UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE SEMICONDUTORES INSTRUMENTOS E FOTÔNICA

EXTRAÇÃO AUTOMÁTICA DE PALAVRAS-CHAVE NA LÍNGUA PORTUGUESA APLICADA A DISSERTAÇÕES E TESES DA ÁREA DAS ENGENHARIAS

Maria Abadia Lacerda Dias
Orientador: Prof. Dr. Mauro Sérgio Miskulin

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação (FEEC-UNICAMP) como parte dos requisitos exigidos para obtenção do título de Mestre em Engenharia Elétrica.

Área de Concentração: Área de Eletrônica, Microeletrônica e Optoeletrônica

Banca Examinadora

Prof. Dr. Mauro Sérgio Miskulin (Orientador)
UNICAMP – Universidade Estadual de Campinas – Campinas – SP

Prof. Dr. Fernando José Von Zuben (Membro Interno)
UNICAMP – Universidade Estadual de Campinas – Campinas – SP

Prof. Dr. Marcelo Araújo Franco (Membro Externo)
UNICAMP – Universidade Estadual de Campinas – Campinas – SP

Profª. Drª. Maria Isabel Santoro (Membro Externo)
UNICSUL – Universidade Cruzeiro do Sul – São Paulo – SP

28 de outubro de 2004

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

D543e

Dias, Maria Abadia Lacerda
Extração automática de palavras-chave na língua portuguesa aplicada a dissertações e teses da área das engenharias / Maria Abadia Lacerda Dias. --Campinas, SP: [s.n.], 2004.

Orientador: Mauro Sérgio Miskulin.
Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Palavras-chave. 2. Recuperação da informação. 3. Algoritmos de computador. 4. Processamento da linguagem natural (Computação). 5. Língua portuguesa – Morfologia. I. Miskulin, Mauro Sérgio. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e da Computação. III. Título.

Eu dedico esta dissertação à memória de minha querida mãe Julieta, que me ensinou a viver com simplicidade, generosidade e um imenso amor dedicado aos filhos.

AGRADECIMENTOS

Ao Prof. Dr. Mauro Sérgio Miskulin, meu orientador, pela oportunidade, confiança e atenção concedidas durante o desenvolvimento deste trabalho.

Ao Rubens Queiroz, por ter me apoiado, incentivado, sugerido grandes idéias e por estar disponível nos momentos em que eu mais precisei para começar e concluir esta dissertação.

Ao Marcelo, pela inesgotável fonte de amor, carinho, atenção, paciência e motivação durante todos os dias. Agradeço também por ter sido o meu melhor amigo, o meu companheiro de laboratório e de discussões sobre este assunto. Alguns experimentos e entendimentos não teriam sido possíveis sem a sua colaboração. Você foi muito importante para eu chegar até aqui.

Aos meus pais, Abadio e Julieta, pelo grande apoio e esforço para eu continuar estudando. Ao meu irmão Sérgio, por ser um irmão maravilhoso. Às minhas queridas irmãs, Juliana e Joelma, por serem as melhores amigas e por terem acreditado em mim e me apoiado no retorno para concluir este objetivo tão importante. Aos meus lindos sobrinhos, Diéssica, Fernanda e Paulo Sérgio, por me darem tantas alegrias e emoções nesta vida.

Aos amigos que sempre estiveram ao meu lado, me dando força, apoio, incentivo e acreditando na minha capacidade.

Ao Centro Universitário UNIVATES por disponibilizar a infraestrutura que auxiliou o desenvolvimento desta pesquisa.

RESUMO

O objetivo desta dissertação é adaptar um algoritmo de extração automática de palavras-chave para a língua portuguesa. Palavras-chave fornecem uma descrição adequada do conteúdo de um documento. Tal descrição facilita aos futuros leitores decidirem se o documento é ou não relevante para os mesmos. As palavras-chave têm também outras aplicações, já que estas resumem documentos de forma sucinta. Portanto podem ser usadas como uma medida eficiente de similaridade entre documentos, tornando possível organizá-los em grupos ao se medir a sobreposição entre as palavras-chave que estão associadas. Esta adaptação consiste na utilização de um algoritmo de radicalização de palavras na língua portuguesa, o qual foi aperfeiçoado neste estudo, e uma lista de *stopwords* da língua portuguesa, apresentada neste trabalho.

ABSTRACT

The goal of this dissertation is to adapt an automatic extraction algorithm of keywords for the Portuguese language. Keywords give an adequate description of a document's contents. Such description helps future readers to decide whether the document is relevant or not for them. The keywords have also other applications, because they summarize documents in a brief way. Therefore, they can be used as an efficient measure of similarity between documents, making possible to organize them in groups when measuring the overlap between the keywords they are associated to. This adaptation consists on the utilization of a stemming algorithm for words of the Portuguese language, which was improved in this study, and a list of stopwords of Portuguese language, also presented in this work.

SUMÁRIO

1 INTRODUÇÃO

Nos dias de hoje a informação é moeda corrente para todas as áreas de conhecimento, e conseqüentemente o volume de dados gerado diariamente é colossal. Muitos afirmam que estamos na chamada "Era da Informação", onde essa massa de dados precisa também ser armazenada, classificada, selecionada e transformada em novas informações.

Em particular, a quantidade de informações armazenadas em bancos de dados aumenta a cada minuto, ultrapassando a capacidade humana para sua interpretação, o que se torna um problema para os especialistas de Tecnologia da Informação.

Um exemplo ocorre no processo de pesquisa, quando deseja-se localizar documentos em uma biblioteca digital. Tal processo não é simples, pois normalmente existe um número muito grande de documentos disponíveis, cada um com milhares de palavras, onde nem todas são relevantes para a busca realizada. Além disso, quando diversos documentos são selecionados, uma outra dificuldade se apresenta: como ordená-los de forma coerente?

Outra situação ocorre em listas de mensagens e fóruns de discussão. Neste caso, os textos são menos formais e contém maior quantidade de ruído (comentários já feitos por outras pessoas, saudações, abreviações e observações não relacionadas ao tema). Além disso os blocos de texto são mais curtos, pois são apenas mensagens e não artigos completos, e ainda muito mais numerosos. O desafio nesse caso é organizar e classificar o conteúdo desses textos, categorizando o que foi escrito pelos usuários das listas de mensagens e dos fóruns de discussão, de forma que o resultado possa ser pesquisado e eventualmente sintetizado futuramente em resumos.

Vale ressaltar que tais informações não são apenas dados numéricos, mas também linguagem humana em forma escrita, e portanto, com uma lógica e coerência específicas da linguagem utilizada. Então, não basta apenas ter domínio de ferramentas computacionais como banco de dados e mecanismos de busca. É fundamental ter conhecimento também da estrutura e eventualmente do sentido da construção das frases utilizadas, que são específicos para cada linguagem humana.

Existe uma área de conhecimento que se dedica ao estudo, tratamento e compreensão da linguagem humana através de tecnologia computacional, denominada Processamento de Linguagem Natural (PLN). Este campo apresenta técnicas gerais que se aplicam a qualquer linguagem escrita ou falada, mas também inclui métodos adicionais que conhecem as particularidades da linguagem tratada.

Em especial, as técnicas de radicalização se configuram como ferramentas muito úteis para tratar linguagem natural, pois permitem reduzir todas as palavras de um texto ou de uma pesquisa aos radicais que as compõem, de forma a agrupar por similaridade variações ortográficas que de outra forma passariam como palavras completamente distintas. Outra técnica importante de PLN é a caracterização de *stopwords*, que são palavras freqüentes em um texto mas que não têm maior relevância para o assunto tratado.

As seções seguintes descrevem as motivações deste estudo, os objetivos e principais contribuições, a metodologia utilizada, a organização deste trabalho e a notação usada no texto.

1.1 Motivação

A motivação inicial para este trabalho foi a Biblioteca Digital da UNICAMP (Libdig, 2004). Esta biblioteca virtual é baseada em um sistema chamado Nou-Rau (Nou-Rau, 2004), e armazena um grande número de publicações em formato digital. Em outubro de 2004, totalizou 9216 documentos, sendo que 3481 eram dissertações e teses defendidas nesta universidade.

O Nou-Rau é um Software Livre, ou seja, um programa que pode ser livremente utilizado, distribuído e adaptado para qualquer propósito. Este sistema é capaz de armazenar diversos tipos de documentos em formato digital, provendo um mecanismo integrado de busca,

que leva em conta o conteúdo textual de cada documento. Por exemplo, cada documento pode conter informações tais como: título, nomes dos autores, endereços eletrônicos dos autores, descrições curtas dos documentos (resumos), lista de palavras-chave e espaço para informações adicionais.

