
}
else if(texto!="Espec."){
    m_b11.EnableWindow(true); m_b12.EnableWindow(true); m_b13.EnableWindow(true);
    m_b14.EnableWindow(true); m_b15.EnableWindow(true); m_b16.EnableWindow(true);
    m_b17.EnableWindow(true); m_b18.EnableWindow(true); m_b19.EnableWindow(true);
    m_b20.EnableWindow(true); m_b41.EnableWindow(true);
    m_b21.EnableWindow(true); m_b23.EnableWindow(true);
    m_b32.EnableWindow(true);

    m_b29.EnableWindow(true); m_b30.EnableWindow(true);
    m_b39.EnableWindow(true); m_b40.EnableWindow(true);
    m_b42.EnableWindow(true); m_b43.EnableWindow(true);
}

if(m_b9.m_cima==1 && texto2=="Num.") {m_b9.SetWindowText("Letras");}
if(m_b8.m_cima==1 && texto1=="Espec.") && texto2=="Num.") {m_b8.SetWindowText("Letras");}
if(m_b9.m_cima==1 && texto2=="Espec.") && texto1=="Num.") {m_b9.SetWindowText("Letras");}
if(m_b8.m_cima==1 && texto2=="Espec.") && texto1=="Num.") {m_b8.SetWindowText("Letras");}

if(m_b9.m_cima==1 && texto2=="Num.") {m_b9.SetWindowText("Espec.");}
if(m_b8.m_cima==1 && texto1=="Letras") && texto2=="Num.") {m_b8.SetWindowText("Espec.");}
if(m_b9.m_cima==1 && texto1=="Letras") && texto2=="Num.") {m_b9.SetWindowText("Espec.");}
if(m_b8.m_cima==1 && texto2=="Letras") && texto1=="Num.") {m_b8.SetWindowText("Espec.");}

if(m_b9.m_cima==1 && texto2=="Letras") {m_b9.SetWindowText("Num.");}
if(m_b8.m_cima==1 && texto1=="Espec.") && texto2=="Letras") {m_b8.SetWindowText("Num.");}

```

```

    if(m_b9.m_cima==1          &&          texto2=="Espec."          &&
    texto1=="Letras") m_b9.SetWindowText("Num.");}
    if(m_b8.m_cima==1          &&          texto2=="Espec."          &&
    texto1=="Letras") m_b8.SetWindowText("Num.");}
}
//*****
****
void CTecladoVirtualDlg::OnLista()
{
    //*****
    // Após clicar sobre a lista *
    //*****
    bool inicio=ChecaInicio();
    int i,j,itemDaLista=m_clist3.GetCurSel();
    int linBusca=mtzPred[indPred].lstPred[fimDaLista-itemDaLista].LinMtzBusca;
    CString txtSeleto=MatrizDeBusca[linBusca].Palavra;
    //dá no mesmo que usar... m_clist3.GetText(m_clist3.GetCurSel(),txtSeleto);
    //else m_clist3.GetText(indLista,txtSeleto);
    m_vedit1=m_vedit1.Left(m_vedit1.ReverseFind(' ')+1)+txtSeleto+" ";
    UpdateData(false);
    // Faz a predição
    if(inicio){
        for(i=0;i<numClassesDaPalavra;i++){ //para cada classe da palavra
    escolhida...
            mtzPred[indPred].classe[i]=MatrizDeBusca[linBusca].parB[i].indClasse;
            mtzPred[indPred].delta[i]=VetorPi[mtzPred[indPred].classe[i]]*

MatrizB1[mtzPred[indPred].classe[i]].Probabilidade[MatrizDeBusca[linBusca].parB[i].
PosiB];
        }
    }
    else{ //Viterbi
        for(i=0;i<numClassesDaPalavra;i++){ //para cada classe da palavra escolhida...
            mtzPred[indPred].delta[i]=0;//inicialmente limpe delta atual e
            mtzPred[indPred].classe[i]=MatrizDeBusca[linBusca].parB[i].indClasse;
            //memorize sua classe e use-a com...
            for(j=0;j<numClassesDaPalavra;j++){ //cada classe do delta anterior...
                if(mtzPred[indPred-1].delta[j]*MatrizA[mtzPred[indPred-1].classe[j]]
                [mtzPred[indPred].classe[i]]>=mtzPred[indPred].delta[i]){
                    mtzPred[indPred].delta[i]=mtzPred[indPred-1].delta[j]*
                    MatrizA[mtzPred[indPred-1].classe[j]][mtzPred[indPred].classe[i]];
                }
            }
            mtzPred[indPred].delta[i]=mtzPred[indPred].delta[i]*

MatrizB1[mtzPred[indPred].classe[i]].Probabilidade[MatrizDeBusca[linBusca].parB[i].
PosiB];
        }
    }
    indPred++;
    Predicao();
}
//*****
****
bool CTecladoVirtualDlg::ChecaInicio()
{
    //*****
    //Verifica se é inicio de frase *
    //*****
    UpdateData(false);
    // Deteta se antes da última palavra, na linha temp, houve pontuação
    bool inicio=false;
    CString frase=m_vedit1;
    int posiEspaco=frase.ReverseFind(' ');
    if(posiEspaco>0){
        frase.Delete(0,posiEspaco-1);
        if(frase.ReverseFind(' ')-1==frase.FindOneOf(".,:;?!")){
            inicio=true;
        }
    }
}

```

