VERA LÚCIA COSTA DE MEDEIROS

PROJETO DE UM VERIFICADOR ORTOGRAFICO PARA A LINGUA PORTUGUESA

Dissertação apresentada ao Curso de MESTRADO EM SISTEMAS E COMPUTAÇÃO da Universidade Federal da Paraíba, em cumprimento às exigências para obtenção do Grau de Mestre.

JACQUES PHILIPPE SAUVÉ

Orientador

M488p Medeiros, Vera Lucia Costa de
Projeto de um verificador ortografico para a lingua portuguesa / Vera Lucia Costa de Medeiros. - Campina Grande, 1988.
88 f. : il.

Dissertacao (Mestrado em Sistemas e Computacao) - Universidade Federal da Paraiba, Centro de Ciencias e Tecnologia.

1. Tratamento de Erros (Informatica) 2. Verificador Ortografico para a Lingua Portuguesa - 3. Verificacao Ortografica de Textos 4. Dissertacao I. Sauve, Jacques Philippe, Dr. II. Universidade Federal da Paraiba - Campina Grande (PB) III. Título

CDU 004.052.4(043)

PROJETO DE UM VERIFICADOR ORTOGRAFICO PARA A LINGUA PORTUGUESA

VERA LUCIA COSTA DE MEDEIROS

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇAO DO CURSO DE POS-GRADUAÇAO EM SISTEMAS E COMPUTAÇAO DA UNIVERSIDADE FEDERAL DA PARAIBA COMO PARTE DOS REQUISITOS NECESSARIOS PARA A OBTENÇAO DO GRAU DE MESTRE EM CIENCIAS (M.Sc.).

Aprovada por:

JACQUES PHILIPPE SAUVE - Ph.D

- Presidente -

SILVIO R. DE LEMOS MEIRA - Ph.D

- Examinador -

JOSE ANTAO BELTRAO MOURA - Ph.D

- Examinador -

CAMPINA GRANDE

ESTADO DA PARAIBA - BRASIL

JANEIRO - 1988

PROJETO DE UM VERIFICADOR ORTOGRAFICO PARA A LINGUA PORTUGUESA

RESUMO

Criar um texto com o auxílio do computador tem se tornado uma tarefa cada vez mais comum nos dias de hoje. Este processo, no entanto, além de não impedir a ocorrência de erros ortográficos resultantes da ignorância do autor, contribui para que novos erros ocorram, sendo estes decorrentes do próprio processo de digitação do texto.

A detecção destes erros através de verificação automática pode ser extremamente útil no processo de revisão de textos, mesmo sabendo-se que este processo envolve a verificação de outros aspectos que não apenas a ortografia.

Este trabalho tem relação com a verificação ortográfica de textos, mais precisamente, com textos escritos na língua portuguesa. O principal objetivo do trabalho consiste em projetar um verificador ortográfico para a língua portuguesa , e criar um método automático de obtenção e criação do dicionário, no qual o verificador se baseia.

SUMARIO

FIGURAS

PROJETO DE UM VERIFICADOR ORTOGRAFICO PARA A LINGUA PORTUGUESA

1. INTRODUÇÃO

O uso do computador na preparação de documentos torna-se cada vez mais extenso, contribuindo para o crescimento do mercado de Sistemas de Processamento da Palavra.

Um sistema operacional típico de computador dispõe de sistemas de arquivos, editores e formatadores de texto. O usuário cria um documento com o editor de texto, armazena-o no sistema de arquivos e usa o formatador de texto para interpretar os comandos de formatação existentes no documento.

No entanto, este método de preparação de documentos não impede a ocorrência de erros ortográficos resultantes da ignorância do autor. Por sua vez, o processo de digitação contribui para a inclusão de erros ortográficos dos quais 80% são causados por [Damerau citado por PETE 80]:

1. transposição de duas letras adjacentes;
2. uma letra extra;
3. omissão de uma letra;
4. uma letra incorreta;

Mas erros ortográficos irritam os leitores. E para a maioria das pessoas, a revisão de um texto objetivando detectá-los é uma tarefa enfadonha e de difícil sucesso. Felizmente este procedimento é perfeitamente adaptável ao computador: trabalho

monótono e repetitivo que requer uma rápida leitura e uma boa memória [BENT 85]. É possível um sistema analisar a ortografia de um texto apontando as palavras potencialmente incorretas e suas prováveis correções.

Existem dois tipos de análise ortográfica: a verificação e a correção. Dado um arquivo texto como entrada, o verificador ortográfico detecta todas as palavras que estão incorretas. O corretor ortográfico, além de detectar as palavras incorretas, apresenta as palavras corretas que mais se "parecem" com as incorretas, partindo do princípio que o erro é resultado das falhas comuns ao processo de digitação [PETE 80].

Ao analisar um texto, um verificador ortográfico pode cometer dois tipos de erro: não detectar uma palavra ortograficamente incorreta, ou classificar como incorreta um palavra ortograficamente correta. Todos os verificadores ortográficos cometem erros - se muitos ou poucos, é uma questão de projeto [BENT 85].

O fato de um verificador ortográfico cometer erros não impede o seu uso, desde que a taxa de erro seja aceitável. Dos dois tipos de erro citados acima, o mais grave é aquele em que o verificador não detecta uma palavra ortograficamente incorreta, fazendo com que o usuário não tome conhecimento da existência do erro no texto.

Segundo Peterson [PETE 80], todos os sistemas de processamento de textos futuros apresentarão análise ortográfica. A existência de analisadores ortográficos sugere que

o passo seguinte será criar analisadores com maior nível de sofisticação, que verifiquem as estruturas sintática e semântica de um texto. Cabe a um verificador sintático analisar se cada sentença do texto encontra-se apropriadamente construída e sintaticamente correta. O verificador semântico analisa se as idéias do texto estão corretamente desenvolvidas e apresentadas, e se o documento se encontra completo e consistente. Fazer uma análise sintática de um texto de forma isolada da análise semântica nem sempre leva a um bom resultado, considerando que frequentemente a semântica das palavras tem efeito sobre a sintaxe da frase em que as mesmas estão inseridas. Para cada verificador deverá existir um corretor específico capaz de resolver quaisquer problemas detectados pelo primeiro.

A verificação ortográfica considera a palavra individualmente, não levando em conta o contexto em que a mesma se encontra, nem a sua semântica.

O assunto abordado neste trabalho é **verificação ortográfica.** Antes de especificarmos os propósitos do mesmo, apresentaremos, de forma sucinta, algumas características básicas deste tipo de verificação, tais como, métodos utilizados e tipos de processamento.

1.1 Métodos de verificação ortográfica

Segundo Turba [TURB 81], os métodos utilizados na verificação ortográfica podem se basear em análise de frequência de digramas e trigramas (sequência de dois e três caracteres, respectivamente) nas palavras do texto ou em dicionários.

1.1.1 Análise estatística

A análise estatística é um método baseado na frequência de digramas e trigramas nas palavras do texto. Aquelas que contêm digramas e trigramas infrequentes, com relação ao restante do texto, ou com relação à língua, sem se basear no texto, são classificadas como bastante peculiares e, consequentemente, como possíveis palavras ortograficamente incorretas.

Uma das vantagens do método é a redução significativa da memória necessária à verificação ortográfica, dispensando a obrigatoriedade da existência de um arquivo de palavras válidas (dicionário) usado em um processo comparativo com o texto. Uma outra vantagem é que a lista de palavras distintas do texto, apresentada ao usuário após a verificação, se encontra decrescentemente ordenada pelo índice de peculiaridade. Isto faz com que as palavras incorretas - palavras com alta peculiaridade - se encontrem no início da lista, estimulando o usuário a pesquisá-las [PETE 80].

No entanto, a taxa de erro apresentada pelo método é alta, já que muitas palavras ortograficamente válidas apresentam índice de peculiaridade alto, enquanto que palavras incorretas

apresentam frequentemente um baixo índice de peculiaridade. Uma outra desvantagem é o tamanho da lista apresentada ao usuário contendo todas as palavras distintas do texto. Uma forma de solucionar este problema é considerar a lista só até um certo grau de peculiaridade.

O **TYPO**, um dos verificadores ortográficos do sistema Unix, faz uso deste método. Com o objetivo de reduzir o tamanho da lista de palavras distintas apresentadas ao usuário, o **TYPO** faz uso de uma lista de palavras válidas e de uso frequente em um processo comparativo, eliminando da primeira todas as palavras que são comuns à ambas [PETE 80] [MORR 75].

1.1.2 Análise de afixos

A análise de afixos é um método de verificação ortográfica que consiste na retirada dos afixos (prefixos e sufixos) de uma palavra até reduzí-la ao seu radical, e em uma posterior pesquisa deste radical no dicionário.

Uma das vantagens do método é que um pequeno dicionário abrange um grande número de palavras, já que muitas delas são variações de uma mesma entrada (radical) do dicionário. Mesmo não estando representadas explicitamente, estas variações podem ser reconhecidas. Uma outra vantagem é que quaisquer prefixos e/ou sufixos podem se unir a qualquer radical, tornando o processo de verificação bastante flexível.

No entanto, é esta flexibilidade que introduz falhas no método. Radical e afixos isoladamente válidos podem conjuntamente

formar uma palavra ortograficamente inválida. Uma vez retirados os afixos desta palavra, pesquisado e encontrado o radical no dicionário, o verificador ortográfico classifica a palavra inicial como correta, não exercendo nenhuma espécie de controle sobre as leis de formação da mesma. Uma outra desvantagem apresentada pelo método é o nível de complexidade do algorítmo responsável pela retirada dos afixos de cada palavra a ser analisada.

Aceitar palavras incorretas formadas a partir de radical e afixos corretos pode ser evitado com o uso de sinalizadores (flags) que associam a cada radical do dicionário os afixos que podem se ligar ao mesmo, criando uma palavra válida. Esta técnica foi adotada pelos verificadores ortográficos **SPELL** do sistema **Unix** [McIL 82] e **SPELL** do **DEC-10** [PETE 80].

1.1.3 Pesquisa global

Um outro método utilizado na verificação ortográfica consiste na pesquisa da palavra integral em um grande dicionário onde se encontram todas as variações válidas das palavras que se deseja abranger. Uma das vantagens encontradas neste método é que todas as palavras incorretas são detectadas, e as demais classificadas como ortograficamente corretas. O método permite que se faça, quando necessário, uma seleção das palavras que devam constar nos textos, forçando, por exemplo, o uso de um vocabulário limitado voltado para uma determinada classe de leitores.

Como desvantagens, o método apresenta o uso de um grande espaço para armazenamento do dicionário e um aumento no tempo de pesquisa, com relação ao método anterior, devido ao crescimento do dicionário. Este método é utilizado pelo verificador ortográfico da IBM [PETE 80].

Os métodos de verificação ortográfica existentes apresentam vantagens e desvantagens com relação a abrangência, velocidade, e requisitos de espaço em memória principal e/ou secundária. A opção por um determinado método requer, do projetista, uma prévia escolha das características que o verificador deverá apresentar. O ideal seria que cada uma destas características se comportasse conforme o desejado. Mas uma maior abrangência implica em um maior espaço de armazenamento e em uma menor velocidade.

Alguns projetistas criam verificadores que se baseiam em mais de um método, visando atingir um melhor ponto de equilíbrio entre estas características [TURB 81].

1.2 Tipos de processamento

De acordo com a interface com o usuário, o verificador ortográfico (VO) tem processamento "batch" ou "on-line". Esta é uma das diferenças básicas entre os diversos verificadores ortográficos existentes [TURB 81].

No processamento "batch" a verificação se faz levando-se em conta todo o texto. Como as duplicações das palavras são frequentes, ordena-se o texto objetivando eliminar estas duplicações. Uma vez ordenado, e com as duplicações de palavras

eliminadas, o texto é submetido a verificação onde cada palavra é verificada de forma isolada. Com o dicionário também ordenado, esta verificação se processa com apenas um passo.

Um verificador com processamento "on-line", ou processamento interativo, caminha no texto até detectar uma palavra incorreta, suspende a verificação e interage com o usuário solicitando o tipo de procedimento a ser tomado sobre aquela palavra - deixar inalterada, modificar, permutar por outra(s) palavra(s), etc. Em seguida, o processo de verificação tem continuidade. A verificação é feita para cada ocorrência da palavra no texto, o que contribui para uma redução na velocidade do verificador, já que o número de duplicações de palavras em um texto é normalmente bastante alto.

O processamento "batch" apresenta a vantagem de verificar apenas uma ocorrência da palavra no texto, e a desvantagem de apresentar as palavras incorretas ao usuário apenas quando o texto já foi totalmente submetido a verificação. O processamento "on-line", ao contrário, apresenta a vantagem de interagir com o usuário logo que detecta uma palavra incorreta no texto, mas verifica uma palavra tantas vezes quantas forem suas ocorrências no texto.

Nós chamamos de "batch + semi-interativo" o processamento em que o verificador analisa todo o texto para, em seguida, manter um processo interativo (opcional) com o usuário, onde este determina se a palavra está realmente incorreta, ou se o verificador deve incluí-la no dicionário.

A apresentação das palavras incorretas através de um processamento interativo é mais estimulante do que através de um processo não interativo. A probabilidade do usuário deixar de perceber uma palavra incorreta diminui, já que cada palavra é apresentada individualmente, em oposição à exibição de uma palavra incorreta entre várias corretas.

1.3 Objetivos do trabalho

Os propósitos deste trabalho consitem em elaborar um projeto de um verificador ortográfico para a língua portuguesa (VOLP), sem , no entanto, implementá-lo. O VOLP fará uso de dicionário, terá um processamento "batch + semi-interativo" e adotará o método análise de afixos associado ao uso de regras que disciplinam a formação das palavras.

1.4 Organização da tese

Esta tese é constituída de 8 capítulos.

O CAPITULO 1 corresponde a esta introdução.

O CAPITULO 2 apresenta os problemas básicos na construção de um verificador ortográfico, abrangendo desde a definição de "palavra" até a criação e manutenção de dicionários.

O CAPITULO 3 fala sobre os principais verificadores ortográficos para a língua inglesa de que se tem conhecimento. Traça ainda um perfil de um verificador ortográfico para a língua portuguesa.

O CAPITULO 4 apresenta as características de um verificador ortográfico voltado para a língua portuguesa (VOLP), e as soluções encontradas para os problemas levantados no capítulo 2.

O CAPITULO 5 mostra as estruturas de dados utilizadas pelo VOLP, e apresenta as razões pelas quais elas foram projetadas daquela forma.

O CAPITULO 6 é dedicado ao algorítmo geral do VOLP.

O CAPITULO 7 apresenta os métodos criados para obtenção e criação do dicionário do VOLP.

Finalmente, o CAPITULO 8 apresenta a conclusão do trabalho.

2. PROBLEMAS BÁSICOS NA CONSTRUÇÃO DE UM VERIFICADOR ORTOGRÁFICO

Seria bastante útil um verificador que verificasse a estrutura das sentenças, o uso apropriado das palavras, além da ortografia propriamente dita. Um verificador que fizesse exatamente o que qualquer pessoa faz ao revisar um texto.

Segundo Turba [TURB 81], uma completa verificação automática é uma pretensão bastante otimista, uma vez que um alto grau de conhecimento, por parte do programa, se faz necessário. As experiências feitas, envolvendo vocabulário muito restrito e limitadas estruturas de sentenças, estão ainda bastante distantes da realidade, onde normalmente se encontram textos com um extenso vocabulário e um grande número de estruturas de sentenças.

Mas, tendo um verificador ortográfico um menor nível de sofisticação, sua construção é elementar e livre de maiores problemas? Observa-se que este pressuposto é falso, e as razões que nos levam a chegar a tal conclusão constituem este capítulo.

2.1 Formação e reconhecimento de palavra

Partindo do princípio que um verificador ortográfico faz uma verificação a nível de palavra, deve-se definir, inicialmente, o que é uma palavra, isto é, o que constitui um átomo de verificação.