Quando é realizada uma busca a partir da página principal do sistema, todos esses dados são consultados para selecionar documentos que possam ser relevantes para a busca realizada. O mecanismo de busca é implementado pela ferramenta livre HTDIG (Htdig, 2004), que faz o processo de indexação e recuperação dos documentos. Esse sistema cria uma base de dados indexados que é otimizada para busca, ou seja, a operação de geração de índices é lenta e complexa, mas permite acelerar o processo de identificação de documentos que possam ser relevantes para uma pesquisa.

Contudo, as técnicas usadas pelo HTDIG são simples, pois o sistema não utiliza nenhum conhecimento específico sobre os documentos para organizá-los. Em particular, são utilizadas apenas informações estatísticas que independem da língua dos textos. Em contrapartida, o mecanismo de busca usa pesos diferentes para os diversos tipos de informações associadas a cada documento, tendo como prioridade o título e as palavras-chave associadas ao documento.

No caso particular da Biblioteca Digital da UNICAMP, as palavras-chave são manualmente definidas pelos autores com o auxílio dos bibliotecários que consultam um grande índice de termos, selecionando entre eles os mais adequados para cada documento. Esse índice é chamado *thesaurus*[1] e é padronizado pela biblioteca nacional. O *thesaurus* não é estático, pois existe a possibilidade de acrescentar termos propostos e comprovados pelos autores.

Palavras-chave fornecem uma descrição do conteúdo de um documento. Esta descrição facilita aos futuros leitores decidirem se o documento é ou não relevante para os mesmos. As palavras-chave também podem ser usadas como uma medida eficiente de similaridade entre documentos, tornando possível agrupar documentos, ao se medir a sobreposição entre as palavras-chave associadas.

Em um ambiente organizado e controlado como a Biblioteca Digital da UNICAMP, é possível garantir que todos os documentos inseridos tenham palavras-chave e que estas sejam

1Repertório alfabético de termos utilizados em indexação e na classificação de documentos.

escolhidas de forma a resumir adequadamente os documentos. Só que de forma geral, isso só se aplica a documentos que são publicados, como artigos de congressos e dissertações. Porém, nem todos os tipos de documentos têm condições de passar por tal controle, como relatórios técnicos, notícias digitalizadas de jornais e textos informativos, em decorrência dos autores não serem obrigadas a fazer isso.

Em geral, quando leva-se em conta documentos em outras organizações e mesmo na Internet, somente uma pequena minoria deles têm palavras-chave determinadas pelo autor. De fato, determinar palavras-chave manualmente para uma grande massa de documentos existentes é muito trabalhoso. Portanto, é altamente desejável a possibilidade de automatizar o processo de determinação de palavras-chave.

Tal ferramenta de extração de palavras-chave também seria bastante útil no contexto citado anteriormente de listas de mensagens e fóruns de discussão, pois as palavras-chave poderiam ser utilizadas como elementos de agregação e sumarização dos textos envolvidos.

Uma outra motivação para se trabalhar com extração de palavras-chave de textos da língua portuguesa veio da relevância e da necessidade de disponibilização de ferramentas para esse campo de aplicações. Os valores desta pesquisa são ressaltados pela dependência da língua portuguesa nesse tipo de tarefa e pela escassez de estudos realizados com esse objetivo.

A UNICAMP também dispõe de um sistema denominado Rau-Tu (Rau-Tu, 2004) que pode ser beneficiado por tais ferramentas. O Rau-Tu é um sistema livre e gratuito de perguntas e respostas via Internet, sendo distribuído como Software Livre sob a licença GPL. O objetivo do sistema Rau-Tu é permitir que vários usuários possam fazer perguntas sobre diversas áreas de conhecimento, e que colaboradores respondam a estas perguntas, alimentando uma base de perguntas e respostas que podem ser consultadas livremente. De fato, o Rau-Tu é uma fusão entre uma lista e um fórum, onde toda comunicação é arquivada para consulta futura. Portanto, o próprio sistema Rau-Tu poderia se beneficiar de ferramentas de extração automática de palavras-chave.

Com base na apresentação feita até o momento, pode-se afirmar que palavras-chave facilitam a filtragem e a organização de documentos, tornando possível selecionar aqueles que são provavelmente relevantes. Portanto, são de grande interesse caminhos para automatizar a

obtenção de palavras-chave.

Existem dois caminhos para abordar o problema: determinação de palavras-chave e extração de palavras-chave. Determinação de palavras-chave seleciona as palavras-chave para um documento escolhendo-as exclusivamente de um vocabulário controlado e pré-existente. Extração de palavras-chave escolhe palavras-chave do próprio texto, sem passar por um controle de vocabulário.

No contexto de uma grande massa de documentos sem uma prévia organização ou controle, estima-se que o processo de extração de palavras-chave seja mais geral e exija menos interferência humana, sendo portanto, a abordagem utilizada neste trabalho.

1.2 Objetivos

Esta dissertação tem três objetivos, listados a seguir.

O primeiro objetivo é estudar técnicas de extração automática de palavras-chave. Em particular, discutem-se métodos propostos na literatura tanto para a língua inglesa quanto para a língua portuguesa.

O segundo objetivo é propor um algoritmo de extração automática de palavras-chave para textos na língua portuguesa. Para isto é necessário estudar técnicas de radicalização de palavras e também elaborar uma lista de *stopwords* para a língua portuguesa.

O terceiro objetivo é validar a solução proposta utilizando para isso teses e dissertações oriundas da Biblioteca Digital da UNICAMP, focando na área das Engenharias.

1.3 Metodologia

A metodologia utilizada para a realização deste estudo foi principalmente execução de pesquisa bibliográfica, identificando o que já foi estudado e realizado na área de extração automática de palavras-chave, tanto para a língua inglesa quanto para a língua portuguesa. Esta identificação foi feita por meio de uma pesquisa de artigos, teses, dissertações e livros relacionados à área de Recuperação de Informação, Extração Automática de Palavras-Chave, Processamento de Linguagem Natural e língua portuguesa.

Em um segundo momento, foi feita uma proposta de uma adaptação para a língua portuguesa de uma técnica já existente, que foi seguida por uma implementação prática. Com base nessa implementação foram realizados uma série de experimentos para avaliar a efetividade do algoritmo proposto.

1.4 Organização

Esta dissertação está organizada da seguinte forma:

No Capítulo 2 são apresentados alguns conceitos básicos sobre o Processamento de Linguagem Natural e também estabelecida sua relação com a área de Recuperação de Informação.

No Capítulo 3 são apresentados quatro algoritmos de extração automática de palavras-chave de textos. Entre estes é descrito em detalhes o algoritmo de extração automática de palavras-chave de textos para a língua inglesa, denominado KEA. A adaptação deste algoritmo para a língua portuguesa é o principal objetivo desta dissertação.

No Capítulo 4 são apresentados três algoritmos de radicalização de palavras para a língua portuguesa. Em particular, é descrito o algoritmo Portuguese Stemmer que é utilizado na implementação do KEA. Os outros algoritmos estudados são o Porter Stemmer e o Pegastemming.

No Capítulo 5 são discutidos os passos realizados para a adaptação do algoritmo KEA para a língua portuguesa, o que envolve a construção de uma lista de *stopwords* e aprimoramento do algoritmo Portuguese Stemmer.

No Capítulo 6 inicialmente são discutidos os testes e feita uma avaliação dos resultados obtidos. Em seguida, são apresentadas as conclusões deste trabalho, sendo comentadas as contribuições do mesmo e sugeridas direções futuras de pesquisa.

1.5 Notação

As notações usadas nesta dissertação são as seguintes:

As palavras que estão em *itálico* são termos da língua inglesa, que foram mantidos em sua forma original. Quando existe tradução adequada para a língua portuguesa, estes termos são colocados entre parênteses e é utilizada a tradução.

As palavras que estão em **negrito** estão sendo definidas naquele momento no texto e serão utilizadas posteriormente. Todas as palavras definidas desta forma estão presentes no Glossário.

Todas as palavras usadas como exemplo estão sublinhadas.

Detalhes de implementação ou nomes de termos vinculados a algoritmos estão na fonte `courier.`

2 PROCESSAMENTO DE LINGUAGEM NATURAL

Este capítulo descreve conceitos básicos do campo de conhecimento chamado Processamento de Linguagem Natural e sua relação com a área denominada Recuperação de Informação.

Em particular, será apresentada brevemente a aplicação de conhecimento lingüístico, envolvendo etiquetagem de texto, normalização de variações lingüísticas, eliminação de *stopwords,* aplicação de métodos estatísticos e técnicas de conflação.

2.1 Recuperação de informação

De acordo com (Jackson & Moulinier, 2002), Recuperação de Informação pode ser definida como a aplicação de tecnologia computacional para a aquisição, organização, armazenamento, recuperação e distribuição de informação. (Scholtes, 1993) *apud* (Zuchini, 2003) afirma que Recuperação de Informação requer a aplicação conjunta de técnicas de Processamento de Linguagem Natural e Inteligência Artificial.