```

    }
}
else inicio=true;
return inicio;
}
//*****
****
void CTecladoVirtualDlg::Majores(double valor, int indBusca)
{
    int i,j=-1,k=0;
    for(i=0;i<=fimDaLista;i++){ //para valor na lista atual(ordenada, decrescente)...
        if(valor>mtzPred[indPred].lstPred[i].Resultado)j=i;
        if(indBusca==mtzPred[indPred].lstPred[i].LinMtzBusca)k=i;
    }
    if(j>=0){ //se delta(n-1) * A (todas) maior que algum delta atual, encaia
valor...
        for(i=k;i<j;i++){

mtzPred[indPred].lstPred[i].Resultado=mtzPred[indPred].lstPred[i+1].Resultado;

mtzPred[indPred].lstPred[i].LinMtzBusca=mtzPred[indPred].lstPred[i+1].LinMtzBusca;
        }
        if(j>=k){
            mtzPred[indPred].lstPred[j].Resultado=valor;
            mtzPred[indPred].lstPred[j].LinMtzBusca=indBusca;
        }
    }
}
//*****
****
void CTecladoVirtualDlg::CarregaLista(int indice)
{
    UpdateData(false);
    m_clist3.ResetContent(); //Apaga o que houver na lista de predição
    for(int i=0;i<=fimDaLista;i++){ //na lista de predição atualmente cabem apenas 10
palavras
        if(mtzPred[indice].lstPred[fimDaLista-i].Resultado>0){
            m_clist3.AddString(MatrizDeBusca[mtzPred[indice].lstPred[fimDaLista-
i].LinMtzBusca].Palavra);
        }
    }
    // Leva à ultima letra na linha de composição
    m_cedit1.SetFocus();
    m_cedit1.SetSel(m_vedit1.GetLength(),m_vedit1.GetLength(),true);
    UpdateData(false);
}
//*****
****
void CTecladoVirtualDlg::Predicao()
{
    //*****
    //Analisa palavras em temp e prediz a próxima *
    //*****
    //O cálculo recursivo é feito de "trás para frente" para diminuir esforço
computacional
    UpdateData(false);
    int i,j,k;
    double Valor=0;
    //char temp[50];
    bool inicio=ChecaInicio(),achou=false;
    for(i=0;i<=fimDaLista;i++)mtzPred[indPred].lstPred[i].Resultado=0; //Limpa lista
atual
    if(inicio){ //se inicio, carregue mtzPred[indPred] com Pi x B
        for(i=0;i<numPalavrasDiferentes;i++){//para cada simbolo observado (cada
palavra)...
            for(j=0;j<numClassesDaPalavra;j++){ //para cada classe dessa palavra:
                Valor=VetorPi[MatrizDeBusca[i].parB[j].indClasse]*

```

```

MatrizB1[MatrizDeBusca[i].parB[j].indClasse].Probabilidade[MatrizDeBusca[i].parB[j]
.PosiB];
    Maiores(Valor,i);
    }
}
}
else{//associada ao delta da lista anterior...
    for(i=0;i<numPalavrasDiferentes;i++){//para cada cada palavra possível de ser
observada...
        for(j=0;j<numClassesDaPalavra;j++){ //teste se para cada delta anterior...
            if(mtzPred[indPred-1].delta[j]>0){ //que for diferentes de 0...
                for(k=0;k<numClassesDaPalavra;k++){//vezes A, para classe da palavra
atual...
                    Valor=mtzPred[indPred-1].delta[j]*
                        MatrizA[mtzPred[indPred-
1].classe[j]][MatrizDeBusca[i].parB[k].indClasse]*
MatrizB1[MatrizDeBusca[i].parB[k].indClasse].Probabilidade[MatrizDeBusca[i].parB[k]
.PosiB];
                    Maiores(Valor,i);
                    //if(Valor>0)achou=true;
                }
            }
        }
    }
}
//if(achou==false){
//    MessageBox("não achou");
//    for(i=0;i<7;i++)MessageBox(ltoa(mtzPred[indPred-1].delta[i],temp,10));
//}
CarregaLista(indPred);
}
//*****
****
void CTecladoVirtualDlg::Completar()
{
//*****
*****
// Quando o usuário digita uma letra, pesquisa na matriz de busca e restringe
escopo *
//*****
*****
    UpdateData(false);
    int i,j,k;
    double Valor=0;
    bool inicio=ChecaInicio();
    CString PalavraDaMatriz;
    CString pedaco=m_vedit1;
    for(i=0;i<=fimDaLista;i++)mtzPred[indPred].lstPred[i].Resultado=0; //Limpa lista
atual
    pedaco.Delete(0,pedaco.ReverseFind(' ')+1);
    if(inicio){ // se for inicio, carregue mtzPred[indPred] com Pi x B
        for(i=0;i<numPalavrasDiferentes;i++){//para cada palavra sendo digitada...
            PalavraDaMatriz=MatrizDeBusca[i].Palavra;
            if(pedaco.CompareNoCase(PalavraDaMatriz.Left(pedaco.GetLength()))==0){//se
houve alguma palavra com o mesmo começo na matriz de busca...
                for(j=0;j<numClassesDaPalavra;j++){ //para cada classe da palavra com o
mesmo começo...
                    Valor=VetorPi[MatrizDeBusca[i].parB[j].indClasse]*
MatrizB1[MatrizDeBusca[i].parB[j].indClasse].Probabilidade[MatrizDeBusca[i].parB[j]
.PosiB];
                    Maiores(Valor,i);
                }
            }
        }
    }
}
}
}

```