Para efeito de reconhecimento de palavras, supõe-se que o texto a ser verificado é constituído de palavras (formadas de

caracteres pertencentes ao alfabeto de palavra) separadas por delimitadores (formados por caracteres pertencentes a um alfabeto de delimitadores). Geralmente imagina-se uma palavra como sendo composta por letras maiúsculas e minúsculas. No entanto, ela pode conter caracteres acentuados, hífen (-), apóstrofe ('), dígitos (0...9) e outros caracteres. Há necessidade, portanto, de analisar a freqüência de uso destes caracteres nas palavras da língua e decidir quais deles serão incluídos no alfabeto de palavras e quais serão incluídos no alfabeto de delimitadores.

Determinar o que delimita uma palavra às vezes não é tão simples quanto parece - qualquer coisa que não é palavra deve delimitar uma palavra. Um fim de linha pode não representar um fim de palavra - o usuário, ao formatar um texto usa hífen no final da linha separando sílabas de uma mesma palavra.

Um outro ponto a considerar são os comandos encontrados nos textos, comandos gerados e interpretados por um formatador de texto, que parecem palavras, mas que não devem ser tratados como tal. Estes comandos variam de um formatador para outro, exigindo que o verificador seja capaz de manipular diversos tipos de comandos, ou que seja voltado exclusivamente para textos gerados por um determinado formatador. No primeiro caso, são necessários vários conjuntos de regras de reconhecimento de palavras, enquanto que no segundo, faz-se necessário apenas um conjunto.

Um outro problema apresentado no âmbito da palavra é quanto à "caixa" das letras. Geralmente, a maioria das letras de um texto é minúscula. Desde que "ortografia" é normalmente

considerada uma palavra idêntica a "Ortografia", grande parte dos verificadores ortográficos mapeia todas as letras do texto para maiúsculas ou para minúsculas, antes de submetê-lo à verificação.

No entanto, o problema é um pouco mais complexo, já que existem palavras que são iniciadas obrigatoriamente por maiúsculas (nomes próprios) ou que usam letras estritamente maiúsculas (siglas). As palavras "ibm" e "fortran", por exemplo, deveriam ser consideradas incorretas? O projetista do verificador ou decide em adotar um controle sobre o uso correto de caixa das letras, ou simplesmente faz com que palavras deste tipo não sejam submetidas à verificação, partindo do princípio que este tipo de palavras - nomes próprios, siglas, acrossemias - não devem ser obrigatoriamente do conhecimento de um verificador ortográfico [PETE 80].

2.2 Obtenção e criação de um dicionário

A tarefa mais difícil no desenvolvimento de um verificador baseado em dicionário é a obtenção e criação do dicionário. Obter um dicionário siginifica pesquisar em alguma fonte e, posteriormente, armazenar no computador as palavras da língua. Por criação do dicionário subtende-se o processo de estruturação destas palavras de acordo com o método de verificação adotado pelo verificador.

Para dar uma idéia da dificuldade enfrentada quando se obtém uma lista de palavras válidas da língua, [BENT 85] faz uma

comparação entre a preparação de um ensopado de elefante e a construção de um verificador ortográfico. No primeiro caso, o primeiro passo seria caçar um elefante. Já no segundo caso, dever-se-ia obter uma lista de palavras válidas na língua para a qual o verificador é voltado. Segundo Bentley, após algum tempo é possível que se chegue a conclusão que é muito mais fácil preparar um delicioso ensopado de elefante.

São vários os fatores que contribuem para tornar esta tarefa tão árdua. Inicialmente podemos citar a relutância dos fabricantes de dicionários em fornecer uma cópia em qualquer meio possível de ser lido no computador. Mesmo que um fabricante forneça uma cópia ou que se use um exemplar de dicionário para se dar entrada dos seus vocábulos no computador, observa-se que em nenhum dos dois constam várias formas de palavras, tais como plural, diminutivo, aumentativo, etc.

Um grande dicionário conterá ainda palavras arcaicas e obscuras cuja frequência de uso não justifica a inclusão das mesmas no dicionário do VO. Além do mais, a inclusão destas palavras contribui para aumentar a taxa de erro do verficador, conforme veremos na secção 2.3.

Apesar destes problemas, a procura de um dicionário, previamente digitado, não deve ser descartada, pois o mesmo poderá ser usado em um processo comparativo com uma lista de palavras que servirão de entrada para o dicionário do VO, agilizando o processo de obtenção do mesmo. Este método foi usado pelo SPELL do Unix [TURB 81] [McIL 82].

Caso não se disponha de um dicionário já digitado, o processo de obtenção é mais lento e difícil, porém não é impossível. Um dos métodos que poderá ser utilizado é a análise da freqüência de uso das palavras, considerando um grande número de textos onde os assuntos devem ser os mais diversos. Palavras muito usadas deverão estar ortograficamente corretas e, portanto, serão incluídas no dicionário do VO. Tendo o número de entradas deste atingido um certo limite, ele passará a ser usado pelo VO, e as palavras corretas que, no processo de verificação de um texto, possam vir a ser classificadas como incorretas, poderão posteriormente ser incluídas no dicionário. Turba fez uso deste método na implementação de um verificador ortográfico [TURB 81].

Independente do método utilizado na obtenção do dicionáro, a tarefa se mostra lenta e cansativa, onde o processo de triagem se concentra demasiadamente no homem.

Controles para modificação do dicionário do VO são necessários e geram a necessidade de um **Administrador de Dicionário** que tem como função a sua manutenção: adicionar novas palavras e excluir aquelas pouco ou não usadas [PETE 80].

2.3 Abrangência

Como todo algorítmo de reconhecimento de padrão, um verificador ortográfico pode cometer dois tipos de erro: falhar na aceitação de uma palavra ortograficamente correta, ou falhar na rejeição de uma palavra ortograficamente incorreta [PETE 86].

Se o verificador é baseado em um algorítmo de busca, ele deverá manter uma lista de palavras ortograficamente corretas (dicionário) que será pesquisada sempre que uma palavra estiver sendo verificada. Se a palavra é encontrada, assume-se que a mesma está correta; caso contrário, a palavra é classificada como incorreta.

A falha de um verificador na aceitação de uma palavra correta se dá quando a mesma não se encontra no dicionário. Para reduzir a probabilidade deste tipo de erro, geralmente adiciona-se mais palavras no dicionário. Deixando de lado os problemas acarretados por este procedimento - necessidade de mais memória e aumento no tempo de pesquisa, já discutidos no CAPITULO 1 -, observa-se uma maior probabilidade do verificador aceitar palavras ortograficamente incorretas [PETE 86].

Mas como pode este procedimento contribuir para que uma palavra incorreta seja classificada como correta? Segundo Peterson, isto ocorre quando o autor/usuário pretendendo digitar a palavra X, digita, na realidade, a palavra Y. "Caixa" pode ser digitada como "baixa", por exemplo. O erro não é detectado, quando a palavra digitada consta no dicionário. Este tipo de erro poderá ser detectado apenas por algorítmos mais complexos que usem informações sintáticas e semânticas.

Peterson acrescenta ainda que a ocorrência deste tipo de erro cresce à medida que cresce o tamanho do dicionário. Quando aumenta-se a abrangência de um dicionário, tende-se a fazer inclusões de palavras obscuras e de pouco uso, cuja ortografia

pode coincidir com a das palavras digitadas incorretamente.

Ainda quanto à abrangência do dicionário, o projetista terá que decidir se o mesmo deverá englobar nomes próprios e palavras estrangeiras cujo uso seja por demais frequente nos textos da comunidade. E se os termos técnicos, voltados para uma área específica, deverão constar neste dicionário ou se fará uso de dicionários especializados.

2.4 Representação do dicionário

O desempenho de um verificador ortográfico é muito importante, principalmente se ele apresenta processamento "on-line", onde o usuário interage com o verificador durante o processo de verificação.

Segundo Peterson [PETE 80], a estrutura do dicionário tem fundamental importância no desempenho do VO. A determinação de uma estrutura correta, por sua vez, depende basicamente da configuração do sistema de computação no qual se deseja implementar o VO, ou seja, qual a memória principal e secundária disponível no computador, quais os métodos de acesso existentes no sistema, qual a velocidade de acesso a disco ou a disquete, etc.

A estrutura do dicionário deve conter, em um espaço razoável, informações suficientes para que se tenha uma boa qualidade de resposta, e que esta resposta se processe em tempo hábil.

Há um grande interesse em se obter dicionários bastante compactos. Isto se deve, principalmente, ao uso de verificadores ortográficos em máquinas de pequeno porte bastante comuns em escritórios. Uma boa compactação pode ser obtida associando-se o uso do método de Análise de afixos a uma representação de dicionário do tipo "Total Hashing", na qual apenas um bit na tabela hash indica se a palavra consta no dicionário (bit ligado) ou não (bit desligado). Esta associação de método de verificação e forma de representação de dicionário reduz drasticamente a área ocupada por este, sem, no entanto, comprometer o desempenho ou a qualidade de resposta do verificador. Estes fatores dependerão ainda do tamanho da tabela hash e/ou do seu grau de utilização, já que um bit não possibilita a representação de colisões no hash.

Muitos verificadores ortográficos usam vários dicionários, tanto estáticos - não se alteram durante a verificação - quanto dinâmicos - podem receber novas palavras durante o processo de verificação - , com o objetivo de melhorar o desempenho do verificador. Cada dicionário podendo apresentar uma estrutura distinta de acordo com as características e a freqüência de uso do mesmo.

Um VO poderá ainda permitir a criação e a manutenção de dicionários especializados voltados para abrangência de termos técnicos que não são de interesse ou de uso geral. A estrutura destes dicionários deverá ser compatível com o carater dinâmico dos mesmos.

2.5 Análise do uso de palavras

A análise das características de uso das palavras em diversos textos pode influenciar no projeto de um verificador ortográfico. Características tais como a freqüência de uso da palavra, bem como o seu tamanho podem contribuir de forma proveitosa na construção do verificador [TURB 81].

De acordo com as informações obtidas através destas análises, pode-se criar estruturas de dados e algorítmos que melhor se adaptem às propriedades da línguagem utilizada nos textos, e que, consequentemente, levem o verificador a apresentar um melhor desempenho.

Estas análises, no entanto, exigem um número significativo de textos contendo uma variada gama de assuntos, para que se possa estabelecer regras. O trabalho fica bastante simplificado quando pessoas pertencentes a áreas voltadas para o estudo do comportamento da língua, como é o caso da Linguística, já tenham levantado este tipo de informação.

A inexistência deste tipo de informação não impede, no entanto, a construção de um VO. Uma vez implantado, um VO poderá evidenciar, através de um uso exaustivo, as características das palavras nos textos, levando o projetista a se certificar se as estruturas e algorítmos escolhidos, inicialmente, são adequados, e a fazer alterações nos mesmos, caso sejam necessárias e possíveis.

3. ALGUNS VERIFICADORES ORTOGRÁFICOS EXISTENTES

Este capítulo tem como objetivo apresentar as características básicas de alguns verificadores ortográficos para a língua inglesa e portuguesa que se tem conhecimento.

Nem sempre um mesmo parâmetro será apresentado por todos os verificadores devido a inexistência da informação na bibliografia pesquisada. As informações sobre os verificadores para a língua inglesa constam em artigos onde são evidenciados os problemas enfrentados durante a fase de projeto, e as soluções encontradas pelos projetistas. As informações sobre o verificador para a língua portuguesa constam em manuais de usuário onde se dá uma maior ênfase à utilização do verificador, e não à sua construção.

3.1 Verificadores para a língua inglesa

3.1.1 **SPELL** (DEC-10) [PETE 80]

Criado em 1971 por Ralph Gorin, o SPELL usa o método de verificação análise de afixos com controle sobre a formação de palavras. Tem processamento "on-line" e apresenta seu dicionário em uma tabela hash de 6760 entradas cujo acesso é determinado pela função hash:

$$h(P) = (L1 * 26 + L2) * 10 + \min(WL-2 , 9)$$

onde **L1** e **L2** = dois primeiros caracteres da palavra **P**, respectivamente e **WL** = tamanho da palavra **P**.

3.1.2 **TYPO** (Unix) [PETE 80]

Este verificador usa o método de análise da frequência de digramas e trigramas nas palavras do texto e usa uma lista de 2500 palavras mais comuns da língua para, através de um processo comparativo, reduzir o tamanho da lista apresentada ao usuário. Apresenta um processamento "batch".

3.1.3 **SPELL** (Unix) [PETE 80] [BENT 85] [McIL 82]

Apresentamos maiores detalhes sobre este verificador por dois motivos: primeiro pela existência de maiores informações na bibliografia pesquisada, e segundo, pela engenhosidade que envolveu o seu projeto.

O SPELL foi projetado por McIlroy em 1978. McIlroy iniciamente dedicou-se à tarefa de criação do dicionário, incluindo vocábulos selecionados de dicionários existentes, os nomes próprios mais comuns da lista telefônica, nomes famosos, nome mitológicos, nomes de grandes companhias, nomes geográficos, de animais e de palavras. O resultado foi uma lista de 75000 palavras.

McIlroy optou pelo método de análise de afixos por achá-lo necessário e conveniente. Necessário por não haver nenhuma lista com total abrangência sobre a língua inglesa, e conveniente pela redução do tamanho de dicionário de 75000 para 30000 palavras. Uma vez que a retirada de afixos poderia destruir a ordem alfabética do texto, McIlroy optou por um acesso randômico ao dicionário.

Para que o SPELL apresentasse um bom desempenho, era necessário manter o dicionário na memória principal. Mas, McIlroy só dispunha de uma memória com 64 Kb de espaço de endereçamento. A saída foi associar o método de análise de afixos à representação "Total Hashing" para o dicionário. Na pesquisa de uma palavra, seria acessado o H(P)-ésimo bit da tabela, classificando a palavra como correta, se o mesmo estiver ligado. H(P) representa a função hash a qual a palavra P é submetida para geração de uma entrada da tabela. Uma palavra incorreta poderá levar a um bit ligado. No entanto, a probabilidade deste fato ocorrer é tão baixa que McIlroy considerou o fato insignificante. A falha é de um erro para cada 4000 palavras verificadas.

Para reduzir ainda mais o espaço utilizado pelo dicionário, McIlroy ordenou a lista de palavras e passou a representar apenas as diferenças entre os sucessivos valores do hash, ou seja, a diferença entre as entradas da tabela cujo bit estaria ligado. Ficou estabelecido que o valor inical seria igual a zero. Gasta-se, em média, 13,6 bits para representar uma diferença. Como resultado, McIlroy obteve um dicionário que usa 64 kb de memória principal.

McIlroy ainda criou a "Lista de Exceções", onde seriam incluídas as palavras incorretas constituídas de radical e afixos corretos, resultantes, na maioria das vezes, de um erro no processo de digitação. As palavras que constam nesta lista não serão reconhecidas como corretas pelo SPELL, mesmo que seus afixos e radical constem nos dicionários específicos.

Com uma única e simples estrutura de representação de dicionário, McIlroy conseguiu criar um VO de excelente desempenho cujas características dificilmente são encontradas em um mesmo verificador ortográfico: baixa taxa de erro, uso de pouco espaço de memória e um tempo de resposta bastante satisfatório.

3.2 Verificador ortográfico para a língua portuguesa

O Best Spell [WILD 86] foi criado pela software house Wild West. Este verificador usa o método de análise de afixos e apresenta processamento "batch + semi-interativo".

A software house alega que seu dicionário tem uma abrangência de, mais ou menos, 100.000 palavras, e é representado em uma tabela hash, permitindo a verificação de 6000 palavras por minuto em microcomputadores compatíveis com o IBM PC americano de 4,77 MHz, com uma taxa de erro de uma falha para cada 100.000 palavras pesquisadas. Permite ainda a criação e manutenção de dicionários especializados.

Algumas características que a software house atribui ao Best Spell nos levam a crer que este verificador utiliza "Total Hashing" para representação do dicionário padrão. Segundo o manual de usuário, a inclusão de uma nova palavra no dicionário padrão não aumenta o tamanho deste. No entanto, ocorre um pequeno aumento na probabilidade de uma palavra incorreta não ser detectada. Além disso, tem-se a alta velocidade de verificação e a pequena área utilizada pelo dicionário quando arquivado em

disco - 50 Kb. A inclusão de mais palavras significa mais bits ligados. Não ocorrendo aumento na tabela hash, a probabilidade de erro é maior. O dicionário armazenado em disco, é, na realidade, uma tabela de bits.