Segundo tais definições, pode-se concluir que sistemas de **Recuperação de Informação** (RI) tratam essencialmente de indexação, busca e classificação de documentos textuais, com o objetivo de satisfazer a necessidade do usuário de acesso à informação. Esse acesso é tradicionalmente expresso através de buscas.

Um usuário de um mecanismo de busca parte de uma necessidade de informação, para a

qual ele ou ela realiza uma consulta com o objetivo de encontrar documentos relevantes. Esta consulta pode não ser a melhor articulação para tal necessidade, ou a melhor abordagem para ser utilizada em um particular conjunto de documentos. Por exemplo, tal consulta poderia conter palavras mal escolhidas ou mal escritas. Poderia ainda conter palavras em excesso ou em número insuficiente. Todavia, essa consulta fornece a única pista que o mecanismo de busca tem com respeito ao objetivo do usuário.

A multitude de material na Internet é um dos fatores que tem mantido interesse em métodos automáticos para extrair informações de textos. Segundo (Jackson & Moulinier, 2002), Extração de Informação difere de Recuperação de Informação. Em **Extração de Informação** o foco não está em encontrar documentos, mas sim em encontrar informações úteis dentro de documentos. Tipicamente, um grande conjunto de documentos digitais são examinados para verificar se eles contém certos termos procurados, e portanto terem mérito de serem analisados posteriormente.

Os sistemas de Recuperação de Informação foram originalmente desenvolvidos para auxiliar o gerenciamento do grande volume de informações que tem sido gerado nos últimos 50 anos. Podemos considerar a existência de três gerações de sistemas de RI (Baeza-Yates & Ribeiro-Neto, 1999).

1. Primeira geração, quando os sistemas de RI consistiam basicamente de catálogos de cartões, contendo principalmente nome do autor e título dos documentos;

2. Segunda geração, quando ocorreram acréscimos nas funcionalidades de busca, permitindo pesquisa por assunto, por palavras-chave e outras consultas mais complexas; e

3. Terceira geração, quando o foco passa a ser a utilização de interfaces gráficas, de formulários eletrônicos, de características de hipertexto e de arquiteturas de sistemas abertos, como ocorre atualmente.

Com a utilização dos computadores e da Internet, cresce de forma cada vez mais rápida a quantidade dos textos armazenados em formato digital e também a quantidade dos que estão sendo disponibilizados para os usuários. Existe muito conhecimento implícito nesta documentação textual, o qual pode ser explorado através de diversas técnicas. Todavia, encontrar

a informação relevante é uma tarefa difícil. De acordo com (Gonzalez & Lima, 2003), o Processamento de Linguagem Natural (detalhado na Seção 2.2) está presente em diferentes níveis nas abordagens de RI para solucionar este problema. O conhecimento lingüístico pode, principalmente através de processamento morfossintático e semântico, trazer estratégias inteligentes para a Recuperação de Informação.

Desde então, os métodos de Recuperação de Informação começaram a ganhar importância, dentre eles as técnicas de conflação, que são de particular interesse para o presente trabalho e descritos na Seção 2.2.5.

Mais genericamente, a noção de encontrar padrões úteis, ou parte deles, em dados inicialmente sem tratamento e de qualquer natureza, tem recebido diversos outros nomes: Descoberta de Conhecimento em Bancos de Dados (*Knowledge Discovery in Databases* - KDD), Mineração de Dados (*Data Mining*), Extração de Conhecimento, Arqueologia de Dados e ainda Processo de Padronização de Dados. De fato, Recuperação de Informação juntamente com todas essas áreas de pesquisa são consolidadas e fazem parte de um grande campo denominado Mineração de Dados (*Data Mining*). O escopo deste trabalho está, porém, restrito a Recuperação de Informação.

Para embasar a discussão sobre técnicas de RI aplicadas à extração de palavras-chave, torna-se inicialmente necessário estudar alguns métodos oriundos do Processamento de Linguagem Natural, descritos a seguir.

2.2 Processamento de Linguagem Natural

De acordo com (Jackson & Moulinier, 2002), o termo **Processamento de Linguagem Natural** (PLN) é normalmente usado para descrever a função de softwares ou de componentes de hardware em um sistema computacional que analisam ou sintetizam linguagem falada ou escrita. O termo "natural" tem a função de diferenciar a fala e a escrita humana de linguagens mais formais, tais como notações matemáticas ou lógicas, ou ainda linguagens de computador, tais como Java, LISP e C++.

Existe ainda uma outra área relacionada ao PLN denominada Entendimento de

Linguagem Natural, que tem como objetivo permitir a um sistema computacional realmente compreender linguagem natural da mesma forma como um humano faz. Contudo, o escopo deste trabalho limita-se ao Processamento de Linguagem Natural.

Segundo (Jackson & Moulinier, 2002), são diversas as aplicações de PLN. Algumas aplicações são listadas abaixo:

- Recuperação de documentos: descoberta de documentos que são considerados relevantes para uma consulta do usuário. Segundo o autor, esta é a aplicação principal na Internet atualmente. Cabe observar que usuários fazendo buscas na Internet estão efetivamente executando recuperação de documentos. A tendência tem sido em direção à crescente sofisticação na indexação, identificação e apresentação de textos relevantes.
- Roteamento de documentos: processo automático segundo o qual documentos que se enquadram em determinado critério são diretamente enviados para o usuário interessado.
- Classificação de documentos: estratégia na qual documentos são associados a categorias baseadas no seu conteúdo, sendo que comumente um mesmo documento pode fazer parte de diversas categorias.
- Indexação de documentos: processo de geração de índices de palavras ou frases vinculadas aos documentos nas quais elas ocorrem.
- Extração de informações: não está relacionada com a localização de um documento, mas com a obtenção de informação específica contida em um ou mais documentos.
- Sumarização de documentos: é um caso particular de extração de informações, na qual é pretendido extrair sentenças que resumam um dado documento.

Para que as atividades citadas anteriormente sejam possíveis, é imprescindível compreender a estrutura da linguagem utilizada. Segundo (Gonzalez & Lima, 2003), o Processamento de Linguagem Natural trata computacionalmente os diversos aspectos da comunicação humana, como sons, palavras, sentenças e discursos, considerando formatos e referências, estruturas e significados, contextos e usos.

Em sentido amplo, pode-se dizer que o PLN visa fazer com que os sistemas

computacionais se comuniquem em linguagem humana, nem sempre em todos os níveis de entendimento e de geração de sons, palavras, sentenças e discursos. De acordo com (Gonzalez & Lima, 2003), tais níveis seriam os seguintes :

- Fonético e fonológico: que trabalham o relacionamento das palavras com os seus sons;
- Morfológico: abrange a construção das palavras a partir das unidades de significado primitivas e sua classificação em categorias morfológicas;
- Sintático: aborda o relacionamento das palavras entre si e como as frases podem ser partes de outras, construindo sentenças;
- Semântico: estuda o relacionamento das palavras com seus significados e como eles são combinados para formar os significados das sentenças;
- Pragmático: abrange o uso de frases e sentenças em diferentes contextos, afetando o significado.

Vale a pena descrever as estratégias de processamento morfossintático e semântico, que são diretamente relevantes para a proposta deste trabalho.

2.2.1 Processamento morfossintático

Fazem parte do processamento morfossintático, a análise morfológica e a análise sintática. A morfologia e a sintaxe tratam da constituição das palavras e dos grupos de palavras que formam os elementos de expressão de uma língua. Enquanto o analisador léxico-morfológico lida com a estrutura das palavras e com a classificação das mesmas em diferentes categorias, o analisador sintático trabalha no nível do agrupamento de palavras, analisando a constituição das frases.

2.2.2 Processamento semântico

Enquanto a sintaxe corresponde ao estudo de como as palavras se agrupam para formar estruturas de sentenças, a semântica está relacionada ao significado, não só de cada palavra, mas

também do conjunto resultante delas. O processamento semântico é considerado um dos grandes desafios do PLN, pois se vincula de um lado com a morfologia e a estrutura sintática e, de outro lado, com informações da pragmática (Saint-Dizier, 1999) *apud* (Gonzalez & Lima, 2003).

Apesar de todos os benefícios que o uso de processamento semântico traria para a extração de palavras-chave, no presente trabalho iremos nos limitar apenas ao processamento morfossintático por razões de viabilidade prática. Deve-se levar em conta que a análise semântica implica em um aumento na complexidade de implementação do algoritmo desejado, além de demandar uma base de conhecimentos estruturados sobre a área trabalhada. Particularmente, é difícil construir essa base, pois ela tem que ser específica para cada área de conhecimento.