```

    }
    else{
        for(i=0;i<numPalavrasDiferentes;i++){//para cada palavra sendo digitada...
            PalavraDaMatriz=MatrizDeBusca[i].Palavra;
            if(pedaco.CompareNoCase(PalavraDaMatriz.Left(pedaco.GetLength()))==0){
                for(j=0;j<numClassesDaPalavra;j++){ //para cada delta anterior...
                    if(mtzPred[indPred-1].delta[j]>0){ //que for diferente de 0...
                        for(k=0;k<numClassesDaPalavra;k++){//para cada classe da palavra sendo
digitada...
                            Valor=mtzPred[indPred-1].delta[j]*
                                MatrizA[mtzPred[indPred-
1].classe[j]][MatrizDeBusca[i].parB[k].indClasse]*
MatrizB1[MatrizDeBusca[i].parB[k].indClasse].Probabilidade[MatrizDeBusca[i].parB[k]
.PosiB];
                                Maires(Valor,i);
                            }
                        }
                    }
                }
            }
        }
        CarregaLista(indPred);
    }
}
//*****
****
void CTecladoVirtualDlg::OnContaClique()
{
    /** Conta Clique
    m_veditContaClique=0 ;
    m_cEditContaClique.ZerarContador();
    Invalidate();
    UpdateData(false);
}
//*****
****
void CTecladoVirtualDlg::OnTempoClique()
{
    // TODO: Add your control notification handler code here
    /** Conta Clique
    CString texto;
    m_cButtonTempoClique.GetWindowText(texto);
    if(texto=="Iniciar")
    {
        m_cButtonTempoClique.SetWindowText("Parar");
        UpdateData(true);
        m_cEditContaClique.SetTimer(1,1000,0);
    }
    if(texto=="Parar")
    {
        m_cButtonTempoClique.SetWindowText("Iniciar");
        UpdateData(true);
        m_cEditContaClique.KillTimer(1);
    }
    UpdateData(false);
}
}

```

ANEXO I

TEXTOS DE TESTE

Aqui são apresentados os textos utilizados na realização dos testes realizados com o aplicativo Teclado Virtual Livre em conjunto com a técnica de predição baseada nos Modelos de Markov.

A Lagarta (fonte: <http://www.usinadeletras.com.br/>)

Autor: Leonor von Osterroht

Era uma lagarta que, a caminhar no verde galhinho, foi procurar um bom lugarzinho para descansar, dormir muito tempo e assim fabricar um fio bem longo, onde se enrolou e quando acordou, que bela surpresa, virou borboleta. Foi uma beleza!

Letrinhas (fonte: <http://www.usinadeletras.com.br/>)

Autor: Maria da Graça Almeida

O g do gato... como é que é?

O g do gato tem gola e pé.

O p do pato tem papo e pé.

O p do ponto, ponto não é!

O b do pode berrar não pode.

O b da barba não tem bigode.

Do q do queijo eu não me esqueço,

o q do queijo é o g do avesso.

O t tem traço, o t tem fio. Sem embaraço escrevo tio.

O z da zebra, zebra não é. Zebra tem mais do que um pé.

O z da zebra faço com fé. Com z de zebra escrevo Zé.

O Menino e a Moeda (fonte: <http://www.contos.poesias.nom.br/>)

Autor: Maria Hilda de J. Alão.

Um menino queria comprar uma bolinha de gude para completar sua coleção. Sem dinheiro, ele caminhava por uma estrada e, em certo trecho, achou uma moeda. Abaixou-se para apanhá-la. Neste instante passou por ele, voando, uma nota de maior valor. Ele largou a moeda e saiu correndo para alcançar a nota. Com ela poderia comprar muitas bolinhas de gude, sonhava. Quanto mais ele corria mais a nota se afastava. Furioso, o menino gritou:

- Olá “seu” vento! Esta nota não te pertence, por que a levas para longe?

O vento, zunindo forte, respondeu:

- Quero ver o que és capaz de fazer por ela. Ela é um sonho, o desejo de querer mais e mais, também chamado de ganância, a certeza tu a deixaste para trás.

E lá se foi o menino disputando com o vento a posse da nota. Subiu e desceu morros, embrenhou-se no mato, e o vento continuava rindo dele levando a nota para bem distante. Depois de muito tempo de corrida, já cansado, o menino percebeu que estava se aproximando da borda de um precipício. Parou. O vento continuou levando a nota até que ela despencou, abismo abaixo, caindo nas águas do mar. Foi como acordar de um pesadelo. O pobre menino pensou:

- Tanto trabalho para nada. É melhor eu voltar e pegar a moeda. Com ela eu posso comprar a bolinha que me falta.

Voltou ao local onde deixara a moeda, mas ela havia desaparecido. Algum passante encontrou e levou a moeda com ele.