4. ORGANIZAÇÃO DE UM VERIFICADOR ORTOGRÁFICO PARA A LINGUA PORTUGUESA (VOLP)

A língua portuguesa tem características que justificam perfeitamente a utilização de um verificador ortográfico:

- português não é uma língua totalmente fonética. Nem sempre existe uma correspondência direta entre o som e a ortografia das palavras.

- muitos prefixos e sufixos da língua portuguesa servem aos mesmos propósitos e apresentam pequenas variações ortográficas, como por exemplo: em-, en-, in-, (movimento para dentro), im-, in-, i- (sentido contrário, negação), ãos-, ões-, ães- (plural de palavras terminadas em ão).

- a língua portuguesa possui inúmeras palavras que se constituem, de uma forma ou de outra, exceção a algum tipo de regra.

- a conjugação de verbos na língua portuguesa é bastante complexa. São inúmeros os verbos que se conjugam de forma totalmente irregular.

Todas estas características levam a frequentes erros de ortografia, erros decorrentes da ignorância do autor. Se a estes erros somam-se aqueles provenientes do processo de digitação de textos, atinge-se um nível de erro tal, que o auxílio de um verificador ortográfico torna-se de grande importância.

O projeto e a implantação de um verificador ortográfico envolvem problemas nem sempre triviais, conforme vimos no CAPITULO 2. E quando este verificador é voltado para uma língua com características iguais a estas apresentadas anteriormente, o esforço do projetista, na busca de uma solução para os mesmos, torna-se bem maior. As seções seguintes são dedicadas à apresentação das nossas soluções para os problemas comuns ao projeto de um VO.

4.1 Método de verificação ortográfica do VOLP

Tínhamos um princípio fundamental: tentar manter a simplicidade o máximo possível, exceto quando a mesma viesse comprometer a eficiência ou o desempenho do VOLP. Era prioritária a criação de estruturas simples que levassem, posteriormente, a algorítmos igualmente simples, facilitando a implantação e a manutenção do verificador.

Baseados no fato que o uso de dicionário na verificação ortográfica representa um nível de sofisticação maior que a utilização de análises estatísticas, gerando, na maioria das vezes melhores resultados [PETE 80], e considerando que o VOLP seria voltado para máquinas de pequeno porte, achamos mais adequado o uso de dicionários e do método de análise de afixos. Para evitar que palavras incorretas, constituídas de radical e afixos corretos, fossem classificadas como corretas, resolvemos estabelecer controle sobre o uso adequado de radicais e afixos através de regras.

Uma regra representa um conjunto de sufixos e, uma vez associada a um radical, estabelece que os mesmos podem se unir formando uma palavra ortograficamente correta. Maiores detalhes sobre o uso de regras serão apresentados no CAPITULO 5, quando especificarmos a estrutura do dicionário.

Nem todos os sufixos que podem se ligar a um radical para formação de uma palavra válida continuam tendo esta característica se a este radical se acrescentar um ou mais prefixos. Como exemplo apresentamos o radical **pobr**. O sufixo **-ezinho** pode se ligar a este radical formando uma palavra ortograficamente correta. No entanto, se associarmos a este radical o prefixo **em-**, este sufixo não mais poderá se unir ao radical **pobr**. O uso do sufixo **-ecer** só então é permitido junto a este radical, devido a associação daquele prefixo.

O uso de regras que determinassem a formação de palavras com relação tanto aos prefixos quanto aos sufixos, levaria a uma estrutura de dicionário razoavelmente complexa que, além de requerer um maior espaço de armazenamento, contribuiria para tornar o algorítmo de análise de afixos mais lento e complexo.

A solução encontrada foi optar-se pela análise apenas de sufixos, mesmo sabendo-se que esta decisão acarretaria em um aumento no tamanho do dicionário. Só o sufixo de uma palavra será extraído, não se fazendo nenhuma distinção entre o prefixo e o radical. O que na língua portuguesa considera-se prefixo mais radical, para o VOLP será considerado apenas um radical. Isto é válido tanto para as palavras do texto quanto para as palavras do

dicionário.

Foi com base no fato de não termos nenhum estudo estatístico sobre as propriedades de uso da língua portuguesa, que decidimos projetar um verificador mais simples. Posteriormente, com o uso do VOLP, as características se evidenciarão, dando maiores condições de se construir verificadores mais complexos (i.e. verificadores com processamento "on-line").

Uma das diferenças básicas entre processamento "batch" e "on-line" é que no primeiro, o dicionário ordenado é varrido em um único passo, no qual faz-se a comparação com as palavras ordenadas do texto. No segundo tipo de processamento, o dicionário é acessado randomicamente a cada ocorrência da palavra no texto. O tempo de resposta, neste caso, é muito mais crítico.

Mesmo com processamento essencialmente "batch", seria mais adequado que o VOLP fizesse acessos randômicos ao dicionário, considerando o tamanho deste dicionário e a possibilidade do uso de mais de um dicionário. Um processamento "batch" se adapta melhor quando a relação entre o tamanho do texto em verificação e o tamanho do dicionário é alta. Quanto mais alta for, melhor. Considerando que português é uma língua de muitas palavras e que os prefixos constarão no dicionário do VOLP, aumentando ainda mais o número de entradas, acesso sequencial não seria a melhor escolha. O VOLP usa, portanto, o processamento "batch + semi-interativo" com acesso randômico ao dicionário.

4.2 Interface com o usuário do VOLP

Percorrer uma lista de palavras desconhecidas, fornecida por um verificador ortográfico, verificando se cada palavra está realmente incorreta ou se apenas não consta no dicionário, nem sempre é uma tarefa muito agradável. E quando pouquíssimas palavras incorretas se distribuem entre inúmeras palavras corretas (desconhecidas pelo VO) nesta lista, a tendência é o fracasso do usuário na pesquisa das mesmas.

Observamos que tomar conhecimento de cada palavra da lista através de um processo interativo (verificador/usuário), faz com que o usuário se sinta mais motivado, e que haja maior sucesso na detecção de palavras incorretas. Este fato foi observado quando testávamos o verificador ortográfico Best Spell, descrito brevemente no CAPITULO 3. Este VO deixa nas mãos do usuário a decisão de ter uma lista apresentada interativamente ou de forma convencional. O VOLP tem, portanto, um processamento "batch + semi-interativo". Caso o usuário opte pela apresentação das palavras incorretas interativamente, o VOLP deve apresentar uma palavra por vez, permitindo que o usuário classifique a palavra como incorreta ou correta, ou que ainda não tome nenhuma decisão quanta àquela palavra (ignorar a palavra).

De acordo com as características acima determinadas, o VOLP apresenta a seguinte estrutura global:

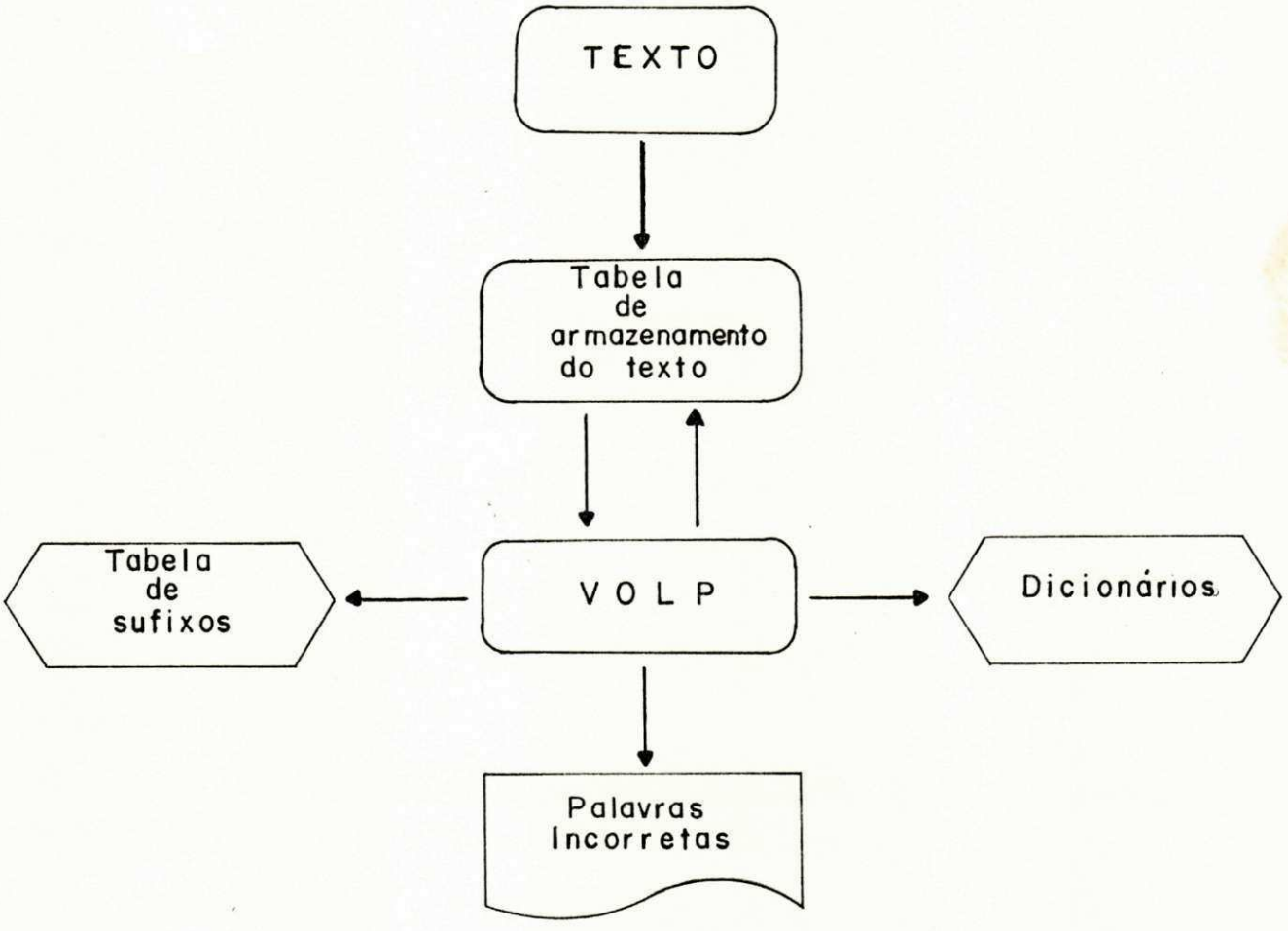


Figura 4.1 - Estrutura global do VOLP

4.3 Formação e reconhecimento de palavras

O VOLP considera caracteres formadores de palavras todas as letras do alfabeto (maiúsculo e minúsculo). Os demais caracteres

- branco, dígitos(0...9) e caracteres especiais (ex. ?, !, $) - são delimitadores de palavras, com exceção do hífen usado pelo usuário no final da linha separando uma palavra em duas. Neste caso o VOLP deverá desconsiderá-lo, reconstituindo a palavra.

Palavras compostas e pronomes ligados a verbos através de hífen serão verificados isoladamente. Considerar a palavra como um todo implicaria na inclusão das diversas combinações no dicionário - um número bastante significativo, considerando a quantidade de verbos existentes e todas as combinações possíveis entre os pronomes e os verbos.

Assim como o hífen, os demais caracteres especiais são classificados como delimitadores, já que os mesmos não são usados nas palavras da língua portuguesa.

Uma vez que o uso conjunto de dígitos e letras não é comum nas palavras da língua portuguesa, e que valores numéricos - conjunto de um ou mais dígitos - fogem do escopo de uma verificação ortográfica, decidimos tratar os dígitos como delimitadores.

Segundo Peterson [PETE 80], uma das funções de um verificador de consistência é verificar o uso apropriado das letras maiúsculas nos nomes próprios encontrados no texto. No entanto, decidimos fazer este controle já a nível de verificação ortográfica.

Palavras como, por exemplo, josé, pmdb e brasil se encontram ortograficamente corretas. No entanto, considerando o uso

apropriado de caixa das letras, estas palavras estão incorretas, sendo sua forma correta a seguinte: José, PMDB e Brasil. Achamos que erros deste tipo não deveriam passar desapercebidos do VOLP. O nosso conceito de verificação ortográfica amplia-se, então. Além de verificar a ortografia da palavra, o VOLP observa se a mesma apresenta um uso apropriado de caixa das letras.

Consideramos 3 categorias de uso de uma palavra: totalmente minúscula, iniciada por maiúscula e totalmente maiúscula. O uso de mais de uma letra maiúscula, e letras minúsculas numa mesma palavra não será considerada pelo VOLP. Palavras com esta característica não serão submetidas à verificação.

As regras de uso apropriado de caixa das letras são as seguintes:

- Uma palavra totalmente minúscula no dicionário casará com qualquer uma das 3 categorias de uso da palavra no texto.

- Uma palavra iniciada por maiúscula no dicionário casará apenas com a palavra iniciada por maiúscula ou totalmente maiúscula no texto.

- Uma palavra totalmente maiúscula no dicionário casará apenas com esta mesma categoria de uso da palavra no texto.

O único caso em que é impossível fazer este tipo de controle é quando uma palavra pertence ao mesmo tempo á classe de palavras

que fazem uso obrigatório de letras maiúsculas - seja apenas a inicial ou todas as letras -, e à classe de palavras que tanto podem usar letras minúsculas ou maiúsculas. Como exemplo apresentamos a palavra **cobra**. Se a palavra se refere ao animal, ela poderá ser escrita de qualquer forma: cobra, Cobra ou COBRA. No entanto, se ela se refere ao fabricante nacional de computadores, deveria ser escrita em uma das seguintes formas: Cobra ou COBRA. O usuário sabe qual o sentido da palavra "cobra" no seu texto. O mesmo não ocorre com o verificador ortográfico. Este grau de conhecimento só é possível em uma verificação a nível semântico. Portanto, para este tipo de palavra o VOLP torna-se incapaz de exercer algum controle. O uso incorreto da palavra "cobra", se referindo ao fabricante de computadores, não será detectado pelo verificador.

A maioria dos processadores de textos usa o ponto (.) para indicar o início de uma linha de comando. As linhas de um texto que apresentarem esta característica não deverão ser submetidas à verificação pelo VOLP.

4.4 Dicionário

Apesar de pertencer logicamente a este capítulo, a discussão sobre a solução dada aos problemas relacionados com a criação e a obtenção do dicionário do VOLP consta no CAPITULO 7. Uma vez conhecida a estrutura utilizada pelo dicionário, descrita no CAPITULO 5, torna-se mais simples tanto a apresentação quanto a compreensão da solução encontrada.

5. ESTRUTURAS DE DADOS DO VOLP

Neste capítulo especificaremos todas as estruturas de dados utilizadas pelo VOLP, discutindo ainda, as razões pelas quais elas apresentam tais características.

5.1 Estrutura de armazenamento do texto na memória

Um verificador ortográfico com processamento "batch" ou "batch + semi-interativo" ordena as palavras do texto a ser verificado visando eliminar as duplicações de palavras. Desta forma uma palavra é pesquisada apenas uma única vez no dicionário.

Em vez de ordenar todo o texto, o verificador poderá pesquisar cada palavra do texto em uma tabela interna. Caso ela não seja encontrada, o VOLP deverá incluí-la.

Escolhemos uma estrutura mista - mistura de **Trie** [HORO 76] com **árvore de pesquisa binária** - que chamamos de **trie/árvore.** O uso isolado da estrutura trie (figura 5.1) foi descartado, devido à enorme área de memória exigida pelos nodos de desvio.