Cabe reforçar que o objetivo deste trabalho é desenvolver um algoritmo de extração de palavras-chave para textos de qualquer domínio. Portanto, o uso de processamento semântico pode ser um aprimoramento das técnicas discutidas nesta dissertação, e alvo de possíveis trabalhos futuros.

2.2.3 Estratégia de processamento utilizada

A estratégia empregada nesta pesquisa estará restrita ao nível morfológico. A opção de trabalhar apenas neste nível se justifica pelos objetivos expostos no capítulo introdutório, de extrair palavras-chave automaticamente de textos em formato digital tendo como pressuposto que não há nenhuma informação adicional sobre o relacionamento semântico das palavras e sobre a área de conhecimento específico em que os documentos a serem processados se enquadram.

O PLN se apóia no conhecimento lingüístico e em métodos estatísticos, não necessariamente de forma excludente, para analisar a estrutura morfológica e sintática de um texto em linguagem natural. De acordo com (Bod, 1995) têm sido apontados benefícios quando há a associação de conhecimento lingüístico e métodos estatísticos.

Nas seções seguintes, faremos um breve levantamento das técnicas de PLN relevantes tanto na área de conhecimento lingüístico quanto na de métodos estatísticos.

2.2.4 Aplicação de conhecimento lingüístico

O conhecimento lingüístico pode ser explorado através de técnicas de etiquetagem de texto, normalização de variações lingüísticas e eliminação de *stopwords*[2].

2.2.4.1 Etiquetagem de texto

Quando um conhecimento lingüístico é considerado, a etiquetagem gramatical do texto é um dos passos iniciais. Conforme (Gonzalez & Lima, 2003), um etiquetador gramatical (*part-of-speech tagger*) é um sistema que identifica, através da colocação de uma etiqueta (*tag*), a categoria gramatical de cada item lexical do texto analisado. Segundo (Bick, 1998), enquanto um etiquetador morfológico inclui informações sobre categorias morfológicas, como substantivos e adjetivos, um etiquetador sintático acrescenta etiquetas indicando as funções sintáticas das palavras, como sujeito e objeto direto.

Além da etiquetagem ou marcação gramatical, existe a etiquetagem semântica (Vieira, 2000) *apud* (Gonzalez & Lima, 2003), que anexa informação relacionada ao significado, podendo indicar os papéis dos itens lexicais na sentença, como agente, processo e estado.

De forma genérica, o termo léxico significa uma relação de palavras com suas categorias gramaticais e seus significados. Em relação a uma determinada língua, um léxico é o universo de todos os seus itens lexicais, que seus falantes utilizam, já utilizaram ou poderão vir a utilizar (Scapini et al., 1995) apud (Gonzalez & Lima, 2003).

2.2.4.2 Normalização de variações lingüísticas

A normalização lingüística pode ser subdividida em três casos distintos (Arampatzis et al., 2000): morfológica, sintática e léxico-semântica.

2 De acordo com a literatura pesquisada na língua portuguesa, será usado o termo na língua inglesa e não uma tradução, pois não existe tradução consagrada.

A **normalização morfológica** ocorre quando há redução dos itens lexicais através de conflação a uma forma que procura representar classes de conceitos. Na Seção 2.2.5 será definido o que é conflação.

A **normalização sintática** ocorre quando há a normalização de frases semanticamente equivalentes mas sintaticamente diferentes, em uma forma única e representativa das mesmas, como fruta madura e saborosa e fruta saborosa e madura.

A **normalização léxico-semântica** ocorre quando são utilizados relacionamentos semânticos (sinonímia, hiponímia e meronímia)[3] entre os itens lexicais para criar um agrupamento de similaridades semânticas, identificado por um item lexical que representa um conceito único.

Novamente cabe lembrar que utilizaremos neste trabalho somente a normalização morfológica.

2.2.4.3 Eliminação de stopwords

Stopwords são palavras freqüentes em um texto e que não representam nenhuma informação de maior relevância para a extração de palavras-chave. Por exemplo: advérbios, artigos, conjunções, preposições e pronomes.

As *stopwords* não fornecem nenhuma contribuição na identificação do conteúdo do texto. A remoção das *stopwords* tem como objetivo eliminar palavras que não são representativas ao documento e isso conseqüentemente diminui o número de palavras a serem analisadas no mesmo, e também o número de palavras a serem armazenadas em uma base de busca de informações.

2.2.5 Aplicação de métodos estatísticos

3 **Sinonímia:** é a relação que se estabelece entre duas palavras ou mais que apresentam significados iguais ou semelhantes – sinônimos. **Hiponímia:** é a relação existente entre uma palavra de sentido mais específico e outra de sentido mais genérico, que tem com a primeira traços semânticos comuns (por exemplo, mamífero está numa relação de hiponímia com animal). **Meronímia:** é a relação de parte-todo; palavra que designa parte de outra (por exemplo, copa e aba são merônimos de chapéu).

Métodos estatísticos têm dado grande contribuição ao Processamento de Linguagem Natural, como são os casos da lei de Zipf e do gráfico de Luhn.

Zipf, em 1949, estabeleceu o que ficou conhecida como *constant rank-frequency law of* Zipf (Moens, 2000) *apud* (Gonzalez & Lima, 2003). Esta lei define que, tomando um determinado texto, o produto $\log(f_t) \times k_t$ é aproximadamente constante, onde f_t é o número de vezes que o termo *t* ocorre no texto e k_t é a posição deste termo em uma relação de todos os termos daquele texto, ordenados pela freqüência de ocorrência.

Por outro lado, Luhn sugeriu, em 1958, que a freqüência de ocorrência das palavras em um texto pode fornecer uma medida útil sobre a expressividade das mesmas (Frantz & Shapiro & Voiskunskii, 1997; Moens, 2000) *apud* (Gonzalez & Lima, 2003), pois o "autor normalmente repete determinadas palavras ao desenvolver ou variar seus argumentos e ao elaborar diferentes enfoques sobre o assunto que trata". As palavras com maior freqüência de ocorrência deveriam ser consideradas pouco expressivas porque este conjunto de palavras é composto normalmente por artigos, preposições e conjunções. Também as palavras que muito raramente ocorrem deveriam ser consideradas pouco expressivas justamente em razão da baixa freqüência. Sobram como expressivas as palavras com maior freqüência de ocorrência intermediária, como mostra a Figura 1.

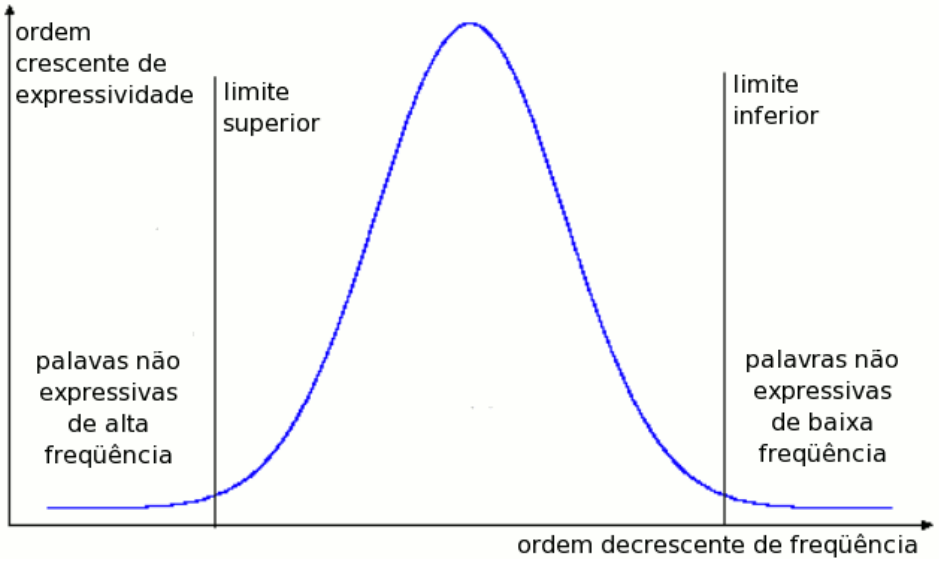

FIGURA 1 - Gráfico de Luhn. Adaptado de (Gonzalez & Lima, 2003)

A utilização da teoria de probabilidade e das abordagens estatísticas em geral, no Processamento de Linguagem Natural, apontam caminhos promissores para o entendimento do significado (Bod, 1995). A teoria da probabilidade é relevante por gerar modelos matematicamente precisos para freqüências de ocorrência, e as abordagens estatísticas, por permitirem suposições valiosas em casos de incertezas.