- Tá vendo! – pensou o menino -, quem manda trocar o certo pelo incerto! Não tenho a moeda nem a nota e, muito menos, a bolinha que me falta.

Texto 36 do *corpus*

O Lobo e a Garça

Um lobo, tendo se engasgado com um pedaço de osso, contratou uma garça por uma grande soma em dinheiro, para ela colocar a cabeça dentro da sua garganta e de lá retirar o osso. Quando a garça retirou o osso e pediu o pagamento que tinham combinado, o lobo, rangendo os dentes, exclamou:

- Ora, Ora! Você já foi recompensada. Ao ser permitido que sua cabeça saísse a salvo de dentro da boca e mandíbulas de um lobo você foi paga.

Ao servir a alguém de má índole, não espere recompensas, ao contrário, agradeça se ele lhe virar as costas e for embora.

Último Parágrafo do Texto 98 do *corpus*

Branca de Neve

A horrível mulher fitava-a como uma serpente ao fascinar um passarinho. Mas sobre o braseiro já estavam prontos um par de sapatos de ferro, que haviam ficado a esquentar em

ponto de brasa; os anões apoderaram-se dela e, calçando-lhe à força aqueles sapatos quentes como fogo, obrigaram-na a dançar, a dançar, a dançar, até cair morta no chão.

Em seguida, realizou-se a festa com um esplendor jamais visto sobre a terra, e todos, grandes e pequenos, ficaram profundamente alegres.

Parágrafos 1, 2 e 3 do Texto 54 do *corpus*

Chapeuzinho Vermelho

Era uma vez, uma menina tão doce e meiga que todos gostavam dela. A avó, então, a adorava, e não sabia mais que presente dar a criança para agradá-la.

Um dia ela presenteou-a com um chapeuzinho de veludo vermelho.

O chapeuzinho agradou tanto a menina e ficou tão bem nela, que ela queria ficar com ele o tempo todo. Por causa disso, ficou conhecida como Chapeuzinho Vermelho.

ANEXO II

CARTA DE APROVAÇÃO DO COMITÊ DE ÉTICA



Pontifícia Universidade Católica do Paraná
Pró-Reitoria Acadêmica e de Pesquisa
Núcleo de Bioética

Curitiba, 01 de dezembro de 2006.
Of. 731/06/CEP-PUCPR

Ref. "Predição de palavras baseada em modelos de Markov: Um acelerador de uso de sistemas de comunicação com auxílio do computador"

Prezado (a) Pesquisador (es),

Venho por meio deste informar a Vossa Senhoria que o Comitê de Ética em Pesquisa da PUCPR, no dia 29 de outubro do corrente ano aprovou o Projeto Intitulado "Predição de palavras baseada em modelos de Markov: Um acelerador de uso de sistemas de comunicação com auxílio do computador", pertencente ao Grupo III, sob o registro no CEP nº 1398, e será encaminhado a CONEP para o devido cadastro. Lembro ao senhor (a) pesquisador (a) que é obrigatório encaminhar relatório anual parcial e relatório final a este CEP.

Atenciosamente,


Profª M. Sc Ana Cristina Miguez Ribeiro
Coordenadora do Comitê de Ética em Pesquisa - PUCPR

Ilma Sra
Monica Jordan

REFERÊNCIAS BIBLIOGRÁFICAS

- ABDALLA, D., GAMA, C., PINHEIRO, V., RODARTE, C., SANTOS, L.N. Onda Digital: A Universidade Protagonizando a Inclusão Digital. Anais do XI Workshop de Informática na Escola, XXV Congresso da Sociedade Brasileira de Computação. São Leopoldo-RS, jul., 2005.
- ANDRADE, R.L., SCHULZ, S., NOHAMA, P., MARKÓ, K., HAHN, U., PACHECO, E.J. The Morphosaurus Medical Subword Lexicon - Lexicographic and Semantic Aspects. Anais do Congresso da Sociedade Brasileira de Computação, III Workshop em Tecnologia da Informação. São Leopoldo. 2005.
- BARRETO, A.A. Mudança Estrutural no Fluxo do Conhecimento: a Comunicação Eletrônica. In: Ciência da Informação, Brasília, v.27, n.2, pp. 122-127, mai./ago., 1998.
- BARTH, C., SANTAROSA, L. Teclado Virtual: Uma Ferramenta Para a Escrita da Língua de Sinais. Anais do IV Congresso Iberoamericano sobre Tecnologias de Apoio a Portadores de Deficiência. Vitória: Copigraf, fev., 2006.
- BORGES, J.A. Recursos Tecnológicos para Acesso de Pessoas Cegas ao Computador - A Abordagem DOSVOX. Anais do Congresso da Reabilitação Internacional. Rio de Janeiro, 2000.
- BORGES, J. A. Projeto Motrix. Disponível em: <http://intervox.nce.ufrj.br/dosvox/intro.htm>. Acessado em set., 2005.
- BORGES, A.T.S. Localização do CEP no Bloco do Endereço em Envelopes Postais. 2004. Dissertação (Mestrado em Informática Aplicada) – Programa de Pós-Graduação em Informática Aplicada, PUCPR, Curitiba.
- BRITTO, A.S. A Two-Stage HMM Based Method for Recognizing Handwritten Numeral Strings. 2001. Tese (Doutorado em Informática Aplicada) – Programa de Pós-Graduação em Informática Aplicada, PUCPR, Curitiba.