A trie da estrutura trie/árvore (figura 5.2) possui dois níveis de nodos de desvio. O caminhamento é determinado pelos dois primeiros caracteres da palavra pesquisada. O caminhamento, portanto, obedece a seguinte função:

SAMPLE(P,i) = i-ésimo caractere da palavra P

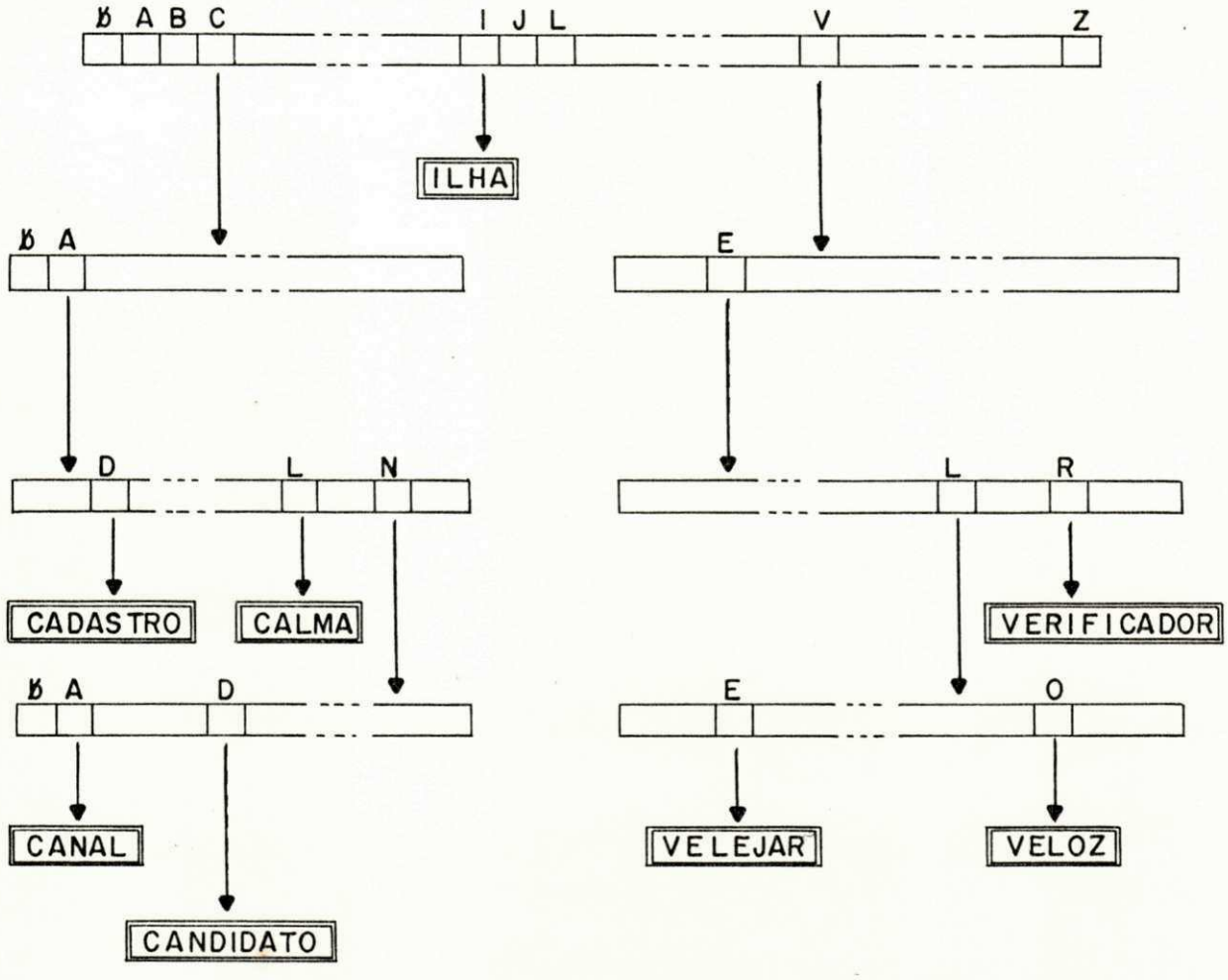


Figura 5.1 - Exemplo de uma estrutura do tipo trie

Cada nodo da "trie" possui 40 entradas, considerando as 26 letras do alfabeto, o branco, o cedilha e as seguintes vogais acentuadas:

á à ã ä
é ê
í
ó õ ö
u ú

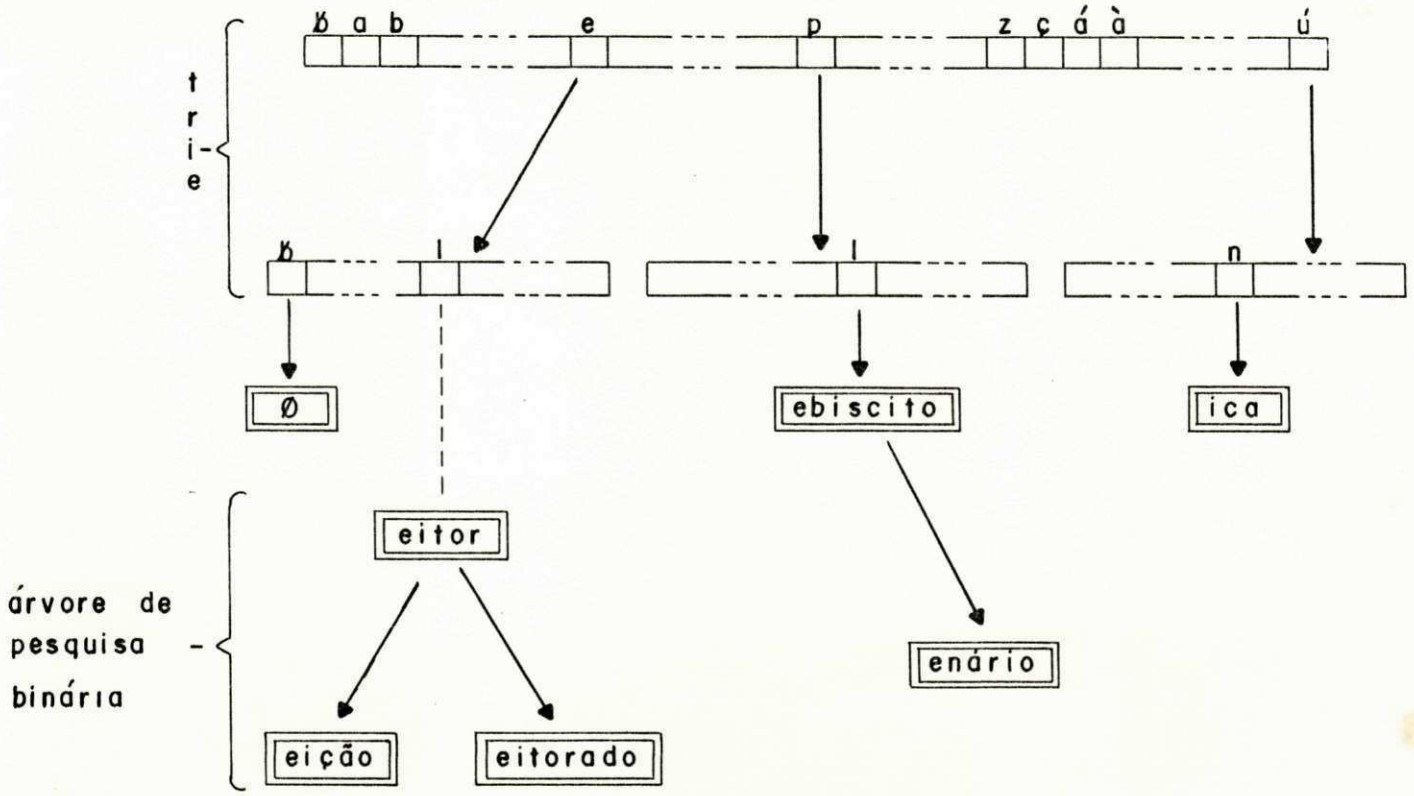


Figura 5.2 - Estrutura de armazenamento do texto: trie/árvore

O nodo de desvio do primeiro nível será criado tão logo seja necessário armazenar a primeira palavra. Cada nodo do segundo nível será criado quando for preciso armazenar uma palavra iniciada por um caractere nunca antes encontrada no texto como caractere inicial de palavra. Haverá tantos nodos no segundo nível da trie quantas forem os caracteres distintos utilizados como inicial de palavras do texto.

O segundo nível da trie aponta para o segundo componente da trie/árvore: uma árvore de pesquisa binária. Cada árvore armazena um conjunto de palavras cujos dois primeiros caracteres são

idênticos para todas. Como estes dois caracteres são comuns, e podem ser gerados a partir do caminhamento na trie, é desnecessário armazená-los nos nodos da árvore. Uma palavra com 7 caracteres, por exemplo, terá apenas os seus últimos 5 caracteres armazenados. Com isto, reduzimos o espaço utilizado para armazenamento das árvores de pesquisa binária, e tornamos o processo de pesquisa mais eficiente.

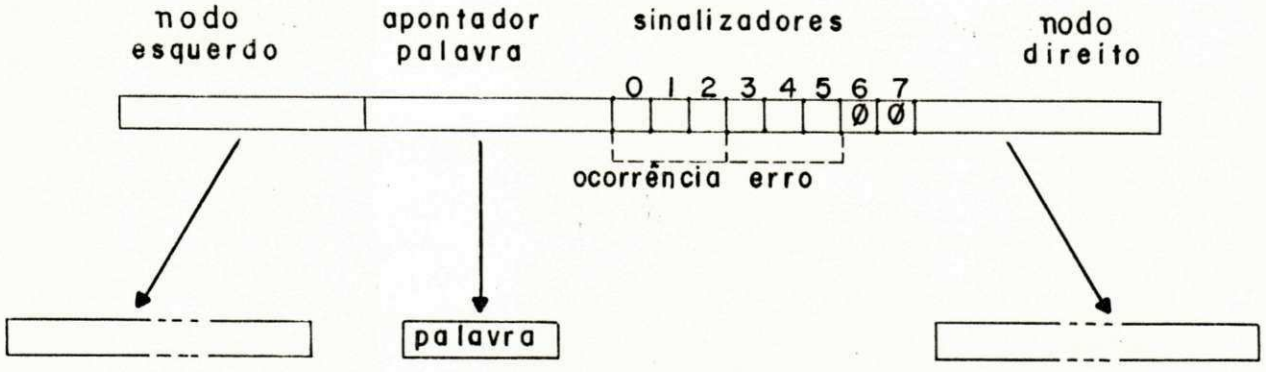


Figura 5.3 - Estrutura de nodo da árvore de pesquisa binária

A figura 5.3 apresenta a estrutura da cada nodo da árvore de pesquisa binária. Os apontadores "apont. nodo esquerdo" e "apont. nodo direito" apontam para as sub-árvores esquerda e direita, respectivamente.

O campo "apontador palavra" aponta para a palavra correspondente àquele nodo. O uso de um apontador para a palavra acarreta uma redução de espaço. Caso contrário, este campo deveria ter um tamanho capaz de armazenar a maior palavra

possível de constar em um texto. Mesmo assim este campo poderia ainda ser insuficiente para um palavra que, na realidade, constitui-se de duas palavras sem um caractere delimitador entre elas (erro de digitação). Haveria, portanto, um grande desperdício de área de memória. Mesmo que o apontador requeira uma área adicional, ele permite que cada palavra utilize apenas o espaço necessário ao seu armazenamento.

O sub-campo "ocor." serve para registrar as categorias de uso da palavra no texto. Isto se faz necessário devido ao fato da palavra ser armazenada apenas uma única vez nesta estrutura, tendo antes sido totalmente mapeada para minúscula, independente do número de ocorrências e das categorias de uso da palavra no texto. Neste campo as informações são registradas a nível de bits. Cada bit ligado indica a ocorrência da categoria correspondente no texto, devendo o VOLP fazer as devidas transformações na palavra antes de submetê-la à verificação. Como o VOLP considerará três categorias de uso, utilizamos três bits, um para cada categoria:

bit 0 - palavra totalmente minúscula

bit 1 - palavra iniciada por maiúscula

bit 2 - palavra totalmente maiúscula

A ordem de uso de cada bit não foi escolhida de forma arbitrária. De acordo com a ordem estabelecida acima, o VOLP ao encontrar uma determinada categoria de uso no dicionário, deverá considerar, automaticamente, a(s) categoria(s) anteriore(s), caso existam, incorretas. A(s) categoria(s) de uso superior(es),

será(ão) considerada(s) corretas, automaticamente, não sendo necessária nenhuma pesquisa ao dicionário.

Como podemos observar, uma palavra poderá apresentar incorreção em uma categoria de uso, e em outra, não. O VOLP deveria apresentar ao usuário a palavra com a categoria de uso incorreta. O campo "erro" é usado pelo VOLP para registro da(s) categoria(s) de uso da palavra que são incorretas, da seguinte forma:

bit 3 - palavra totalmente minúscula incorreta
bit 4 - palavra iniciada por maiúscula incorreta
bit 5 - palavra totalmente maiúscula incorreta

Todas as categorias de uso da palavra incorretas devem ser apresentadas pelo VOLP ao usuário, permitindo que este use editores ou processadores de texto que fazem distinção entre letras maiúsculas e minúsculas, na busca destas palavras no texto.

A existência de palavras no texto com um ou dois caracteres implica na ocorrência de nodos da árvore vazios, ou seja, nodos cujo campo reservado para a palavra contêm apenas o caractere utilizado para terminar palavras - '\0'.

5.2 Dicionário principal

Uma das coisas que mais podem afetar o desempenho de um verificador ortográfico é o tempo gasto para acessar um dicionário armazenado em disco, principalmente, quando há

possibilidade de se fazer vários acessos antes do verificador classificar a palavra como correta ou incorreta. O ideal é manter o dicionário na memória principal. Mas na maioria das vezes este espaço é insuficiente para conter o dicionário completo, levando o projetista a criar um dicionário na memória e outro no disco, de acordo com algum critério.

Quando é inevitável a criação do dicionário em disco, a estrutura de representação do mesmo, bem como o seu método de acesso passam a ter um papel ainda mais importante no que diz respeito ao desenvolvimento do verificador. Estes devem minimizar, tanto quanto possível, os acessos a disco na verificação de uma palavra.

O VOLP faz uso de dicionário em memória e em disco. O critério utilizado para inclusão de uma palavra na memória é a freqüência de uso da mesma, com relação às demais, nos diversos textos analisados com este fim (ver capítulo 6). O uso de **regras** estabelecendo controle sobre a formação das palavras é feito em ambos os dicionários, principal (memória principal) e secundário (disco).

Estabelecemos que cada regra associada a um radical, em qualquer um dos dois dicionários, deveria ocupar um byte, possibilitando a criação de 256 regras distintas, aqui representadas por Rn, onde 0<=n<=255.

O uso da regra zero (R0) é especial. Quando associada a uma entrada de um dicionário, esta regra indica que a entrada é uma palavra integral.

Cada entrada do dicionário principal (figura 5.4) contém o radical de um verbo associado a uma regra que abrange os sufixos das formas verbais deste verbo, ou contém uma palavra integral associada a R0.

palavra	regra
entrada 1	R1
entrada 2	R2
entrada n	Rn

Rn : Regra n
$0 \leq n \leq 255$

Figura 5.4 - Estrutura do dicionário principal

Com o uso de regras associadas aos radicais dos verbos se obtém uma boa redução de espaço, já que correspondendo a todas as formas geradas por uma conjugação de um verbo, tem-se apenas uma entrada no dicionário. O uso de regras associadas ao radical de palavras necessitaria de uma maior área reservada ás regras. Com isto teríamos desperdício de memória ao incluirmos as palavras que não usam regras, como as palavras de dois ou três caracteres e as palavras invariáveis, todas frequentemente

usadas. Economia de memória significa mais memória disponível para novas palavras e, consequentemente, menos acesso a disco.

O método de acesso adotado para ambos os dicionários - principal e secundário - é o **hashing**. Este método, segundo Knuth [KNUT 73], permite um melhor desempenho do verificador, caso sejam escolhidos uma boa função de Hash e um bom método de resolução de colisões. Uma boa função de hash h(K) deve preencher dois pré-requisitos: ser computada rapidamente e minimizar colisões.

Knuth afirma que nenhum dos métodos de hashing sugeridos até então provou ser superior ao método da simples divisão ou da multiplicação. Optamos pelo metodo da divisão que consiste em usar como índice da tabela hash o resto de uma divisão.

$$h(K) = K \bmod M$$

De posse de uma palavra candidata à verificação, o VOLP extrai o sufixo, caso ele conste na palavra, e soma os valores dos caracteres do radical ou da palavra integral (palavra sem sufixo) gerando desta forma o valor de **K**.