No âmbito da teoria da informação, a estatística provê mecanismos para indicar quanta informação ou quanta incerteza temos em relação a um evento, ou, ainda, qual o grau de associação de eventos que ocorrem simultaneamente (Krenn & Samuelson, 1997). Um desses mecanismos, a informação mútua, leva em conta a probabilidade de ocorrência dos eventos isolados e em conjunto. Quanto maior a probabilidade dos termos ocorrerem juntos, em relação às probabilidades de suas ocorrências isoladas, maior a informação mútua. Podem ser encontradas aplicações desta medida, por exemplo, para calcular o grau de similaridade entre palavras (Gauch & Futrele, 1994) e (Mandala & Tokunaga & Tanaka, 1999) *apud* (Gonzalez & Lima, 2003).

Os métodos estatísticos podem ser utilizados para auxiliar o PLN em diversas situações. Eles têm sido utilizados na etiquetagem gramatical, na resolução de ambigüidade e na aquisição de conhecimento lexical (Krenn & Samuelson, 1997), entre outras aplicações.

Em particular, vários algoritmos de extração automática de palavras-chave se amparam fortemente em métodos estatísticos, como será descrito em detalhes no Capítulo 3.

2.2.6 Conflação

Conflação é o ato de fusão ou combinação para igualar variantes morfológicas de palavras. A conflação pode ser manual, usando algum tipo de expressão regular, ou automática, via programas chamados radicalizadores (*stemmers*).

Para reduzir as variações de uma palavra para uma forma única utilizam-se técnicas de conflação. Segundo (Sparck-Jones & Willet, 1997) *apud* (Gonzalez & Lima, 2003), existem dois métodos principais de conflação para se obter tal redução: radicalização e redução à forma canônica (tratada por alguns autores como lematização).

2.2.6.1 Radicalização

Radicalização (*stemming*) é o processo de combinar as formas diferentes de uma palavra em uma representação comum, o radical (*stem*) (Orengo & Huyck, 2001). **Radical** é o conjunto de caracteres resultante de um processo de radicalização. Este não é necessariamente igual à raiz lingüística, mas permite tratar variações diferentes de uma palavra da mesma forma. Por exemplo, conector e conectores são essencialmente iguais, mas sem sofrerem a redução por radicalização serão tratadas como palavras distintas.

Na língua inglesa, *stem* é o mesmo que radical ou tema. Raiz seria *root*. Assim, segundo (Oxford, 1996): *Stem – 4. Gram. The root or main part of a noun, verb, etc. to which inflections are added; the part that appears unchanged throughout the cases and derivatives of a noun, persons of a tense, etc.*

Conforme (American, 1991): *Stem – 5. The main part of a word to which affixes are added.*

Freqüentemente, é especificada uma palavra em uma consulta, mas somente uma variante desta palavra é apresentada em um documento relevante. Plurais, formas de gerúndio e sufixos de tempos verbais são exemplos de variações sintáticas que impedem uma perfeita combinação entre uma palavra de consulta e uma palavra do respectivo documento. Por exemplo, as palavras durabilidade, duradouro e durável poderiam ser reduzidas para a representação comum dur-.

Este método é amplamente usado em processamento de textos para Recuperação de Informação baseado na suposição de que uma consulta com o termo durável implica num interesse em documentos que contenham também as palavras durabilidade e duradouro. Este problema é solucionado com a substituição de palavras pelos seus respectivos radicais.

De acordo com (Sparck-Jones & Willet, 1997) *apud* (Gonzalez & Lima, 2003), radicalização consiste em reduzir todas as palavras ao mesmo radical, por meio da retirada dos afixos (sufixos e prefixos) da palavra. Segundo (Porter, 1980), radicalização é o processo de remoção das terminações morfológicas e flexionais das palavras. O algoritmo de Porter foi desenvolvido especificamente para a língua inglesa e tem a característica de remover apenas os sufixos das palavras.

Conforme tais definições, optou-se por utilizar a segunda definição, pois foi considerado que o sentido de algumas palavras se altera quando é adicionado um prefixo às mesmas. Por exemplo, limitado e ilimitado.

Os algoritmos de radicalização são tradicionalmente utilizados em Recuperação de Informação com o objetivo de melhorar a abrangência dos resultados da busca. Isso parte da constatação de que as palavras que aparecem em documentos e consultas freqüentemente têm muitas variantes morfológicas. Por exemplo, as palavras casa e casarão não serão reconhecidas como equivalentes sem a aplicação de uma técnica de radicalização.

Existem vários estudos que avaliam a importância do processo de radicalização para a área de Recuperação de Informação e que chegaram a conclusões distintas:

Em (Harman, 1991) *apud* (Orengo & Huyck, 2001), foram examinados os efeitos de três algoritmos sobre três coleções de teste e não foram encontradas melhorias no desempenho de recuperação, visto que o número de consultas com desempenho melhor tende a ser igual ao número com desempenho pior. Entretanto, em (Krovetz, 1993) *apud* (Orengo & Huyck, 2001), o uso de radicalização melhorou o desempenho de recuperação no máximo em 35% de algumas coleções. Finalmente, depois de uma análise exaustiva, (Hull, 1996) *apud* (Orengo & Huyck, 2001) concluiu que algumas formas de radicalização são quase sempre proveitosas. Este autor determinou que a melhora geral da qualidade da recuperação foi de apenas 1-3%, mas para muitas consultas individuais o uso de radicalização fez uma grande diferença.

Todos esses experimentos foram realizados com coleções da língua inglesa e parece possível que linguagens altamente flexíveis como a língua portuguesa possam beneficiar-se mais da aplicação de técnicas de radicalização.

Dois erros que costumam ocorrer durante o processo de radicalização são: *overstemming* e *understemming*.

- ***Overstemming:*** ocorre quando a parte removida da palavra não é um sufixo, mas parte do seu radical. Isto pode fazer com que palavras não relacionadas sejam combinadas. Por exemplo, a palavra confortável após ser processada por um radicalizador, é transformada no radical confor-. Neste caso, o radicalizador removeu parte do radical correto, a saber confort-.
- ***Understemming:*** ocorre quando um sufixo não é removido completamente. Isto pode fazer com que palavras relacionadas não sejam combinadas. Por exemplo, a palavra referência, é transformada no radical referênc-, ao invés de ser transformada no radical correto refer-.

2.2.6.2 Redução à Forma Canônica

Redução à forma canônica é o ato de representar as palavras através do infinitivo dos verbos e masculino singular dos substantivos e adjetivos. Redução à forma canônica é tratada por alguns autores como lematização. A palavra reduzida desta forma recebe a denominação de **lema** ou **forma canônica**.

Essa representação gráfica das palavras ocorre nos dicionários, pois todas as ocorrências de uma palavra são reunidas sob uma única forma, em vez de apresentá-las tal como aparecem nos textos, com variações no gênero, no número ou na grafia.

De acordo com (De Lucca & Nunes, 2002), redução à forma canônica difere fundamentalmente de radicalização. Enquanto redução à forma canônica existe puramente no contexto lexicográfico, radicalização não. Assim, as estruturas são diferentes, embora eventualmente possam ser graficamente semelhantes.

No caso da redução à forma canônica, não há perda da categoria morfológica original, ao contrário de um radical que pode ser oriundo de palavras de categorias diferentes. Por exemplo, se as palavras salvando e meninas fossem radicalizadas, estas seriam transformadas nos respectivos radicais salv- e menin- e se as mesmas palavras fossem reduzidas à forma canônica seriam representadas como salvar e menino, respectivamente.

3 ALGORITMOS DE EXTRAÇÃO DE PALAVRAS-CHAVE

Este capítulo apresenta quatro algoritmos de extração de palavras-chave e algumas considerações relevantes.

A primeira seção discute o uso de palavras-chave e tece comentários sobre duas abordagens para encontrar palavras-chave a partir de documentos.

As seções seguintes descrevem os algoritmos GenEx, EPC-P e EPC-R, mostrando uma breve comparação entre os dois últimos.

A última seção descreve detalhadamente o algoritmo KEA, o qual é o foco principal desta dissertação.

3.1 Considerações sobre palavras-chave

Palavras-chave (*keywords* ou *keyphrases*)[4] fornecem um breve resumo do conteúdo de um documento. À medida que grandes coleções de documentos tais como bibliotecas digitais tornam-se populares, o valor dessas informações resumidas aumentam.

Palavras-chave são particularmente úteis porque podem ser interpretadas individualmente e independentemente umas das outras. Elas podem ser usadas em sistemas de Recuperação de Informação como descrições de documentos retornados por uma pesquisa, como

4 Ao longo deste documento, usa-se o termo "palavras-chave" (*keywords*) para indicar "frases-chave" (*keyphrases*). O segundo seria o termo mais apropriado, visto que palavras-chave não são compostas necessariamente por palavras isoladas e sim por frases nominais ou verbais. Contudo, preferiu-se utilizar o termo comumente adotado na literatura.

itens dos índices de busca, como caminhos para navegação de uma coleção e como referências para agrupamento de documentos.