- BRITTO, A.S., SABOURIN, R., BORTOLOZZI, F., SUEN, C.Y. The Recognition of Handwritten Numeral Strings Using a Two-Stage HMM Based Method. In: International Journal of Document Analysis and Recognition. V.5, pp. 102-117. 2003.
- BUENO, S. Mini-Dicionário da Língua Portuguesa. São Paulo: FTD, 2000.
- CANNAO, M. Comunicação Alternativa e Deficiência Mental. In: TUPY, T.M., PRAVETTONI, D.G. ...e se Falta a Palavra, qual Comunicação, Qual Linguagem? São Paulo: Memnon, 1999.
- CAPOVILLA, F.C. Pesquisa e Desenvolvimento de Novos Recursos Tecnológicos para Educação Especial: Boas Novas para Pesquisadores, Clínicos, Professores, Pais e Alunos. In: Alencar, E. (Org.). Tendências e Desafios de Educação Especial. Brasília: Secretaria de Educação Especial. pp. 196-211, 1994.
- CLARKSON, P.R. Adaptation of Statistical Language Models for Automatic Speech Recognition. 1999. Tese (Doutorado em Filosofia) – Cambridge University Engineering Department. University of Cambridge, Cambridge.
- DIAS, R.S.F. 2000. Normalização de Locutor em Sistema de Reconhecimento de Fala. Dissertação (Mestrado em Engenharia Elétrica) – Programa de Pós-Graduação em Engenharia Elétrica da Universidade Estadual de Campinas. UNICAMP, Campinas.
- DIAS, N., OSOWSKY, J., GAMBA, H.R., NOHAMA, P. Controle do Cursor do Mouse pelo Movimento da Cabeça Usando Câmera CCD e Processamento de Imagem. Anais do III Congresso Latino Americano de Engenharia Biomédica. Paraíba, set., 2004.
- EISNER, J. Na Interactive Spreadsheet for Teaching the Forward-Backward Algorithm. Anais do ACL Workshop on Effective Tools and Methodologies for Teaching NLP and CL. Philadelphia, jul., 2002.
- EVANS, D.G., DREW, R., BLENKHORN, P. Controlling a Mouse Pointer Position Using an Infrared Head-Operated Joystick. In: IEEE Transactions on Rehabilitation Engineering. v. 8, n.1, pp. 107-117. mar., 2000.

- FONSECA, P.M., CARVALHO, T. Teclado Virtual Dinâmico com Predição. Anais do VII Workshop de Engenharia Biomédica. Lisboa, Portugal, 2002.
- FREITAS, M.I.C., BORGES, J.A., VENTORINI, S.E., TAKANO, D.F., CUNHA, A.R. Maquete Tátil e DOSVOX – Os Desafios da Construção de um Sistema que Aprimore a Transmissão do Conhecimento para Alunos Cegos e de Baixa Visão: Resultados Parciais. Anais do III Congresso Iberoamericano sobre Tecnologias de Apoio a Portadores de Deficiência. São Jose, Costa Rica, mar., 2004.
- GASPERIN, C.V., LIMA, V.L.S. Fundamentos do Processamento Estatístico da Linguagem Natural. PUCRS: Technical Report Series, n.21, 2001.
- GAVA, M.L. AAC – Augmentative and Alternative Communication – Como Resposta às Deficiências Verbais. In: TUPY, T.M., PRAVETTONI, D.G. ...e se Falta a Palavra, qual Comunicação, Qual Linguagem? São Paulo: Memnon, 1999.
- GUSTAVIL, E., PETTERSSON, E. A Swedish Grammar for Word Prediction. 2003. Dissertação (Mestrado em Linguística Computacional) – Departamento de Linguística da Universidade de Uppsala, Suécia.
- HENZEN, A.F. Acessibilidade ao Computador para Portadores de Paralisia Cerebral. 2003. Dissertação (Mestrado em Engenharia Elétrica e Informática Industrial) – Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, UTFPR, Curitiba.
- HUNNICUTT, S., NOZADZE, L., CHIKOIDZE, G. Russian Word Prediction with Morphological Support. Anais do 5th International Symposium on Language, Logic and Computation. Tbilisi, Georgia, oct., 2006.
- IBGE. Censo Demográfico - 2000: Características Gerais da População. Disponível em: http://www.ibge.gov.br/home/estatistica/populacao/censo2000/default_populacao.shtm.
- JOHNSON, R. The Picture Communication Symbols. Book II. Solana Beach: Mayer Johnson Co, 1981.

- JOHNSON, R. The Picture Communication Symbols. Book II. Solana Beach: Mayer Johnson Co, 1985.
- JORDAN, M., BRANDANI, E.G., MARTINS, R.A., NOHAMA, P. Mouse Infravermelho Controlado pelos Movimentos da Cabeça – Uma Nova Solução. Anais do III Congresso Latino Americano de Engenharia Biomédica. Paraíba, set., 2004.
- JORDAN, M., MATIAS, D.H., NOHAMA, P. Amplisoft: Comunicação Alternativa para Todos. Anais do IV Congresso Iberoamericano sobre Tecnologias de Apoio a Portadores de Deficiência. Vitória: Copigraf, fev., 2006.