Alguns valores de M contribuem mais ou menos para ocorrência de colisões na tabela hash. Uma boa sugestão é que M seja igual ao maior número primo menor que o número de entradas da tabela hash. As características que o valor de M deve ter se tornarão mais claras quando especificarmos o método de resolução de colisões.

O método de resolução de colisões criado por Brent [KNUT 73] se adapta perfeitamente ao nosso caso. Brent parte do princípio que as pequisas com sucesso em tabelas hash são muito mais comuns do que a inclusão de um item nesta mesma tabela. Sem falar nos casos em que uma tabela é criada uma única vez, para depois ser acessada exclusivamente para pesquisa. Portanto, o ideal é que se tenha um maior trabalho na inclusão de um item na tabela, rearrumando os itens, com o propósito de reduzir o tempo de pesquisa nesta tabela.

Os dicionários principal e secundário serão utilizados pelo VOLP apenas para consulta, cabendo a responsabilidade de criação dos mesmos ao administrador de dicionário, no ambiente de implantação. O método escolhido então é o de Brent.

O método de Brent altera o processo de inserção do método "double hashing" (rehash) que consiste em aplicar duas funções hash h1(K) e h2(K) para obtenção da entrada da tabela hash.

Para um "double hashing", Knuth sugere como melhor escolha do valor de M aquela em que M e M - 2 são números primos, como, por exemplo, 1021 e 1019.

Para que obtivéssemos maior redução no tempo de pesquisa nos dicionários, levamos ainda em consideração fatores como a densidade da tabela hash e a freqüência de uso das palavras nos textos (ver CAPITULO 7).

5.3 Tabela de Sufixos

Para que o VOLP faça a análise de sufixos é necessário a existência de uma estrutura que contenha os sufixos que podem e devem ser extraídos das palavras em verificação.

Como o VOLP também faz uso de regras que indicam quais sufixos podem se ligar a cada radical formando palavras ortograficamente corretas, achamos conveniente que esta tabela (figura 5.5) contivesse além dos sufixos as regras que os abrangem.

Um verificador reconhece de forma mais direta e mais simples o sufixo de uma palavra lendo-a na sua ordem inversa, ou seja, de trás para a frente. O primeiro caractere de uma palavra a ser considerado (i.e. o último caractere da palavra) pertence certamente ao sufixo, exceto no caso em que a palavra não possui sufixo. Cabe ao verificador apenas determinar qual é o sufixo através de uma processo comparativo.

Na ordem normal - da esquerda para a direita - a dificuldade enfrentada é perceber, de imediato, o verdadeiro início do sufixo da palavra. O processo comparativo pode ter início quando ainda não se atingiu o sufixo da palavra, pelo simples fato do seu radical conter um caractere coincidente com o inicial de um ou vários sufixos da tabela.

A pesquisa na tabela é determinada pelo caractere final do sufixo a ser pesquisado que corresponderá a uma das entradas do arranjo de apontadores para lista de sufixos. Cada lista conterá

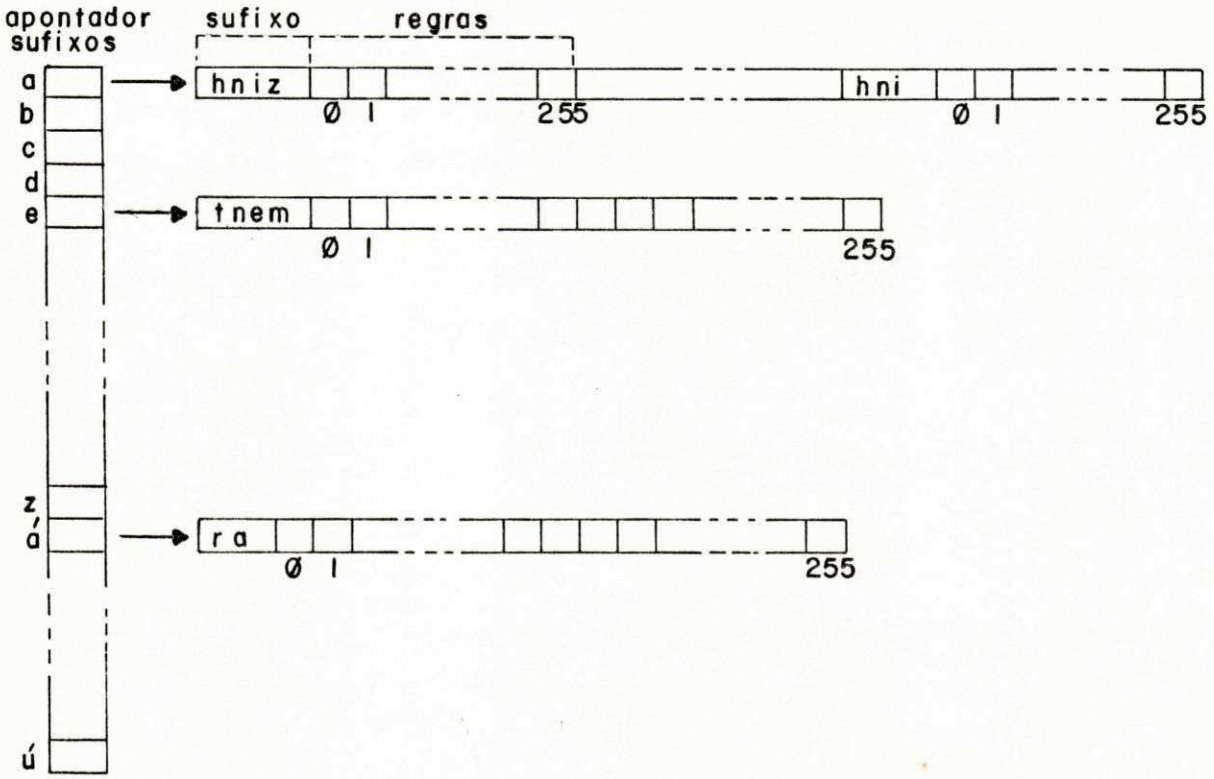


Figura 5.5 - Tabela de sufixos

apenas sufixos terminados com o mesmo caractere. Este processo de triagem, feito quando se inicia a pesquisa de um sufixo na tabela, contribui para a redução do número de comparações e consequentemente para a diminuição no tempo de pesquisa. O fim de cada lista de sufixos é representado por um campo reservado para o sufixo preenchido com o caractere '$'.

O último caractere do sufixo não precisa ser armazenado já que o mesmo pode ser determinado pela posição do apontador no

"array". Assim fazemos a economia de 1 caractere por sufixo da tabela. Quando o sufixo é composto de um caractere apenas, o campo reservado para o seu armazenamento conterá apenas o caractere 'Ø', representando o fim do sufixo.

Este arranjo de apontadores para lista de sufixos permite ainda que, eventualmente, a busca de um determinado sufixo obtenha insucesso sem que haja nenhuma comparação com os sufixos da tabela. Para isto basta que a palavra da qual o VOLP deseja extrair o sufixo tenha como último caractere, um caractere que não é usado como final em nenhum sufixo da tabela. Naturalmente o apontador correspondente a este caractere estará nulo, indicando a inexistência de sufixos e evitando qualquer processo comparativo.

Sufixos que contenham outro, como, por exemplo, INHA que contém o sufixo A, devem vir antes deste para evitar que o VOLP faça uma extração do menor sufixo e, não obtendo sucesso ao pesquisar no dicionário o radical obtido, tenha que fazer uma outra extração, desta vez a do sufixo de maior tamanho. Naturalmente, a criação dos dicionários deve seguir este mesmo critério - radicais obtidos com a extração do maior sufixo contido na palavra.

Para cada sufixo contido nesta tabela existe um campo arranjo de bits associado, especificando as regras que abrange aquele sufixo. De acordo com as várias combinações que podem ser feitas com os inúmeros sufixos existentes, visando um uso mais flexível e mais acertado das regras, um sufixo pode ser abrangido

por muitas regras. O campo ocupa 32 bytes e guarda informação a nível de bit. O array possui 256 entradas e quando uma delas contém o valor '1' significa que a regra correspondente àquela entrada abrange o sufixo. Entradas com valor 'Ø' significa que a regra correspondente não abrange o sufixo.

O VOLP reconhece o sufixo de uma palavra candidata a verificação baseado nesta tabela, extraindo-o em seguida. O radical obtido com a extração é usado para geração da chave **K** que será usada nas funções hash h1(k) e h2(k). Caso a palavra não contenha sufixo válido para o VOLP, toda a palavra será considerada para calcular a chave **K**.

Quando o VOLP obtém sucesso na pesquisa de um radical no dicionário, ele volta a usar esta tabela com o intuito de se certificar de que o sufixo anteriormente extraído pode se unir àquele radical formando uma palavra válida. Caso a regra ou uma das regras associadas ao sufixo na tabela esteja também associada ao radical no dicionário, a palavra é aceita como correta. Há exceção no caso do dicionário secundário, que será especificada na seção 5.4, quando da apresentação da estrutura daquele dicionário. No caso de uma palavra não conter sufixo, ela só será classificada como correta se associada à palavra do dicionário encontrar-se uma regra zero (**R0**).

O fato de associarmos a cada sufixo um campo ocupando 32 bytes, dos quais muitos podem não conter qualquer informação, pode aparentar, de princípio, um grande desperdício de área. A opção de fazer um controle sobre o uso correto dos sufixos

através de regras exige que, de alguma forma, especifiquem os sufixos abrangidos por cada regra.

Uma das opções é deixar que a tabela de sufixos contenha apenas os sufixos e criar uma estrutura representando as regras e os sufixos gerados por cada uma delas. Neste caso teríamos também desperdício de área, não pela inexistência de informação mas, pela repetição de informações - um mesmo sufixo pode ser gerado por inúmeras regras.

Para considerar que o uso de um determinado sufixo estaria correto, seria necessário comparar literalmente o sufixo extraído da palavra em verificação com todos os sufixos gerados pela regra associada ao radical (obtido com a extração do sufixo) no dicionário até que houvesse uma coincidência de sufixos. A situação pioraria muito mais no caso do dicionário secundário que, como veremos na secão 5.4, pode ter várias regras associadas a um mesmo radical. Teríamos pois, um verificador com um desempenho desastroso.

Com o uso da estrutura apresentada nesta secão o controle sobre a legalidade de uma ligação entre um sufixo e um radical torna-se extremamente simples. Basta verificar se a entrada do arranjo de bits correspondente à regra ou a uma das regras (dicionário secundário) associada ao radical no dicionário contém o valor '1' (i.e. se o bit correspondente a regra está ligado).

Considerando que ambas as formas de representar a associação dos sufixos às regras desperdiçam área, mas que apenas aquela embutida na estrutura acima apresentada permite uma verificação

com bom desempenho, decidimos optar pela mesma. O uso de uma maior área no armazenamento de uma estrutura de apoio ao processo de verificação seria recompensado pela qualidade e pelo tempo de resposta.

5.4 Dicionário secundário

O reduzido espaço na memória principal faz com que haja necessidade de dicionário armazenado em disco, abrangendo as palavras não incluídas no dicionário principal.

São quatro as diferenças básicas entre os dicionários secundário (figura 5.6) e principal. A primeira diferença se refere ao número de regras que podem se associar a uma mesma entrada do dicionário. Como o acesso a disco é crítico com relação ao tempo de pesquisa, um maior número de regras associadas a um radical permite que, com um único acesso a disco, se tenha acesso a um número razoável de palavras. Este número de regras não pode ser, no entanto, muito alto, aumentando a probabilidade de grandes perdas de memória quando em uma entrada constar apenas uma palavra integral, ou poucas regras.

A segunda diferença diz respeito à natureza da entrada à qual se associam regras. Qualquer radical, seja de um verbo ou de uma palavra (ou pertencente a ambos), pode constar em uma entrada do dicionário, com regras associadas a ele.

A terceira diferença consiste na adição de um novo campo para cada entrada do dicionário, chamado de sinalizadores de

exceções que tem como função indicar se aquela entrada constitui alguma exceção de uma ou mais regras a ele associadas. Um sinalizador de exceção ligado fará com que o VOLP não considere a palavra correta pelo simples fato da regra correspondente ao sinalizador abranger aquele sufixo. É necessário que antes seja feita uma pesquisa na Lista de Exceções. Só então é possível classificar a palavra como correta - se não constar na lista - ou como incorreta - constando na lista.

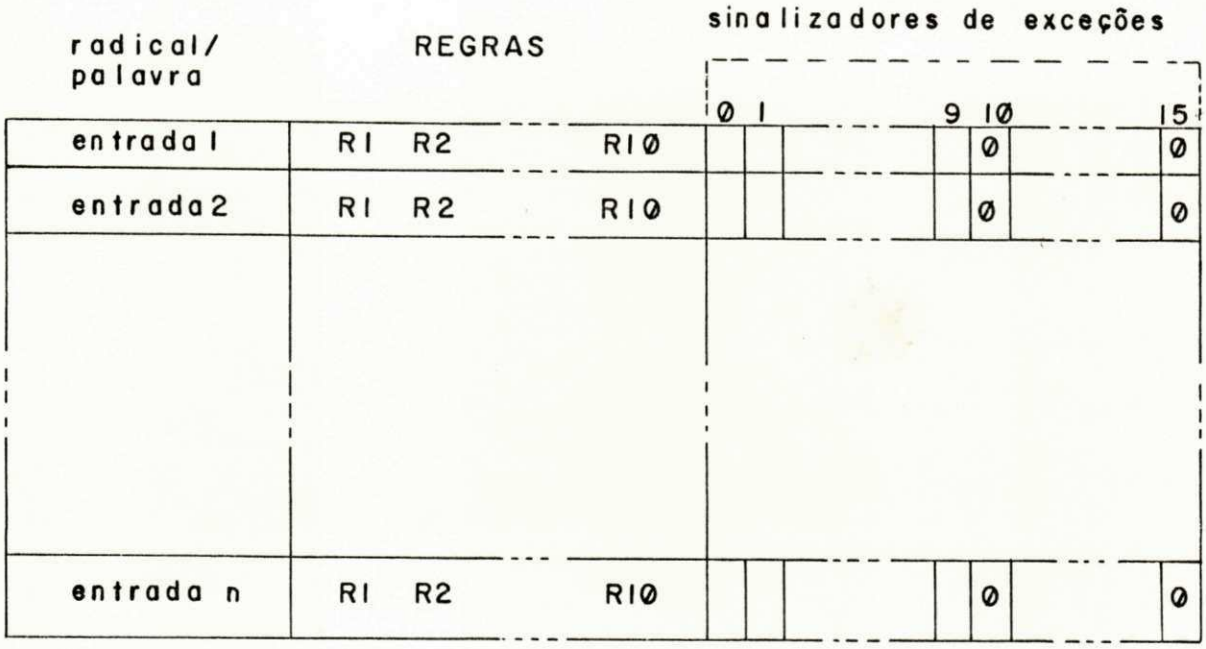


Figura 5.6 - Estrutura do dicionário secundário

Expliquemos o uso da Lista de Exceções. O uso de sinalizadores associados as regras no dicionário possibilita uma maior flexibilidade na criação das regras. Caso uma regra abranja

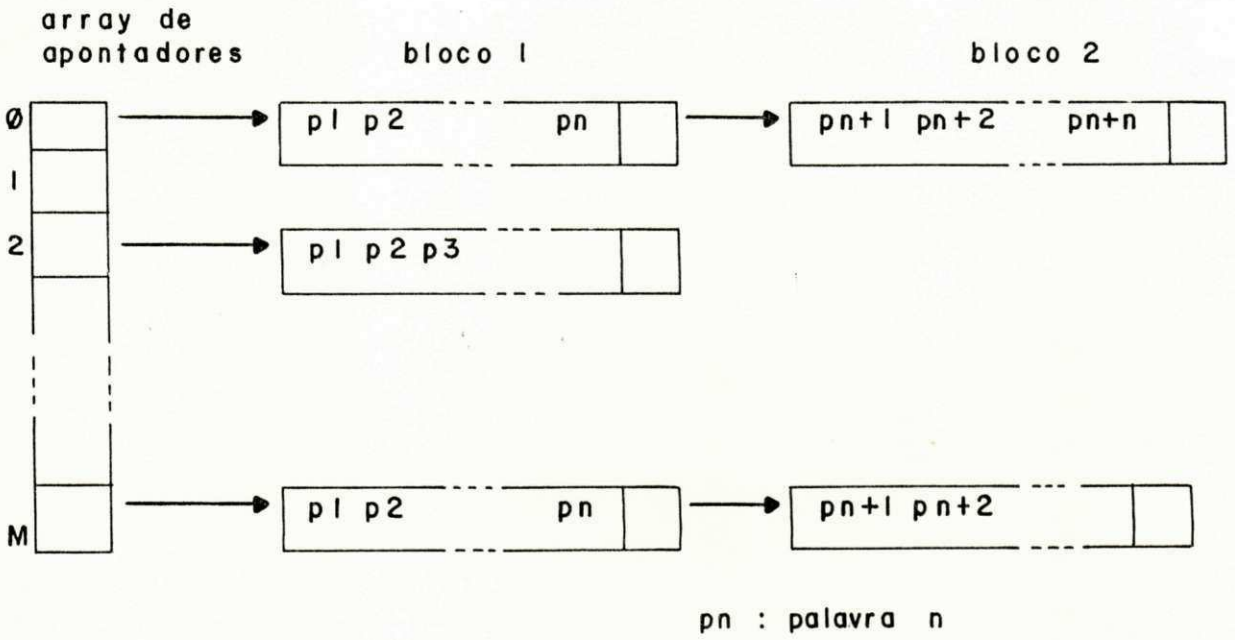
um determinado conjunto de sufixos comumente usados por um certo número de radicais, e haja uma ou outra palavra que use todos os sufixos desta regra, com exceção de um deles, esta regra poderá ser associada ao radical desta palavra no dicionário com o sinalizador posicionalmente correspondente a esta regra ligado. Caso não houvesse um campo reservado às exceções seria necessário a criação de uma outra regra que satisfizesse esta palavra. Teríamos, portanto, duas regras cuja única diferença entre os conjuntos de sufixos abrangidos por elas seria um sufixo.