Palavras-chave são tradicionalmente escolhidas de forma manual, o que exige certo esforço. Então, técnicas de extração automática são potencialmente de grande utilidade. Indexadores profissionais freqüentemente escolhem frases de um vocabulário controlado, pré-definido e relevante ao domínio em questão. Entretanto, a grande maioria dos documentos utilizados no dia-a-dia não possuem palavras-chave e associá-las manualmente é um processo trabalhoso, que ainda requer conhecimento do assunto tratado.

Em muitos contextos acadêmicos, os próprios autores determinam palavras-chave para documentos escritos por eles. Segundo (Pereira & Nunes, 2001), acrescenta-se a esse cenário o fato de que nem sempre as palavras-chave do autor expressam corretamente os tópicos compreendidos pelo artigo. É comum que uma certa porcentagem das palavras-chave identificadas pelos autores dos artigos sequer apareça no texto, ou apareça com baixa freqüência, fazendo com que métodos estatísticos baseados em freqüência de ocorrência não consigam selecioná-las. Isso pode ocorrer por várias razões:

- o autor pode preferir adotar como palavra-chave um termo mais geral;
- em um artigo científico na língua portuguesa, ele pode optar por usar como palavra-chave o termo mais conhecido na língua inglesa, e no texto, usa sua tradução; ou
- o autor tem a possibilidade de escolher as palavras-chave de forma eventualmente subjetiva e aleatória.

Mesmo sendo pouco determinísticos, no entanto, é desejável que esses critérios se traduzam em valores numéricos, para uma fácil comparação. A maioria dos trabalhos similares em Recuperação de Informação usa as medidas complementares de precisão (*precision*) e recuperação (*recall*) para medir o desempenho de seus algoritmos. A medida de **precisão** fornece a relação entre o número de documentos recuperados relevantes e o total de documentos obtidos, ou seja, indica a utilidade da busca feita. Por outro lado, **recuperação** fornece a relação entre o número de documentos relevantes recuperados e o total de documentos relevantes que existem, ou seja, indica a capacidade da busca de encontrar tudo que é importante para o assunto da busca

realizada.

Para a língua inglesa, vários métodos foram propostos para geração ou extração de informação resumida de textos (por exemplo (Kupiec & Pedersen & Chen, 1995), ((Brandow & Kitze & Rau, 1994), (Johnson et al., 1993)) *apud* (Witten et al., 1999). Em particular um dos trabalhos mais relevantes para a língua inglesa é o GenEx (Turney, 1999), descrito na Seção 3.2.

Para a língua portuguesa, ainda não existem muitos trabalhos nesta área. Foi encontrado apenas um trabalho sobre este assunto (Pereira & Nunes, 2001). O trabalho apresenta o projeto, a implementação, a comparação e a análise de dois algoritmos de extração de palavras-chave de textos na língua portuguesa. O primeiro, EPC-P, de concepção dos autores, é baseado na freqüência de determinados padrões morfossintáticos obtidos de um conjunto (*corpus*) de artigos científicos. O segundo, EPC-R, é uma adaptação do algoritmo concebido para a língua inglesa, denominado Extractor e descrito em [Turney, 1997; Turney, 1999], e se baseia nas freqüências de radicais. Ambos serão apresentados de forma mais detalhada na Seção 3.3. Conforme a conclusão dos autores, os algoritmos ainda precisam ser melhorados, ou seja, não tiveram um desempenho satisfatório.

Por outro lado, em (Witten et al., 1999) foi proposto um algoritmo, denominado KEA, de extração automática de palavras-chave e computacionalmente muito mais simples e eficiente do que os três citados anteriormente (GenEx, EPC-P e EPC-R). Adicionalmente, esse algoritmo é baseado em Aprendizado de Máquina (usando métodos estatísticos) e possui uma implementação disponível como Software Livre, que foi utilizada como base para o desenvolvimento desta dissertação. Esta implementação original está acessível a partir do projeto da Biblioteca Digital da Nova Zelândia (NZDL, 2004). Entretanto, o algoritmo KEA foi elaborado apenas para a língua inglesa. Em decorrência destes fatores, optou-se por adaptá-lo para a língua portuguesa. O algoritmo original está descrito na Seção 3.4 e no Capítulo 4 estão descritas as alterações feitas que possibilitam a extração de palavras-chave de documentos na língua portuguesa.

Aprendizado de Máquina (AM) é uma subárea de pesquisa muito importante em Inteligência Artificial (IA), pois a capacidade de aprender é essencial para um comportamento inteligente. AM estuda métodos computacionais para adquirir novos conhecimentos, novas habilidades e novos meios de organizar o conhecimento já existente (Mitchell, 1997).

O campo de Aprendizado de Máquina emprega algoritmos computacionais que melhoram automaticamente através de sua utilização, adquirindo experiência a partir de uma série de exemplos. Tipicamente isso ocorre em duas etapas. A primeira, onde é feito o treinamento, e a segunda, onde ocorre a classificação.

Treinamento é a etapa onde o sistema é alimentado com um conjunto de testes previamente classificados e organizados. O resultado desta etapa é um modelo de dados que permite executar a segunda etapa.

Classificação é a etapa onde o modelo construído é utilizado para avaliar automaticamente novos documentos, classificando-os sem intervenção humana. Os métodos de classificação são os procedimentos que efetivamente classificam o documento em determinada classe.

Existem duas abordagens fundamentalmente diferentes para encontrar palavras-chave em textos. Ambas usam métodos de Aprendizado de Máquina e requerem para propósito de treinamento um conjunto de documentos com palavras-chave já marcadas. As abordagens são as seguintes:

- **Determinação de palavras-chave:** procura selecionar as frases de um vocabulário controlado que descrevam melhor um documento. Os dados de treinamento associam um conjunto de documentos com cada frase no vocabulário, construindo assim um classificador para cada frase. Um novo documento é processado por cada classificador e associado às palavras-chave dos modelos que o classificam como positivo (por exemplo (Dumais et al., 1998)). Note que as palavras-chave que podem ser determinadas são apenas aquelas que já foram vistas nos dados de treinamento.
- **Extração de palavras-chave:** é a abordagem usada pelo algoritmo KEA. Esta abordagem não usa um vocabulário controlado e sim escolhe palavras-chave do próprio texto. Esta emprega técnicas de análise léxica e de Recuperação de Informação para extrair frases de um documento texto que são candidatas prováveis de caracterizá-los adequadamente (Turney, 1999). Nesta abordagem, os dados de treinamento são usados para ajustar os parâmetros do algoritmo de extração que ocorrerá justamente na etapa de classificação de novos documentos.

3.2 Algoritmo GenEx

O algoritmo GenEx (Turney, 1997) possui dois componentes, o algoritmo genético denominado Genitor e descrito em (Whitley, 1989) e o algoritmo de extração de palavras-chave, Extractor (Nrc, 2004).

O Extractor é um algoritmo de extração de palavras-chave por medição de freqüência de radicais no texto. O algoritmo conta quantas vezes ocorrem seqüências de palavras (simples, duplas ou trios) no texto, utilizando-se de métodos estatísticos e de listas de *stopwords* para decidir quais são mais relevantes.

O algoritmo Extractor recebe um documento como entrada e produz uma lista de palavras-chave como saída. O Extractor tem doze parâmetros que determinam como processar o texto de entrada. No GenEx, os parâmetros do Extractor são ajustados pelo algoritmo genético Genitor, para aumentar a adequação (*fitness*) aos dados de treinamento. O Genitor é usado para direcionar o Extractor, mas o Genitor torna-se desnecessário uma vez que o processo de treinamento esteja completo. Deste modo, o sistema de aprendizagem é chamado GenEx (Genitor mais Extractor) e o sistema treinado é chamado simplesmente Extractor (GenEx menos Genitor).

Para o nosso objetivo o GenEx apresenta uma série de limitações. Primeira, o Extractor foi projetado especificamente para a língua inglesa. Segunda, o Extractor é um processo bastante complexo e caro computacionalmente. Terceira, o algoritmo Extractor é patenteado, o que implica em limitações para uso em pesquisas. Quarta, é necessária uma segunda ferramenta, que é o Genitor, apenas para ajustar os parâmetros, o que aumenta a complexidade de todo o processo.

Na seção seguinte, serão apresentados dois algoritmos de extração de palavras-chave de textos na língua portuguesa e a análise do desempenho de cada um.

3.3 EPC-P e EPC-R

Em (Pereira & Nunes, 2001), foram discutidas a implementação e a análise de dois algoritmos de extração de palavras-chave de textos na língua portuguesa, utilizando técnicas

extrativas distintas.

O primeiro algoritmo é o EPC-P (Extrator de Palavras-Chave por freqüência de Padrões). O segundo é o EPC-R (Extrator de Palavras-Chave por freqüência de Radicais), baseado no algoritmo Extractor citado anteriormente.