- JULIATO, M., TANAKA, E.H., BAUDET, C., GALVES, M., COELHO, T.T., ROCHA, H.V. TFlex: Proposta de Simulador de Teclado com Diferentes Varreduras e Layouts. Anais do VII Congresso Iberoamericano de Informática Educativa. Monterrey, 2004.
- KLARLUND, N., RILEY, M. Word N-Grams for Cluster Keyboard. Anais do Workshop on Language Modeling for Text Entry Methods. Hungria, 2003.
- KUHN, T., NIEMANN, H., SCHUKAT-TALAMAZZINI, E.G. Ergodic Hidden Markov Models and Polygrams for Language Modeling. In: IEEE International Conference on Acoustics, Speech, and Signal Processing. pp. I357-I360. Australia, 1994.
- LAW, H.H., CHAN, C. N-th Order Ergodic Multigram HMM for Modeling of Languages without Marked Word Boundaries. Disponível em:
<http://www.acl.ldc.upenn.edu/C/C96/C96-1036.pdf> . Acesso em jan., 2006.
- LEE, H., YEH, C., WU, C., TSUANG, M. Wireless Communication for Speech Impaired Subjects Via Portable Augmentative and Alternative System. Anais do 23th Annual International Conference of the IEEE, Engineering in Medicine and Biology Society. V.4, pp. 3777-3779. 2001.

- LEVINSON, S.E., RABINER, L.R., SONDHI, M.M. An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition. In: Bell System Technical Journal. v.62, n.4, pp.1035-1074, abr., 1983.
- LIEGEL, L.A., NOHAMA, P. Proposta de Layout de Teclado para Comunicação Alternativa. Anais do IV Congresso Iberoamericano sobre Tecnologias de Apoio a Portadores de Deficiência, Vitória: Copigraf, fev., 2006.
- LODI, L.H. Ética e Cidadania: Construindo Valores na Escola e na Sociedade – Módulo 4 – Inclusão Social. Brasília: Secretaria Especial dos Direitos Humanos do Ministério da Educação, SEIF, SEMTEC, SEED, 2003.
- MAHARAJ, S. Pictogram Ideogram Communication. In: The George Reed Foundation for the Handicapped. Canadá, 1980.
- MACKAY, D. Dasher - An Efficient Keyboard Alternative. Disponível em: <http://dasher.org.uk/Publications.html>. Acesso em jun., 2004.
- MATIAS, D.H., NOHAMA, P. Teclado Virtual Alfanumérico com Predição de Palavras. Anais do II Seminário e I Oficinas ATIID. v. 1. pp. 1-4. São Paulo, 2003.
- MATIAS, D.H.; JORDAN, M.; NOHAMA, P. Interface de Comunicação para Crianças com Problemas Motores e Vocais. Anais do IV Congresso Iberoamericano sobre Tecnologias de Apoio a Portadores de Deficiência, São José, Costa Rica, mar., 2004.
- MARTINS, J.A. Avaliação de Diferentes Técnicas para Reconhecimento da Fala. 1997. Tese (Doutorado em Engenharia Elétrica) – Programa de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, UNICAM, Campinas.
- MITTLER, P. Educação para Todos. In: Revista Pátio On-Line. n.20. abr., 2002.
- MIZUKO, M. Personal Computers as Augmentative and Alternative Communication Aids. In: American Journal of Speech-Language Pathology, v.2, n.3, pp.8–10, set., 1993.

- MOECKEL, A. Semáforo para Deficientes Visuais. In: Semana de Iniciação Científica e Tecnológica. Rio de Janeiro, 1992.
- MONTEIRO, A.R. O Direito à Educação. Lisboa: Livros Horizontes, 1998.
- MORETTI, G. Princípios e Significados da Comunicação Alternativa. In: TUPY, T.M., PRAVETTONI, D.G. ...e se Falta a Palavra, qual Comunicação, Qual Linguagem? São Paulo: Memnon, 1999.
- MULLER, R.W., SCHMAL, R.M., JORDAN, M., NOHAMA, P. A Novel Ultrasonic Mouse Activated by Head Movement. Anais da 11th Biennial Conference of the International Society for Augmentative and Alternative Communication. v.11, pp.1-12. Natal, out., 2004.
- NAKAMURA, M., MARUYAMA, K., KAWABATA, T., SHIKANO, K. Neural Network Approach to Word Category Prediction for English Texts. In: Proceedings of the International Conference on Computational Linguistics. pp 213-218. Helsinki, 1990.
- NERI, M.C., SOARES, W.L. Idade, Incapacidade e o Número de Pessoas com Deficiência. In: Revista Brasileira de Estudos de População. v.21, n.2, pp. 303-321. Campinas, jul./dez., 2004
- NIESLER, T.R., WOODLAND, P.C., Computer Speech and Language. In: Academic Press, n.13, p.99-124. Reino Unido, 1999.
- NOGUEIRA, F. Modelagem e Simulação – Cadeias de Markov. Disponível em: www.engprod.ufjf.br/fernando/epd042/cadeiaMarkov.pdf . Acesso em dez., 2005.
- NOHAMA, P., LINARTH, A.G., LIPPMANN JR., RIELLA, R.J. VOXsis: Sistema Computadorizado de Auxílio à Fala para Deficientes Auditivos. In: Saber Eletrônica. v. 37, n. 341, pp. 23-28. São Paulo, 2001.