A quarta e última diferença está relacionada com a tabela hash. A tabela hash de ambos os dicionários é do tipo "closed hashing", ou seja, a entrada ocupada da tabela conterá apenas uma chave (radical ou palavra). Isto difere de uma tabela "open hashing", cuja entrada ocupada contém um apontador para uma lista de chaves que levam, através da função hash, àquela entrada.

Considerando a área utilizada por cada entrada de ambos os dicionários, determinamos que as entradas da tabela hash do dicionário principal armazenaria a própria chave. A tabela hash do dicionário secundário armazenaria um apontador para uma única chave correspondente àquela entrada. A tabela continua sendo "closed hashing", no entanto, fazemos uma grande economia de área considerando que o dicionário será contínuo, e apenas a tabela de apontadores terá entradas desocupadas.

5.5 Lista de exceções

A lista de exceções (figura 5.7) contém todas as palavras que incorretamente poderiam ser aceitas pelo VOLP baseado nas regras que se associam ao radical da palavra no dicionário secundário.



5.7 - Estrutura da lista de exceções e dos dicionários particular(es) e de nomes próprios

A estrutura de representação desta lista, bem como a dos outros dicionários em disco, que não o dicionário secundário, é uma tabela hash do tipo "open hashing". Cada entrada ocupada da tabela contém um endereço físico de uma lista de blocos do disco. Este tipo de estrutura otimiza o acesso a disco [AHO 83], já que, uma vez tendo sido lido um bloco físico do disco, a pesquisa da chave é feita na memória principal.

5.6 Dicionário de nomes próprios

Os nomes próprios não são abrangidos pelo dicionário secundário para maior economia de área de disco, e para maior simplificação do controle de "caixa" das letras.

Se armazenássemos os nomes próprios no dicionário secundário, cada entrada deste dicionário que contivesse um nome próprio, teria toda a área reservada para as regras e sinalizadores de exceções desperdiçada por não conter informação. Considerando que um número bastante grande de nomes próprios poderia ser incluído, teríamos um desperdício de área bastante significativo.

Para que se fizesse um controle sobre o uso apropriado de caixa das letras seria necessária uma informação adicional na estrutura do dicionário secundário que determinasse as categorias de uso da palavra que são ortograficamente corretas ou incorretas.

Nem sempre os textos contêm nomes próprios. E quando os têm, na maioria das vezes, é em número bastante reduzido. Com a exclusão dos nomes próprios do dicionário secundário, reduzimos o seu tamanho consideravelmente, obtendo, por conseguinte, um melhor tempo de resposta, sem, no entanto, eliminar a possibilidade de se fazer uma verificação ortográfica abrangendo os nomes próprios, quando for conveniente.

O VOLP permite o uso exclusivo deste dicionário na verificação de um texto, se assim o usuário desejar. O usuário

poderá ainda incluir novas entradas neste dicionário. No entanto, nenhum controle será feito sobre o uso adequado de "caixa" das letras ou mesmo sobre a correção ortográfica das mesmas. Caberá ao usuário fazer este controle.

5.7 Dicionário particular

O uso de dicionários particulares por um verificador ortográfico torna o processo de verificação mais flexível e eficiente.

A flexibilidade está relacionada com a possibilidade que o usuário tem de criar seu próprio dicionário, seja voltado para uma área específica (ex. medicina) ou mesmo contendo termos de uso geral, não abrangidos pelo dicionário do VOLP. Como o uso exclusivo de dicionários particulares, em um processo de verificação de um texto, é permitido pelo VOLP, o usuário poderá usar seu próprio dicionário, limitando as palavras do texto a um vocabulário restrito.

A eliminação dos termos específicos de uma área de interesse particular do dicionário secundário implica em uma redução do tamanho deste dicionário, tornando o processo de pesquisa de uma palavra mais rápido.

A inclusão de palavras no dicionário particular não sofrerá nenhum controle por parte do VOLP no que diz respeito a correção ortográfica ou uso adequado de caixa das letras das mesmas.

6. ALGORITMO GERAL DO VOLP

Neste capítulo apresentamos o algorítmo de funcionamento do VOLP, a nível de módulos.

Antes de especificarmos o algorítmo propriamente dito, faremos algumas obsevações sobre aspectos a serem considerados pelo mesmo.

O VOLP poderá ser usado com uma das duas finalidades: fazer verificação ortográfica de um texto, ou incluir uma lista de palavras no dicionário - de nomes próprios ou particular. A finalidade do uso do VOLP poderá ser determinada pelo usuário. O "default" é a verificação ortográfica.

Um outro parâmetro a ser estabelecido pelo usuário é o uso de dicionários. Em um processamento normal, o VOLP usa apenas os dicionários principal e secundário, nesta ordem. De acordo com a opção do usuário, ele poderá acessar, além destes dois dicionários, o dicionário de nomes próprios e/ou dicionário(s) particular(es). A ordem de uso destes dicionários, com exceção do principal e do secundário, dependerá da opção do usuário. O "default" é primeiro acessar o(s) dicionário(s) particular(es), na ordem em que foram especificados pelo usuário, e, em seguida, pesquisar o dicionário de nomes próprios.

De acordo com o dicionário que está sendo pesquisado pelo VOLP, são consideradas as seguintes categorias de uso:

- a pesquisa nos dicionários principal e secundário considera apenas a categoria de uso totalmente

minúscula, caso ela tenha ocorrido.

- a pesquisa no dicionário de nomes próprios considera as categorias de uso iniciada por maiúscula e totalmente maiúscula, caso tenham ocorrido.

- a pesquisa em um dicionário particular considera qualquer uma das três categorias de uso que tenha ocorrido no texto.

A classificação de uma palavra como incorreta dependerá do dicionário que se está pesquisando, no momento, e de quais outros dicionários estão sendo usados na pesquisa. Considerando a ocorrência de cada uma das categorias de uso, estabelecemos as seguintes regras:

1) uma palavra totalmente minúscula será considerada incorreta quando:

   . esta categoria de uso não consta no dicionário principal nem no dicionário secundário, e o usuário não solicitou o uso de outros dicionários. Neste caso as demais categorias serão consideradas incorretas.

   . esta categoria não consta nos dicionários principal e secundário nem no(s) dicionário(s) particular(es). Neste caso o usuário solicitou o uso deste(s) dicionário(s). As demais categorias devem ser pesquisadas neste(s) dicionário(s).

2) uma palavra iniciada por maiúscula será considerada incorreta quando:

- esta categoria de uso não consta em nenhum dos dicionários solicitados pelo usuário, seja particular ou de nomes próprios. A categoria de uso totalmente maiúscula, caso tenha ocorrido, deve ser pesquisada.

3) uma palavra totalmente maiúscula será considerada incorreta quando:

- esta categoria de uso não consta em nenhum dos dicionários solicitados pelo usuário, seja particular ou de nomes próprios.

O usuário poderá optar por uma apresentação interativa das palavras incorretas. O mesmo poderá ainda optar por um uso exclusivo do dicionário de nomes próprios e/ou dicionário(s) particular(es). Caso o uso destes dicionários se faça conjuntamente com os dicionários principal e secundário, estes serão pesquisados primeiro, e nesta ordem.

Para maior facilidade de especificação do algoritmo do VOLP, especificamos os módulos que compreendem a pesquisa de uma palavra no dicionário principal e no dicionário secundário à parte.

6.1 Pesquisa no dicionário principal

P1. SE a palavra possui sufixo

P1.1 extrair sufixo da palavra

P1.2 pesquisar radical no dicionário principal

P1.2.1 SE encontrar radical no dicionário principal

P1.2.1.1 SE a regra associada ao radical gerar o sufixo, SUCESSO

P2. pesquisar a palavra integral no dicionário principal

P3. SE encontrar a palavra no dicionário principal E RO estiver associada à palavra, SUCESSO

6.2 Pesquisa no dicionário secundário

S1. SE a palavra possui radical

S1.1 extrair sufixo

S1.2 pesquisar radical no dicionário secundário

S1.3 SE encontrar radical no dicionário secundário

S1.3.1 SE uma das regras associadas ao radical gerar o sufixo

S1.3.1.1 SE o "bit" de excegão da regra está ligado

S1.3.1.1.1 SE palavra consta na Lista de excegão, INSUCESSO

S1.3.1.1.2 SENÃO, SUCESSO

S1.3.1.2 SENÃO, SUCESSO

S2. pesquisar a palavra integral no dicionário secundário

S3. SE encontrar palavra no dicionário E RO estiver associada a palavra, SUCESSO.

6.3 Algorítmo geral do VOLP

M1. SE a finalidade de uso do VOLP é a inclusão de uma lista de palavras em um dicionário

M1.1 fazer inclusão de todas as palavras no dicionário especificado pelo usuário

M1.2 encerrar processamento do VOLP

M2. Estabelecer parâmetros de uso do dicionário, da ordem de pesquisa, e da forma de apresentação das palavras incorretas

M3. Mapear palavras do texto para minúsculas, fazer inclusão na trie/árvore e ligar bit correspondente a categoria de uso (apenas se usar outros dicionários diferentes do principal e secundário)

M4. SE o usuário não fez opção pelo uso exclusivo do dicionário de nomes próprios e/ou dicionários particulares

M4.1 SE pesquisa da palavra no dicionário principal obtem SUCESSO, descartar a palavra

M4.2 SE pesquisa da palavra no dicionário secundário obtém SUCESSO, descartar palavra

M4.3 SE a opção do usuário é o uso exclusivo do dicionário padrão do VOLP

M4.3.1 Ligar bits de erro cujo bit de ocorrência correspondente esteja ligado E VÁ para M4.

M5. SE o usuário solicita uso do dicionário de nomes próprios e/ou dicionário(s) particular(es)

M5.1 Mapear a palavra para maiúscula de acordo com a categoria de uso registrada no campo de ocorrência (um mapeamento por vez)

M5.2 Pesquisar a palavra mapeada no dicionário específico

M5.3 SE a palavra consta no dicionário, descartá-la

M5.4 SE a palavra não consta no dicionário E aquele dicionário é o único ou o último a ser pesquisado

M5.4.1 ligar bit de erro correspondente a categoria de uso

M6. Para cada entrada da trie/arvore que tenha bit de erro ligado

M6.1 Fazer o devido mapeamento para maiúsculo de acordo com a categoria de uso incorreta

M6.2 SE o usuário solicitou apresentação das palavras incorretas de forma interativa

M6.2.1 Apresentar palavra na tela solicitando a classificação do usuário

M6.2.2 SE o usuário classificar a palavra como correta, desligar bit de erro correspondente àquela categoria de uso

M7. Listar cada mapeiamento da palavra cujo bit de erro correspondente esteja ligado

7. OBTENÇÃO E CRIAÇÃO DO DICIONÁRIO DO VOLP

A obtenção e criação do dicionário constitui uma tarefa cujo êxito requer um grande empenho por parte dos responsáveis pela construção de um verificador ortográfico. Quando se trata do dicionário do VOLP, não existe diferença, conforme se constatará no decorrer deste capítulo.

De um modo geral, o processo utilizado pelo VOLP consiste em obter gradativamente palavras da língua portuguesa, a partir de textos, e armazená-las no dicionário de acordo com o método de análise de sufixos e o controle sobre a formação de palavras. Todo este processo é feito de forma automática buscando uma maior rapidez e uma menor probabilidade da existência de inconsistências. A única exceção se dá com relação aos nomes dos verbos. Estes serão extraídos de dicionários de verbos e de gramáticas da língua portuguesa, e o processo de obtenção se dá através da digitação dos mesmos, sob o controle do Administrador do Dicionário.

Entre a fase inicial de obtenção dos vocábulos e a criação do dicionário que será utilizado pelo VOLP, existem várias etapas que constituem três fases do processo: criação da base de dados, criação do dicionário fonte e criação do dicionário objeto (Figura 7.1). Inicialmente, definiremos cada um destes componentes para, em seguida, falarmos sobre o processo de criação de cada um deles.

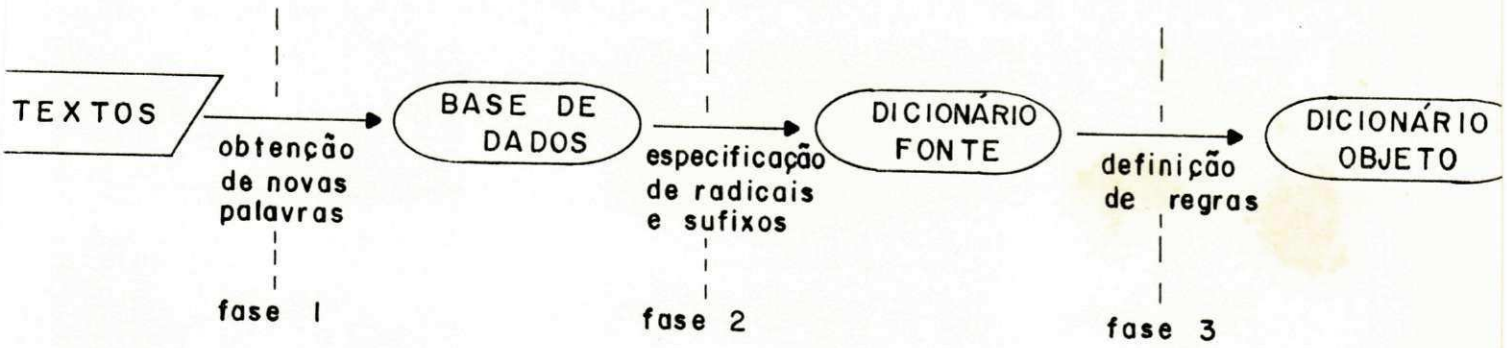


Figura 7.1 - Obtenção e Criação de Dicionários

Base de dados: compreende os dicionários obtidos através de textos (palavras) ou de dicionários e gramáticas da língua portuguesa (verbos). Cada entrada encontra-se na sua forma literal, ou seja, com todos os caracteres com os quais ela é grafada corretamente. Estes dicionários tem como objetivo armazenar todas as palavras que deverão constar no dicionário do VOLP, não havendo nenhuma preocupação quanto a forma que estas palavras terão no dicionário final. Os verbos, novamente, constituem uma exceção. A cada radical de um verbo associa-se uma regra simbólica (v. seção 7.1.1) que abrange todos os sufixos que podem se associar a este radical para criação de uma forma verbal. A base de dados compreende ainda as tabelas que dão suporte ao tratamento diferenciado dado aos verbos, e

ao processo de criação do dicionário fonte e do dicionário objeto - Tabela de regras simbólicas, Tabela de regras e Tabela de sufixos.