3.3.1 O algoritmo EPC-P

De acordo com (Pereira & Nunes, 2001), para adquirir um levantamento de padrões morfossintáticos (combinações de categorias gramaticais) das palavras-chave elaboradas pelos autores, foi realizado um levantamento em 12 exemplares de revistas científicas brasileiras da área de Computação, formando um conjunto de 58 artigos na língua portuguesa. Os 58 artigos analisados possuíam palavras-chave definidas pelos seus respectivos autores. Os padrões mais freqüentemente encontrados foram:

- nome;
- nome preposição nome;
- nome adjetivo;
- nome adjetivo adjetivo;
- nome adjetivo preposição nome; e
- nome preposição nome adjetivo.

onde "nome" trata-se de um nome próprio ou um substantivo comum.

Após o levantamento de todas as frases que se encaixam nos padrões, utilizaram-se métodos estatísticos para definir, dentre estas, quais seriam as de maior relevância, podendo ser consideradas palavras-chave para o texto.

Como o EPC-P necessita das classes gramaticais das palavras do texto para detectar os padrões citados, ele não trabalha sobre o texto original, mas sobre um texto etiquetado, onde todas as palavras aparecem associadas às suas categorias gramaticais. O texto precisa, portanto, ser pré-processado por um etiquetador (*Part-of-Speech Tagger*) da língua portuguesa, no qual foi utilizado o descrito em (Aires et al., 2000).

O texto etiquetado é percorrido em sua totalidade e são construídas seis listas, cada uma contendo todas as seqüências do texto (em ordem alfabética) que se encaixam em um dos seis padrões procurados (definidos anteriormente), assim como o número de vezes que cada uma das mesmas ocorrem no texto. Também é armazenado o número de vezes que cada padrão se repete.

Paralelamente à construção dessas seis listas, são construídas mais seis listas, uma para cada padrão, sendo que nestas são inseridos somente os radicais das palavras. Foi usado o radicalizador de Porter (Porter, 1980), em versão adaptada para a língua portuguesa. Segundo os autores, foi feita uma adaptação que desconsidera os radicais das preposições, por serem de classe fechada e não carregarem conteúdo semântico.

Feito o percurso do texto, ordenam-se as seis listas de radicais, deixando-as em ordem decrescente em relação ao número de ocorrências de cada palavra no texto, enquanto que as seis listas que contêm as palavras originais continuam ordenadas alfabeticamente.

Terminada a fase de construção das 12 listas, inicia-se a construção da lista de palavras-chave. Primeiramente, cria-se uma lista com o nome de `lista1`. Esta lista é preenchida da seguinte forma:

- mantém-se um ponteiro associado ao primeiro elemento de cada uma das seis listas de radicais; para cada um dos seis elementos analisados, divide-se o número de ocorrências pelo número de ocorrências do padrão em que ele se encontra (chamado de freqüência relativa);
- o radical que tiver a maior freqüência relativa (e que ocorrer pelo menos duas vezes no texto) será inserido na `lista1`, justamente com um marcador para indicar de qual lista ele foi retirado;
- repete-se o processo até que a `lista1` possua 50 elementos (este número foi escolhido por se mostrar suficiente para a obtenção das 30 palavras-chave finais, no próximo passo do algoritmo).

Depois de construída a `lista1`, inicia-se a construção da lista final de palavras-chave, a qual recebe o nome de `lista2`. A `lista2` é preenchida da seguinte maneira:

- obtém-se o primeiro elemento da lista de radicais de nomes;

- pega-se a primeira ocorrência deste radical na `lista1`, caso ele ocorra na mesma. Desta forma, são consideradas as primeiras palavras da lista de nomes como sendo as mais relevantes para o texto, encontrando-se a melhor ocorrência de cada uma delas considerando todos os padrões;
- insere-se o radical encontrado na `lista2`, junto com o marcador que indica a classe a qual este elemento pertencia;
- pega-se o próximo elemento da lista de radicais de nomes;
- repete-se o processo (inserindo um novo radical na `lista2` sempre que ele ainda não pertencer à mesma) até que a `lista2` tenha 30 elementos, ou que acabe a lista de radicais de nome, o que faz com que a lista de palavras-chave não tenha necessariamente 30 elementos.

Considerando que os autores dos textos costumam definir de 3 a 6 palavras-chave para seus artigos, (Pereira & Nunes, 2001) optou inicialmente por gerar 15 palavras-chave para cada artigo analisado. Porém, para que fosse feita uma análise mais ampla do desempenho do algoritmo, este número foi ampliado para 30.

Segundo os autores, a `lista1` poderia ser considerada a lista final de palavras-chave, porém possuindo algumas imperfeições. A principal delas é que acabam por coexistir termos muito parecidos, que poderiam ser resumidas a apenas um item. Por exemplo, foram obtidas na `lista1` em um dos testes as frases projeto cooperativo na Internet, participação em projetos cooperativos, projeto cooperativo. Com o método utilizado para a criação da `lista2`, projeto cooperativo na Internet foi considerada a melhor forma em que aparece o radical projet- e, portanto, a única relevante entre as três.

Finalmente, precisa-se recuperar as palavras originais da `lista2`, já que a mesma é composta apenas por radicais. Para isto, basta para cada elemento percorrer a lista de palavras originais correspondente ao padrão em que o mesmo se encaixa, recuperando a melhor ocorrência deste radical na lista. Com isto, tem-se uma lista com até 30 palavras-chave para o texto.

3.3.2 O algoritmo EPC-R

Segundo (Pereira & Nunes, 2001), os bons resultados do algoritmo Extractor para a língua inglesa (Turney, 1997) motivaram a sua utilização como base para construir um extrator de palavras-chave por freqüência de radicais para a língua portuguesa (EPC-R).

Como já foi falado anteriormente, o Extractor é parametrizado por uma série de valores, que são obtidos por meio de algoritmos de Aprendizado de Máquina. Segundo citam os autores, não era intenção perseguir o mesmo caminho, então decidiu-se utilizar os parâmetros tal como definidos em ['Turney-1997], mesmo sabendo que tais valores carregam consigo uma dependência do conjunto de documentos a partir do qual foram determinados.

Por outro lado, parâmetros como `FIRST_LOW_THRESH` (que representa até que posição do texto uma palavra deve correr pela primeira vez para ser considerada relevante) e `FIRST_HIGH_THRESH` (que representa a partir de quando uma palavra deve correr pela primeira vez para ser considerada uma ocorrência tardia) foram alterados, para se acomodarem melhor aos padrões de tamanho dos textos na língua portuguesa. Para obtenção dos mesmos, (Pereira & Nunes, 2001) realizou uma pesquisa em um conjunto de artigos científicos, calculando o tamanho médio das introduções dos artigos (supondo que se uma palavra é relevante para o texto, ela deve aparecer na introdução). Os autores chegaram aos valores: 450 e 800, respectivamente. Segundo eles, isso significa que palavras relevantes tendem a ocorrer pela primeira vez até a posição 450, e que palavras que apenas ocorrem pela primeira vez após a posição 800 podem ser consideradas irrelevantes.

Outra diferença entre o Extractor e o EPC-R decorre da utilização das listas de *stopwords*. No EPC-R, são utilizadas duas listas de *stopwords*, para que uma consideração especial fosse feita quanto às preposições. Adicionalmente, o Extractor faz o uso de uma lista de verbos mais freqüentes da língua inglesa (uma vez que não trabalha com etiquetas morfossintáticas, o algoritmo precisa reconhecer essa categoria de outra forma). Segundo os autores do EPC-R, foi decidido não proceder da mesma forma para a língua portuguesa, uma vez que um algoritmo de reconhecimento morfológico das formas verbais não seria simples e os desviaria de seus objetivos. Essa diferença, no entanto, teve um impacto sensível no desempenho do EPC-R.

Conforme (Pereira & Nunes, 2001), o EPC-R não precisou das classes gramaticais das palavras do texto, apenas foi necessário fazer algumas modificações no texto original, separando as pontuações das palavras do texto, para que cada palavra fosse identificada com uma maior facilidade.

Com o texto adaptado em mãos, este é percorrido em sua totalidade e três listas são construídas. A primeira contém todas as palavras simples do texto, a segunda, todas as duplas e a terceira, todos os trios de palavras do texto (em ordem alfabética). As listas armazenam também o número de vezes que cada uma das palavras ocorre no texto.

Como neste método não se tem conhecimento algum sobre a classe gramatical de cada palavra, tornou-se necessário utilizar uma lista contendo as *stopwords* mais freqüentes, para que, conforme o texto seja percorrido, torne-se possível que palavras sem importância sejam descartadas.