- NUNES, L.R.O.P. E se Não Podemos Falar? A Comunicação Alternativa para Portadores de Distúrbios da Fala. In: *Consciência: Boletim Informativo do Conselho Regional de Psicologia do Rio de Janeiro*. v.1, n.3, pp.12-13. Rio de Janeiro, 1999.
- ORTIZ, K.Z. (Org.) *Distúrbios Neurológicos Adquiridos – Linguagem e Cognição*. São Paulo: Manole, 2005.
- PARDO, T.A.S., RINO, L.H.M. TeMário: Um Corpus para Sumarização Automática de Textos. *Série de Relatórios do Núcleo Interinstitucional de Linguística Computacional*. São Carlos, out., 2003.
- PATEL, S. A Lower-Complexity Viterbi Algorithm. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, v.1, pp. 592-595. may , 1995.
- PELOSI, M. *Software Comunique – Software de Autoria para o Desenvolvimento da Comunicação Oral e Escrita de Pacientes com Dificuldades Motoras Graves*. In: QUEVEDO, A.A.F., OLIVEIRA, J.R., MANTOAN, M.T.E. (Orgs.). *Mobilidade e Comunicação – Desafios à Tecnologia e à Inclusão Social*. Campinas: Edição Própria, 1999.
- PELOSI, M. *Apostila de Comunicação Alternativa e Ampliada (CAA)*. Disponível em: <http://www.comunicacaoalternativa.com.br>. Acesso em set., 2003.
- PESCINA, G.F. *Desarrollando Tecnologia Asistida para Personas com Discapacidad Multiple em Países em Desarrollo*. Anais do III Congresso Iberoamericano sobre Tecnologias de Apoio a Portadores de Deficiência. San Jose, Costa Rica, mar., 2004.
- PELOSI, M. *Software Comunique – Software de autoria para o desenvolvimento da comunicação oral e escrita de pacientes com dificuldades motoras graves*. In: A.A.F. Quevedo, J.R. Oliveira e M.T.E. Mantoan (eds). *Mobilidade e Comunicação – Desafios à Tecnologia e Inclusão Social* (pp. 173-176). Campinas: Edição do Autor.
- PEÑA-CASANOVA, J., PAMIES, M.P. *Reabilitação da Afasia e Transtornos Associados*. 2ª ed. São Paulo: Manole. 2005.

- PINHEIRO, P.C.P. Inclusão Digital, Inclusão Social e o Deficiente Físico. Disponível em: <http://www.crp.ce.gov.br/artigosinclusaodigital.htm#patricia>. Acesso em jan. 2006.
- PINKER, S., TROVATO, R. The Language Instinct, In: Europa Scienze Umane. n.143-44, pp. 18-23. Milano, Itália, 1997.
- POLLO, T.C., Brazilian Portuguese Preschool Word List. Disponível em: http://www.artsci.wustl.edu/~tcpollo/palavras_explic.html. Acesso em jul., 2003.
- RABINER, L.R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In: Proceedings of the IEEE. V. 77, n. 2, pp. 257-286. fev., 1989.
- RAY, S., CRAVEN, M. Representing Sentence Structure in Hidden Markov Models for Information Extraction. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence. 2001.
- RECSKI, L.Utilizando *Corpora* de Aprendizizes para a Investigação de Aspectos Discursivos, Metodologias de Ensino e Design de Material Pedagógico. In: Linguagem & Ensino, v. 8, n. 2, pp. 249-273. 2005.
- ROSA, J.L.G. Next Word Prediction in a Connectionist Distributed Representation System. In: Proceedings of the 2002 IEEE International Conference on Systems, Man, and Cybernetics. Tunisia. out., 2002.
- ROSENFELD, R. Two Decades of Statistical Language Modeling: Where Do We Go from Here?. In: Proceedings of IEEE. v.88, n. 8, pp. 1270-1278, aug., 2000.
- SARDINHA, T.B. Lingüística de *Corpus*: Histórico e Problemática. São Paulo. Delta. v.16, n. 2, 2000. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-44502000000200005&lng=pt&nrm=iso. Acesso em abr, 2006
- SARDINHA, T.B. Lingüística de *Corpus*. São Paulo: Manole, 2004.

- SCAVONE, A.P.R. Reconhecimento de Palavras por Modelos Ocultos de Markov. 1996. Dissertação (Mestrado em Engenharia) – Programa de Pós-Graduação da Escola Politécnica da Universidade de São Paulo, USP, São Paulo.
- SILVEIRA, S. Inclusão Digital, Software Livre e Globalização Contra-Hegemônica. In: Parcerias Estratégicas, Seminários Temáticos para a 3ª Conferência Nacional de Ciência, Tecnologia e Inovação. n.20, pp. 421-446. jun., 2005.
- SINGH, R. Treinando HMMs Densidade Contínua. Sistemas de Reconhecimento de Voz. Disponível em www.universiabrasil.net/mit/6/6345/PDF/lecture15.pdf. Acessado em jul., 2006.
- SOUKOREFF, R.W. Text Entry for Mobile Systems: Models, Measures, and Analyses for Text Entry Research. 2002. Dissertação (Mestrado em Ciência) – The Faculty of Graduate Studies of York University, York.