Dicionário fonte: criado a partir dos dicionários da base de dados, com o auxílio das tabelas pertencentes a esta mesma base de dados, o dicionário fonte corresponde ao estágio intermediário entre a base de dados e o dicionário objeto. Cada entrada deste dicionário compreende um radical e todos os sufixos que a ele se ligam nos dicionários da base de dados. Quando se trata de um radical de um verbo, a ele também se associa uma regra simbólica, representando um conjunto de sufixos verbais. A finalidade deste dicionário é representar todas as palavras que possuem o mesmo radical em uma única entrada, facilitando o processo de criação do dicionário objeto.

Dicionário objeto: obtido a partir do dicionário fonte, este dicionário representa a forma final do dicionário, ou seja, o dicionário que o VOLP realmente utiliza. Cada entrada deste

dicionário compreende um radical e uma ou mais regras (v. seção 7.1.2). Estas regras são criadas de acordo com os sufixos que a este radical se associavam no dicionário fonte. O objetivo deste dicionário é auxiliar o VOLP na verificação das palavras.

Cada uma das três seções seguintes descreve o processo de criação de um destes componentes.

7.1 Criação da base de dados

A criação da base de dados corresponde à obtenção do dicionário, ou seja, corresponde à fase em que se obtém, de alguma maneira, as palavras que devem constar, de uma forma ou de outra, no dicionário do verificador ortográfico.

Decidimos obter os nomes dos verbos através de um método diferente daquele usado para obtenção das demais palavras. Isto se deve ao fato de que um verbo existente em um texto não deve ter o mesmo tipo de tratamento dado a uma outra palavra, já que cada flexão verbal encontrada no texto não deve constar explicitamente no dicionário, mas sim de forma implícita. Com a criação dos dicionários específicos de verbos é possível reconhecer um verbo no texto e, consequentemente, não submetê-lo ao processo de inclusão no dicionário de palavras.

Nos textos constam tanto o nome do verbo propriamente dito, quanto as flexões verbais correspondentes a este verbo. Com um

dicionário contendo apenas o nome dos verbos, estes são reconhecidos em um texto, o mesmo não acontecendo com suas formas verbais. A partir do dicionário de verbos e de uma tabela (Tabela de regras simbólicas) que especifica todos os sufixos abrangidos por uma regra simbólica, é possível se fazer uma expansão deste dicionário onde constam todas as formas verbais compreendidas pela conjugação dos verbos existentes no primeiro dicionário.

As palavras obtidas nos textos e que não são verbos constituirão o Dicionário de palavras. Independente do dicionário do qual a palavra faz parte, a sua ocorrência nos textos será contabilizada para efeito de triagem das palavras que devem constar no dicionário de memória.

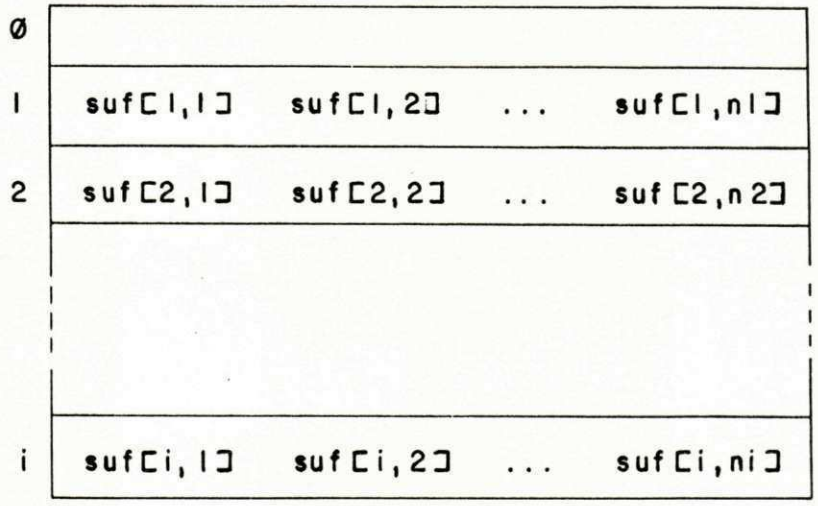
Com o objetivo de dar suporte ao processo de criação destes dicionários, bem como dos dicionários fonte e objeto, são criadas as seguintes tabelas: Tabela de regras simbólicas, Tabela de regras e Tabela de sufixos.

7.1.1 Tabela de Regras Simbólicas

Uma regra simbólica compreende todos os sufixos gerados pela conjugação de um verbo regular ou a um conjunto de sufixos regulares dentro da conjugação de verbos irregulares.

O uso de regras simbólicas permite a geração automática de toda a conjugação de um verbo regular ou de algumas formas de um verbo irregular. Além do mais, elimina a necessidade de associar todos os sufixos verbais ao radical de um verbo no dicionário fonte.

Sufixos Verbais



suf[j,i] = i-ésimo sufixo abrangido pela regra simbólica j (RSj)

7.2 - Tabela de regras simbólicas

Com base na conjugação dos verbos regulares e dos irregulares que possuam algumas formas verbais regulares constrói-se, a tabela de regras simbólicas (figura 7.2)

7.1.2 Tabela de regras

Como poderá ser constatado no decorrer deste capítulo, todo o processo de criação do dicionário se baseia na Tabela de regras (figura 7.3). De princípio, a criação desta tabela não será definitiva, já que as combinações feitas entre os diversos sufixos considerados, objetivando gerar as regras, são praticamente baseadas na intuição. Durante todo o processo de

obtenção do dicionário é possível observar o uso constante de determinados grupos de sufixos, possibilitando uma escolha mais acertada sobre a abrangência de determinadadas regras.

	Regras Simbólicas/sufixos	total de regras
Ø		
1	RS[1,1] RS[1,2] ... RS[1,n1]	Ø
2	RS[2,1] RS[2,2] ... RS[2,n2]	Ø
i	RS[i,1] RS[i,2] ... RS[i,ni]	Ø
i+1	suf[1,1] suf[1,2] ... suf[1,ni +1]	ni+1
255	suf[j,1] suf[j,2] ... suf[j,n255]	n255

RS[j,i] = i-ésima regra simbólica abrangida pela regra j(Rj)

suf[j,i] = i-ésimo sufixo abrangido pela regra j (Rj)

7.3 - Tabela de regras

Qualquer mudança nesta tabela implica na recriação do dicionário.

As regras que abrangem regras simbólicas devem ter valores sucessivos (i.e. devem corresponder a um certo intervalo entre os valores 1 a 255), para facilitar o processo de criação do dicionário objeto (seção 7.3).

Uma das restrições feita quanto à escolha do código de uma regra é que se o conjunto de sufixos abrangidos pela regra X (**RX**) contém o conjunto de sufixos abrangidos pela regra Y (**RY**), então o valor 'X' tem que ser menor que 'Y'. Esta restrição torna o processo de criação de regras para o dicionário objeto mais eficiente, (seção 7.3).

7.1.3 Tabela de sufixos

A tabela de sufixos (figura 5.5) é criada com base na tabela de regras simbólicas e na tabela de regras.

O uso da tabela de sufixos não é exclusivo do processo de verificação. A criação do dicionário de um VO que utiliza a análise de sufixos requer a extração dos sufixos das palavras antes que as mesmas sejam incluídas no dicionário. O VOLP usa a tabela de sufixos para reconhecer o sufixo que deverá ser extraído da palavra.

Esta tabela também será utilizada durante o processo de criação do dicionário objeto (seção 7.3).

7.1.4 Dicionário de verbos regulares

O dicionário de verbos regulares (figura 7.4) é criado

através da inclusão dos vocábulos contidos em dicionários de verbos da língua portuguesa.

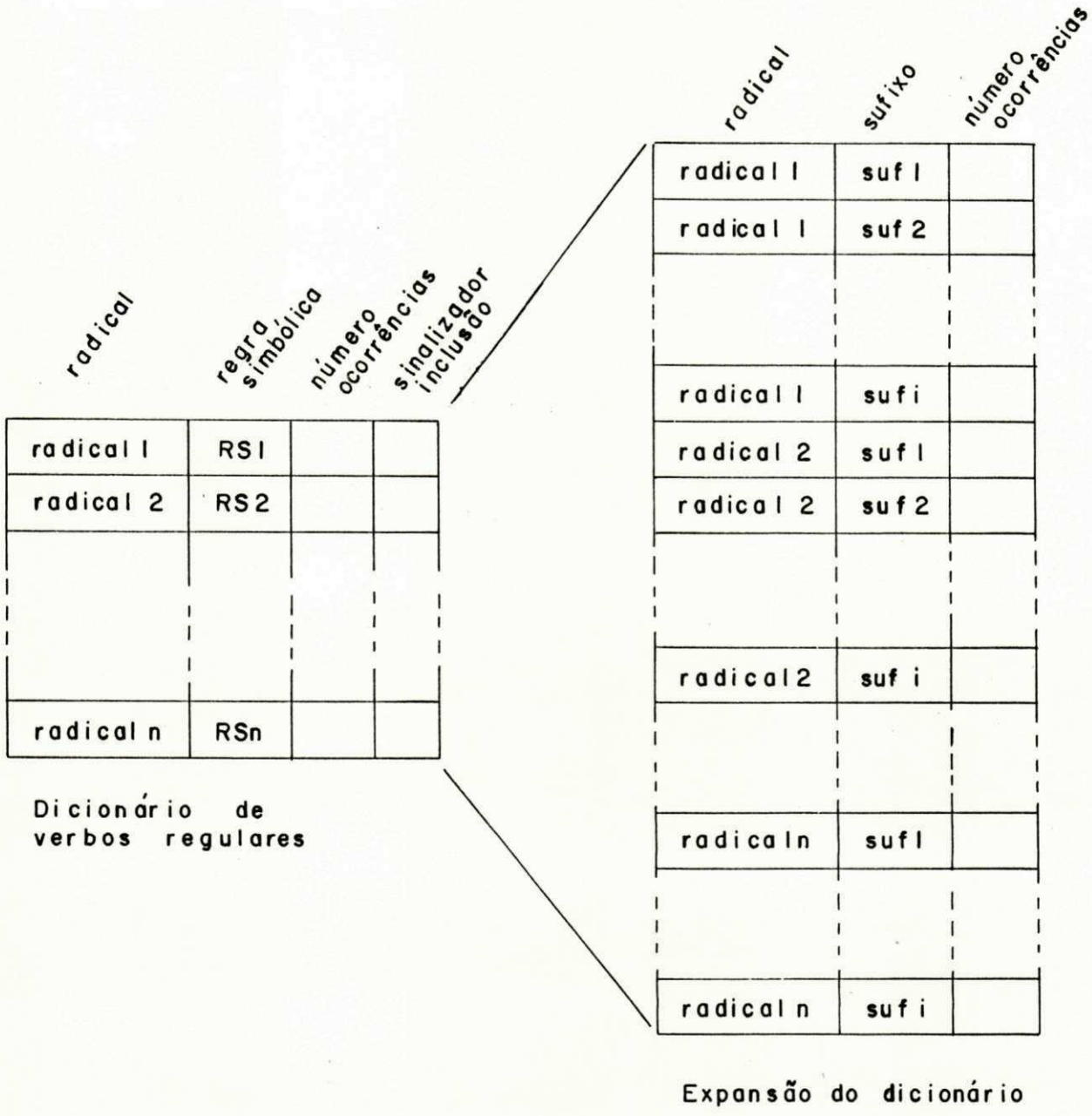


Figura 7.4 - Estruturas dos dicionários de verbos regulares

Após a sua criação este dicionário conterá todos os verbos - regulares e irregulares - da língua portuguesa. Com a criação

do dicionário de verbos irregulares (seção 7.1.5), é possível extrair-se do primeiro, todos os verbos contidos no segundo.

Feita a exclusão dos verbos irregulares, elimina-se a terminação de cada verbo ('AR', 'ER', 'IR'), gerando, automaticamente, a regra simbólica correspondente. Uma vez associada uma regra simbólica a cada radical, cria-se, com base na tabela de regras simbólicas, uma expansão deste dicionário contendo a conjugação completa de cada verbo. A finalidade deste dicionário expandido será a detecção dos verbos regulares nos textos utilizados para análise de freqüência de uso das palavras.

O campo "sinalizador inclusão" servirá para indicar, na fase de criação do dicionário secundário, se o verbo já foi incluído no dicionário principal ou não. Neste caso, o mesmo deverá ser incluído no dicionário secundário.

O campo "número ocorrências" na expansão do dicionário registra o total de ocorrências daquela forma verbal nos textos utilizados para obtenção de palavras e para análise de freqüência de uso de palavras.

O campo "número ocorrências" no dicionário armazena para cada verbo, o numero de ocorrências de sua forma verbal mais usada, ou seja, o maior valor registrado no campo "número ocorrências" de uma das entradas que constituem a conjugação do verbo na expansão do dicionário.

É através desta informação - número de ocorrências - que se faz a seleção das palavras que devem constar na memória ou no

disco.

7.1.5 Dicionário de verbos irregulares

As entradas do dicionário de verbos irregulares (figura 7.5) serão extraídas de gramáticas da língua portuguesa, cabendo ao Administrador de Dicionário fazer esta pesquisa, determinar se há uma regra simbólica que abranja parcialmente a conjugação do verbo, e incluir o radical do verbo associado a uma regra simbólica ou uma forma verbal literal. Neste caso não deve haver regra simbólica associada.

Verbos regulares que têm seus radicais alterados durante a conjugação são considerados irregulares e, portanto, deverão constar neste dicionário. Cada modificação do radical constituirá uma entrada do dicionário associada a uma regra simbólica.

Com base na tabela de regras simbólicas será feita então uma expansão deste dicionário, obtendo-se assim a conjugação de todos os verbos irregulares. As formas que constam literalmente no dicionário devem ser incluídas de forma idêntica na expansão do dicionário.

Esta expansão será utilizada para detecção dos verbos irregulares nos textos utilizados para análise de frequência de uso das palavras.

O uso dos campos "número ocorrências" e "sinalizador inclusão" no dicionário e "número ocorrências" na expansão é idêntico ao uso no dicionário de verbos regulares (secção 7.1.4).

Quando uma entrada do dicionário corresponder a uma forma verbal o número de ocorrências será igual ao número de ocorrências desta forma verbal na expansão.