- SOUSA, M.T.R. Uma Abordagem sobre o Problema da Mobilidade e Acessibilidade do Transporte Coletivo: o Caso do Bairro Jardim São João no Município de Guarulhos – SP. 2003. Dissertação (Mestrado em Engenharia Civil) – Comissão de Pós-Graduação da Faculdade de Engenharia Civil, Arquitetura e Urbanismo da Universidade Estadual de Campinas. UNICAMP, Campinas.
- STELLA, G. Deficiências Cognitivas e Uso dos Procedimentos Informatizados. In: TUPY, T.M., PRAVETTONI, D.G. ...e se Falta a Palavra, qual Comunicação, Qual Linguagem? São Paulo: Memnon, 1999.
- TERCEIRO, A. Métodos Probabilísticos de Pesquisa Operacional. Disponível em: <http://twiki.im.ufba.br/bin/view/MAT160/ProcessosEstocasticos>. Acesso em fev., 2006.
- THEDE, S.M., HARPER, M.P. A SecondOrder Hidden Markov Model for PartofSpeech Tagging. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics. pp. 175-182. Maryland, 1999.

TUPY, T.M., PRAVETTONI, D.G. ...e se Falta a Palavra, qual Comunicação, Qual Linguagem? São Paulo: Memnon, 1999.

VASCONCELOS, R. Paralisia Cerebral: a Fala na Escrita. 1999. Dissertação (Mestrado em Lingüística Aplicada e Estudos da Linguagem) – Pontifícia Universidade Católica de São Paulo. PUCSP, São Paulo.

WANG, X. Incorporating Knowledge on Segmental Duration in HMM-Based Continuous Speech Recognition. 1997. Tese (Doutorado) – The University of Amsterdam – Amsterdam.

WOODCOCK, R., CLARK, C., DAVIES, C.O., Peabody Rebus Program. In: American Guidance Service Inc, 1983.

ZIMMERMANN, M., BUNKE, H. N-Gram Language Models for Offline Handwritten Text Recognition. In: Proceedings of the 9th Workshop on Frontiers in Handwriting Recognition. 2004.

RESUMO

A busca pela inclusão social tem promovido a pesquisa de ferramentas que ampliam o uso do computador por pessoas com necessidades especiais. Dentro da gama de ferramentas auxiliares, um importante acelerador de uso é a predição de palavras, que diminui o número de ações a ser executado e, conseqüentemente, tempo e esforços envolvidos. A antecipação de uma palavra dentro de um texto digitado é denominada predição de palavras, e pode ser desenvolvida a partir de considerações estatísticas como contagem de ocorrências, suas antecessoras e sucessoras. Um processo de análise estocástica muito utilizado na predição de acontecimentos, como reconhecimento da fala, envolve os Modelos Ocultos de Markov. A literatura aponta tal processo como um dos mais indicados para a predição de palavras, quando baseado num conjunto de textos etiquetado, de forma a apresentar a classe gramatical de cada uma de suas palavras, chamado *corpus*. Então, uniu-se a eficiência de um aplicativo desenvolvido para usuários com necessidades especiais à ferramenta de predição baseada em Modelos Ocultos de Markov. Desenvolveu-se um algoritmo, que foi associado a um simulador de teclado livre. O aplicativo final oferece ao usuário uma lista com as dez palavras próximas mais prováveis de ocorrerem, considerando a classe gramatical mais provável, e, dentro dos grupos de classes, as palavras de maior ocorrência. O algoritmo baseia-se num *corpus* elaborado com textos infantis criteriosamente selecionados. Realizaram-se testes de digitação de textos do *corpus* a fim de verificar a eficiência e coerência do algoritmo; testes contendo preparo de textos não presentes no *corpus*, observando a capacidade de predição e consistência textual do aplicativo; e, enfim, testes com dois grupos de voluntários, sendo um deles de pessoas com necessidades especiais, para verificar a funcionalidade e aplicabilidade. Os resultados mostram que o aplicativo diminui consideravelmente a dificuldade na produção textual, e oferece ao usuário palavras gramaticalmente corretas. Os Modelos Ocultos de Markov tornaram o processo de predição rápido e coerente. Assim, aliando o desenvolvimento de software à Engenharia de Reabilitação, este trabalho oferece uma técnica facilitadora para a produção de textos, colaborando para a inclusão digital e, conseqüentemente, social, de pessoas, muitas vezes, apontadas como incapazes de viverem independentemente.

PALAVRAS-CHAVE

Software, educação inclusiva, informática educativa, produção de textos pelo aluno, apoio à pessoa com deficiência, Modelos Ocultos de Markov.

ÁREA/SUB-ÁREA DE CONHECIMENTO

- 1.03.03.04 - 9 : Sistemas de Informação
- 1.03.02.02 - 6 : Modelos Analíticos e de Simulação
- 1.03.03.03 - 0 : Banco de Dados

2007

456

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)