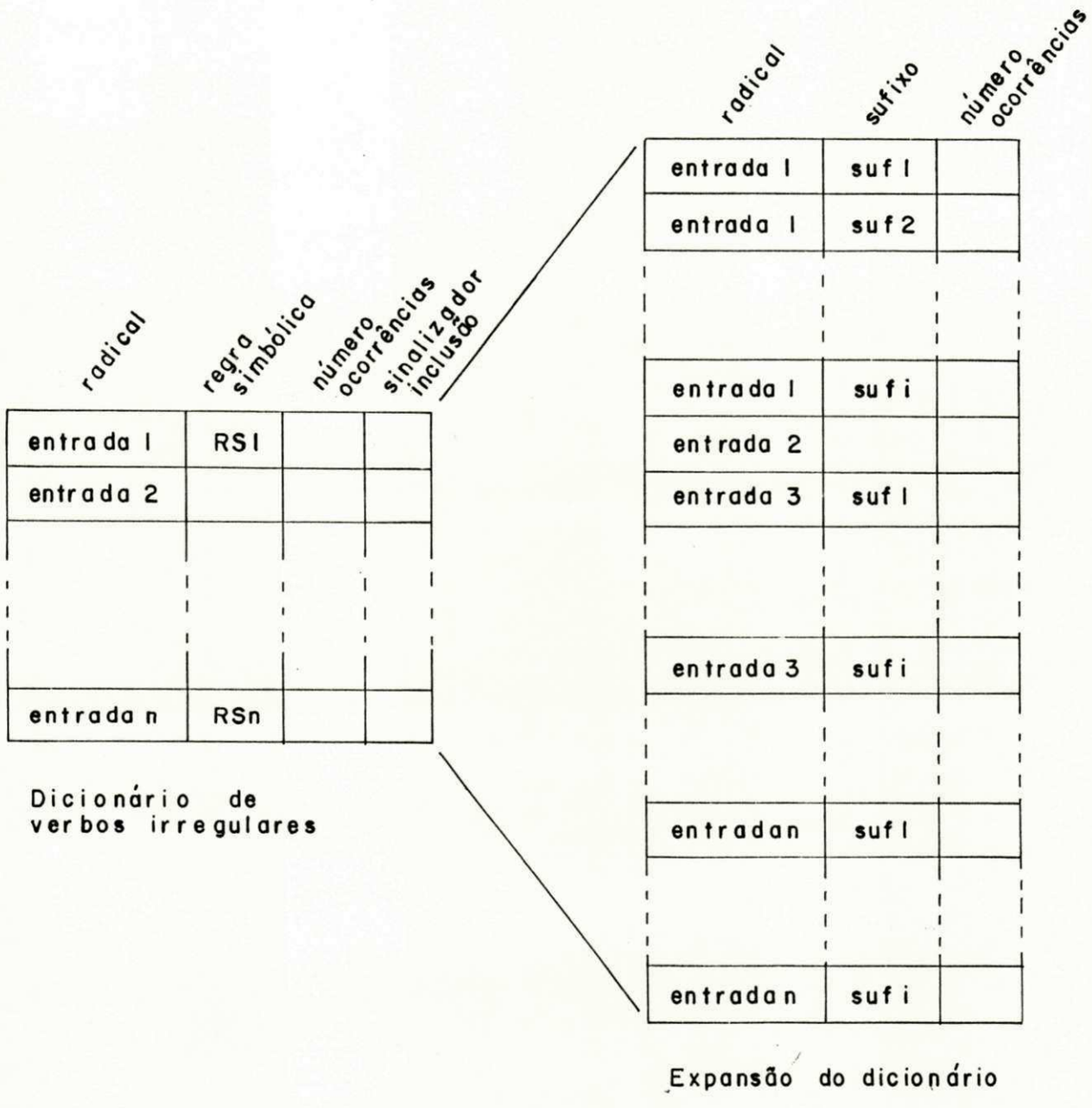


Figura 7.5 - Estruturas dos dicionários de verbos irregulares

7.1.6 Dicionário de palavras

Este dicionário (figura 7.6) é constituído de palavras obtidas em textos da língua portuguesa.

Uma vez implantado o VOLP, as listas de palavras desconhecidas e apresentadas ao usuário como incorretas poderão servir de entrada para este dicionário, tendo-se antes o cuidado de se eliminar as palavras realmente incorretas.

entrada	sufixo	número ocorrências	sinalizador inclusão
entrada 1	suf 1		
entrada 2	suf 2		
entrada 3	-		
entrada n	suf n		

Figura 7.6 - dicionário de palavras

Caso a palavra contenha um sufixo reconhecido pelo VOLP, a mesma estará armazenada neste dicionário de forma desmembrada - radical e sufixo. Caso contrário, ela será armazenada de forma integral. O reconhecimento de um sufixo na palavra antes de sua inclusão no dicionário se dá através da tabela de sufixos.

O campo "número ocorrências" indica o número de ocorrências da palavra nos diversos textos já analisados com este fim.

O campo "sinalizador inclusão", da mesma forma que nos dicionários de verbos regulares e irregulares, indica se a palavra foi incluída na memória ou não.

A análise de um texto é feita de acordo com os seguintes passos:

passo 1: Pesquisar no texto nomes próprios, siglas e termos técnicos, retirando-os. Poderão ser incluídos nos dicionários específicos, posteriormente.

passo 2: Ordenar o texto, somar o número de ocorrências, eliminando, em seguida, as duplicações de palavras. Analisar então, cada palavra do texto.

passo 3: Verificar se a palavra é um verbo, ou seja, se ela consta em uma das duas expansões de verbos. Se for verbo, somar número de ocorrências na forma verbal correspondente.

passo 4: Verificar se a palavra já existe no dicionário de palavras. Caso exista somar número de ocorrências da palavra. Caso contrário, armazenar temporariamente a palavra e seu número de ocorrências em uma lista de novas palavras.

passo 5: Após a análise de todas as palavras distintas do texto incluir novas palavras e o número de ocorrências no

dicionário de palavras.

passo 6: Atualizar o número de ocorrências dos dicionários de verbos regulares e irregulares baseado no número de ocorrências das expansões correspondentes.

passo 7: Ordenar decrescentemente os dicionários de verbos regulares, de verbos irregulares e de palavras, pelo número de ocorrências.

A criação do dicionário principal já é possível com as informações contidas na base de dados. Isto torna a estrutura do dicionário fonte mais simples, como também simplifica o processo de triagem das palavras que vão para a memória.

Para a criação deste dicionário, serão considerados, conjuntamente, os três dicionários ordenados decrescentemente pelo número de ocorrências. Para cada entrada de um destes dicionários que apresentar o maior número de ocorrências fazer as seguintes considerações:

- Se for um radical de um verbo, incluí-lo no dicionário principal, associando-lhe a regra correspondente a regra simbólica existente na entrada do dicionário da base de dados.

- Se for uma forma verbal ou uma palavra, incluí-la integralmente no dicionário principal associando-lhe a regra zero (**RO**).

- Independente do caso, setar o "sinalizador inclusão" da entrada da base de dados que foi incluída no dicionário principal

Este procedimento deverá ter continuidade até que se tenha atingido o limite de entradas permitidas no dicionário principal,(i.e. o limite de densidade da tabela "Hash" pré-estabelecido, que permite um uso eficiente do método).

7.2 Criação do dicionário fonte

O nosso objetivo, com a criação do dicionário fonte, é especificar todos os sufixos e regras simbólicas que se ligam ao mesmo radical, possibilitando a criação, a partir dele, de um dicionário com a estrutura adequada ao método de verificação utilizado pelo VOLP.

Cada entrada do dicionário fonte (figura 7.7) compreende um radical mais os sufixos e/ou regras simbólicas que, na base de dados, a ele se associam.

Este dicionário serve de base para criação do dicionário secundário (objeto). No entanto, os sufixos e/ou regras simbólicas já incluídos no dicionário principal devem constar no dicionário fonte. Isto evita que, durante a criação do dicionário secundário, faça-se uso inadequado da lista de exceções, uma vez que a inexistência de um determinado sufixo significaria que o mesmo já tinha sido incluído no dicionário de memória, em vez de significar que o sufixo não consta na base de dados.

entrada	sufixos/sinalizadores de inclusão						
entrada 1	suf 1	sin 1	suf 2	sin 2		suf n	sin n
entrada 2	suf 1	sin 1	suf 2	sin 2		suf m	sin m
entrada i	suf 1	sin 1	suf 2	sin 2		suf j	sin j

sufi = i-ésimo sufixo ou i-ésima regra simbólica
sini = sinalizador de inclusão do sufi

Figura 7.7 - Estrutura do Dicionário Fonte

Os sufixos e/ou regras simbólicas incluídos no dicionário de memória constam deste dicionário com os seus respectivos campos "sinlalizadores inclusão" ligados. Os demais terão seus "sinalizadores inclusão" desligados e só serão ligados a medida que forem sendo incluídos no dicionário secundário.

Caso não se deseje incluir no dicionário do VOLP palavras ou verbos raramente usados, pode-se estabelecer um **número mínimo de ocorrências** para que uma palavra ou verbo passe a constar no dicionário fonte, e, consequentemente, no dicionário do VOLP. Como os três dicionários se encontram decrescentemente ordenados pelo número de ocorrências, após encontrar-se a primeira entrada de um dos dicionários que tenha um número de ocorrências menor que o limite mínimo pré-estabelecido, deixa-se de considerar

aquele dicionário.

Caberá ao Administrador de Dicionário analisar se o fato de não se incluir determinados sufixos (pertencentes a palavras com número de ocorrência inferior ao limite mínimo) contribui para a geração de muitas exceções. Quando isto ocorrer, o administrador poderá atribuir diretamete a estas palavras um número de ocorrências igual ou maior ao limite mínimo.

Na criação do dicionário fonte, o critério de inclusão continua sendo a entrada com maior número de ocorrências, levando-se em consideração os três dicionários, em paralelo.

Quando a entrada a ser incluída no dicionário corresponder a um radical, deve-se antes pesquisá-lo no dicionário fonte, a fim de se certificar que ele já não foi incluído. Neste caso, basta associar ao radical do dicionário fonte, o sufixo ou a regra simbólica.

7.3 Criação do dicionário objeto

O dicionário secundário (figura 5.6) deverá abranger todos os sufixos e/ou regras simbólicas associados a cada entrada do dicionário fonte que não tenham sido incluídos no dicionário principal.

Para cada sufixo e/ou regra simbólica disponível no dicionário fonte devem ser feitas as seguintes considerações:

- Para uma regra simbólica deve-se pesquisar a existência de outra(s) regra(s) simbólica(s) que possam

ser abrangidas por uma única regra. Caso não existam outras regras simbólicas, incluir no dicionário objeto a regra que abranja unicamente aquela RS. Ligar o campo "sinalizadores inclusão" da(s) regra(s) simbólica(s) incluída(s).

- Para um sufixo deve-se considerar todas as regras (Tabela de sufixos) que abranjam aquele sufixo.

- Caso todos os sufixos abrangidos por uma regra constem na entrada do dicionário fonte (considerando também aqueles já incluídos no dicionário principal), incluir esta regra no dicionário objeto, ligando o "sinalizador inclusão" de todos os sufixos abrangidos.

- Caso nem todos os sufixos abrangidos por uma regra constem na entrada do dicionário fonte, deve-se calcular o percentual de abrangência da regra de acordo com o total de sufixos abrangidos pela regra e a existência deste sufixos na entrada do dicionário fonte. Incluir a regra e o seu percentual de abrangência na Lista de regras consideradas (figura 7.8). Após considerar todas as regras que abrangem aquele sufixo, pesquisar na Lista de regras consideradas a regra de maior percentual de abrangência e considerar a abrangência desta regra.

- Considerar a abrangência de uma regra significa

analisar se a regra atingiu a abrangência mínima pré-estabelecida. Caso tenha atingido, a regra deve ser incluída no dicionário objeto com restrições. Caso a abrangência mínima não tenha sido atingida, deve-se incluir a palavra integral (radical + sufixo considerado) em uma outra entrada do dicionário objeto, associando-lhe a regra zero (R0).

- Incluir uma regra com restrições no dicionário objeto significa dizer que além de se incluir a regra neste dicionário, deve-se ligar o "bit" posicionalmente correspondente a regra, gerar e incluir na Lista de exceções as palavras que constituem a exceção da regra. Ligar "sinalizador inclusão" dos sufixos incluídos.

- A cada nova entrada do dicionário fonte a ser analisada, eliminar lista de regras consideradas.

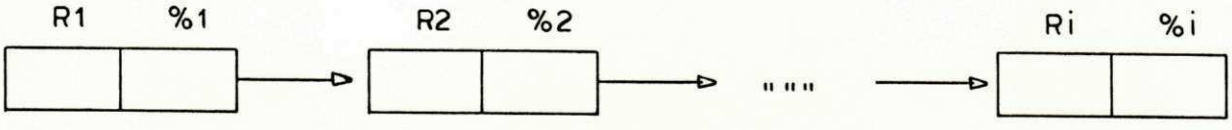


figura 7.8 - Lista de regras consideradas

A análise da freqüência de uso contribui para que as palavras de um texto em verificação sejam mais facilmente encontradas no dicionário de memória, reduzindo-se os acessos a disco. Se a este critério de inclusão de palavras no dicionário associamos o método proposto por Brent, que reduz o tempo gasto na pesquisa de uma palavra, seja no dicionário principal ou no secundário, obtém-se um verificador ortográfico de melhor "performance".

8. CONCLUSÕES

O desenvolvimento de interfaces mais flexíveis para a entrada de dados de um documento no computador tem sido um tópico de constantes pesquisas na ciência da computação.

Dentre estas pesquisas destacamos aquelas voltadas para a verificação ortográfica de textos. Diversos verificadores ortográficos para a língua inglesa já foram desenvolvidos, facilitando a tarefa de revisão de textos escritos nesta língua.

A revisão automática da ortografia de um texto é de grande importância, haja visto a capacidade que tem a mente humana de trocar, mesmo que inconscientemente, uma palavra ortograficamente incorreta por outra, ortograficamente correta, ao revisar um texto.

Algumas características da língua portuguesa (v. capítulo 4) levam as pessoas a cometer, frequentemente, erros ortográficos na criação de textos. O uso constante do computador como ferramenta na preparação destes textos contribui para a inclusão de novos erros, sendo estes decorrentes do processo de digitação.

Na época em que iniciamos a pesquisa sobre o assunto deste trabalho, não existia nenhum verificador ortográfico no mercado, que detectasse erros da língua portuguesa.

Todos estes fatores contribuíram para que decidíssemos desenvolver uma pesquisa nesta área. O propósito de nosso trabalho foi então estabelecido: projetar um verificador

ortográfico para a língua portuguesa. Mesmo com o posterior surgimento de um verficador ortográfico para a língua portuguesa, o Best Spell, nós continuamos com o nosso propósito.

Projetamos um verificador ortográfico para a língua portuguesa (VOLP) que além de detectar os erros ortográficos de um texto, faz um controle sobre o uso adequado de caixa das letras.

Como continuidade do nosso trabalho sugerimos, inicialmente, o desenvolvimento do VOLP e, a obtenção, em paralelo, do dicionário. O uso de linguagens de alto nível tem influenciado negativamente no desempenho de verificadores ortográficos para a língua inglesa [PETE 80]. Melhor seria desenvolver o VOLP na linguagem de programação C, considerando-se ainda a necessidade de se manipular informações a nível de bit.

Com a implantação do VOLP surge a possibilidade de se desenvolver uma análise, através de um uso exaustivo e criterioso, objetivando detectar se este verificador apresenta características que se adaptam perfeitamente às propriedades e características de uso da língua portuguesa. Com isto se adquire estatísticas de desempenho do VOLP.

Um outro trabalho que poderia ser feito seria adaptar o VOLP a um processamento "on-line". Com o processamento "on-line", poder-se-ia ir mais longe: ampliar a funcao básica do VOLP de verificação para correção ortográfica.

O VOLP se adapta perfeitamente a outra línguas, desde que

um número razoável de suas palavras sejam formadas de radical e sufixo. Uma vez especificados os sufixos que serão abrangidos pela análise de sufixos, é possível criar as regras que estabelecerão o controle sobre a ligação dos radicais com estes sufixos, na formação de palavras. Criando-se as regras da língua, todo o processo de obtenção de dicionários, bem como o processo de verificação não sofrem nenhuma alteração.

2. REFERÊNCIAS BIBLIOGRÁFICAS

[AHO 83] AHO, Alfred V HOPCROFT, Jonh E e ULLMAN, Jeffrey D. **Data Structures and Algorithms.** Addison-Wesley Publishing Company 1983.

[BENT 85] BENTLEY, Jon. **Programming Pearls**, CACM, Vol. 28, No. 5, maio,1980 pp. 676-687.

[HORO 76] HOROWITZ, E e SAHNI, S. **Fundamentals of Data Structures.** Computer Science Press, Inc. 1976.

[KNUT 73] KNUTH, D E. **The Art of Computer Programming "Sorting and Searching".** Vol. 3. Addison-Wesley Publishing Company 1973.

[McIL 82] McILROY, M Douglas. **Development of Spelling List**, IEEE Transaction on Communications, Vol. COM-30, No. 1, janeiro,1982 pp. 91-99.

[PETE 80] PETERSON, James L. **Computer Programs for Detecting and Correcting Spelling Errors**, CACM, Vol. 23, No. 12, dezembro,1980, pp. 676-687.

[PETE 86] PETERSON, James L. **A Note on Undetected Typing Errors**, CACM, Vol. 29, No. 7, julho,1986, pp. 633-637.

[WILD 86] WILD West Software. **Best Spell - Corretor Ortográfico.** Manual de usuário 1986.

[TURB 85] TURBA, T. N. **"Checking for Spelling and Typographical Errors in Computer-Based Text"** Computer Text Recognition and Error Correction - Tutorial, Sargur N. Srihari IEEE Computer Society, 1985 pp. 294-303.