De fato, os autores utilizam duas listas de *stopwords*, uma delas contendo as preposições junto com as demais palavras sem conteúdo semântico, e a outra sem as preposições. Segundo os mesmos, foram utilizadas essas duas listas para que a tripla de palavras pudesse conter preposições como palavra do meio. Por exemplo, gerenciamento de software seria uma candidata válida, mas de software educacional não seria, pois a preposição de ocorre no começo da tripla. Em resumo, sempre descartam-se as preposições, a não ser que ocorram como ligações entre duas palavras.

Paralelamente à construção destas três listas, são construídas mais três listas (correspondentes às listas de simples, duplas e trios), sendo que nestas são inseridos somente os radicais das palavras, que são também obtidos pelo radicalizador de Porter.

Feito o percurso do texto todo, ordena-se cada uma das três listas de radicais, deixando-as na forma decrescente de freqüência, enquanto que as três listas que contêm as palavras originais continuam ordenadas alfabeticamente.

Terminada a fase de construção das seis listas, é iniciada a construção da lista de palavras-chave. Primeiramente, cria-se a `lista1` com os 50 elementos de maior freqüência das três listas de radicais. A `lista2` é então preenchida da seguinte maneira:

- pega-se o primeiro elemento da lista de radicais simples;
- obtém-se a primeira ocorrência deste radical na `lista1`, se é que ele ocorre na mesma (desta forma, são consideradas as primeiras palavras da lista de radicais simples como sendo as mais relevantes para o texto, e encontrada a melhor ocorrência de cada uma delas considerando todas as três listas);
- insere-se o radical encontrado na `lista2`;
- pega-se o próximo elemento da lista de radicais simples;
- repete-se o processo (inserindo um novo radical na `lista2` sempre que ele ainda não pertencer à mesma) até que a `lista2` tenha 30 elementos, ou que acabe a lista de radicais simples (o que faz com que a lista de palavras-chave não tenha necessariamente 30 elementos).

Como no caso do EPC-P, a `lista1` poderia ser considerada a lista final de palavras-chave, mas nela ainda podem coexistir frases redundantes.

Finalmente, recuperam-se as palavras originais da `lista2`, já que esta é composta apenas por radicais. Segundo os autores, isto é um processo simples, uma vez que para cada elemento da `lista2` basta contar por quantos radicais ele é composto para descobrir a que lista ele pertencia. Basta então, para cada elemento, percorrer a lista de palavras originais correspondente ao padrão em que o mesmo se encaixa, recuperando a melhor ocorrência deste radical na lista. Com isto, tem-se uma lista com até 30 palavras-chave para o texto.

3.3.3 Comparação entre EPC-P e EPC-R

Segundo (Pereira & Nunes, 2001), o algoritmo EPC-P faz uma análise da freqüência de determinados padrões morfossintáticos no texto, para decidir quais palavras podem ser utilizadas para representar o tema central do mesmo. Um dos problemas encontrados na sua utilização foi que, por estar preso aos padrões, acaba por não encontrar outras construções interessantes, mas fora dos padrões (sendo, por exemplo, escolhidas como palavras-chave pelo autor). Um lado positivo de sua utilização foi que, também por estar preso aos padrões, acaba desconsiderando

termos que certamente não têm importância, como verbos, numerais e pronomes. Segundo os autores, uma perspectiva para a sua melhoria seria a utilização de mais padrões, ou de recursos para verificação de sinônimos, como um *thesaurus*.

Por outro lado, o EPC-R faz uma análise da freqüência de radicais (simples, duplas ou trios) no texto, em detrimento à utilização de padrões. Uma das limitações é o fato de estar restrito à busca de, no máximo, três radicais. Outra provém do fato de não estar preso aos padrões, acabando por considerar relevantes alguns termos que deveriam ser descartados, sendo o principal problema o tratamento dos verbos. Em contrapartida, isso permitiu aceitar uma variedade maior de padrões de palavras-chave. Ainda, segundo os autores, uma perspectiva para a sua melhoria seria a possibilidade de busca de uma seqüência maior de radicais (com quatro ou mais termos), assim como a utilização de outras técnicas adicionais como a verificação de verbos, numerais, pronomes, entre outros, além da verificação de sinônimos.

Avaliando os resultados tabulados em (Pereira & Nunes, 2001), pode-se dizer que a efetividade é relativamente baixa, pois apenas 23% das palavras-chave foram encontradas em média. Além disso, apesar da relativa simplicidade dos algoritmos (comparados com o GenEx), uma desvantagem é que não ocorre Aprendizado de Máquina, portanto ambos algoritmos não são capazes de se adaptarem a estruturas diferentes de documentos.

3.4 Algoritmo KEA

Neste capítulo será apresentado o algoritmo de extração automática de palavras-chave (Keyphrase Extraction Algorithm – KEA), que foi proposto em (Witten et al., 1999) por Ian H. Witten, Gordon W. Paynter e Eibe Frank, da Universidade de Waikato, Nova Zelândia; por Carl Gutwin, da Universidade de Saskatchewan, Canadá; e Craig G. Nevill-Manning, da Rutgers University, dos EUA.

O KEA é um algoritmo para extrair automaticamente palavras-chave de textos. Para isto, ele identifica palavras-chave candidatas usando métodos de análise léxica, calculando valores característicos para cada candidata e usando a técnica de Aprendizado de Máquina Naïve-Bayes para treinamento e extração de palavras-chave.

A técnica de Aprendizado de Máquina constrói um modelo de predição usando documentos de treinamento com palavras-chave conhecidas e então usa o modelo construído para encontrar palavras-chave em novos documentos, ou seja, em documentos cujas palavras-chave não são conhecidas. Os autores usaram uma grande massa de teste para avaliar a eficácia do KEA em termos de quantas palavras-chave determinadas pelo autor são identificadas corretamente.

Exemplos de resultados do KEA estão ilustrados na Tabela 1. Nesta tabela são apresentados os títulos de três artigos de pesquisas e dois conjuntos de palavras-chave para cada artigo. Um conjunto mostra as palavras-chave determinadas pelo autor; o outro as que foram determinadas automaticamente a partir do texto completo dos artigos. Frases comuns entre os dois conjuntos são colocadas em negrito.

TABELA 1 - Exemplos de saída do KEA. Adaptado de (Witten et al., 1999).

Protocols for secure, atomic transaction execution in eletronic commerce		Neural multigrid for gauge theories and other disordered systems		Proof nets, garbage, and computations	
anonymity	**atomicity**	disordered-systems	disordered gauge	**cut-elimination**	cut
atomicity	**auction**	**gauge fields**	**gauge fields**	linear logic	**cut-elimination**
auction	customer	**multigrid**	interpolation kernels	**proof nets**	garbage
eletronic	**eletronic**	neural multigrid	lenght scale	sharing graphs	**proof net**
commerce	**commerce**	neural networks	**multigrid**	typed lambda-calculus	weakening
privacy	intruder		smooth		
real-time	merchant				
security	protocols				
transaction	**security**				
	third party				
	transaction				

De acordo com (Witten et al., 1999), as palavras-chave determinadas pelo autor e as palavras-chave extraídas automaticamente são bastante similares, mas não é muito difícil adivinhar quais são as palavras-chave dos autores. Pode-se verificar que o KEA escolhe diversas palavras-chave boas, mas também escolhe algumas que são improváveis dos autores usarem, por

exemplo *gauge*, s*mooth* e especialmente *garbage* (lixo, que poderia até ser considerada ofensiva). Apesar destas anomalias, as listas de palavras-chave extraídas automaticamente fornecem uma descrição adequada dos três artigos. No caso em que nenhuma palavra-chave especificada pelo autor estiver disponível, as escolhas do KEA poderiam ser um recurso valioso para alguém encontrar estes três artigos na primeira busca.

Baseado em (Witten et al., 1999), o objetivo dos autores do KEA era produzir metadados poderosos que ainda não existem. Apesar deles terem avaliado o desempenho do KEA com as próprias palavras-chave dos autores, não era esperado igualar-se a elas. Se pudessem extrair resumos razoáveis a partir de textos dos documentos, já estariam fornecendo uma ferramenta valiosa para os projetistas e usuários de bibliotecas digitais.

O restante deste capítulo descreve em detalhes o algoritmo KEA. A próxima seção detalha o projeto do algoritmo. A seguir, serão feitos comentários sobre diversos experimentos projetados para testar a efetividade do KEA e explorar os efeitos da variação dos parâmetros no processo de extração.

3.4.1 Etapas

O algoritmo KEA têm dois estágios:

1. **Treinamento:** é criado um modelo para identificar palavras-chave usando documentos de treinamento onde as palavras-chave dos autores são conhecidas.
2. **Extração:** são extraídas palavras-chave de um novo documento usando o modelo de treinamento construído anteriormente.

O processo é ilustrado na Figura 2. Ambos os estágios escolhem um conjunto de frases candidatas a partir da entrada de seus documentos, e então calculam os valores de certos atributos, chamado de características, para cada frase candidata.

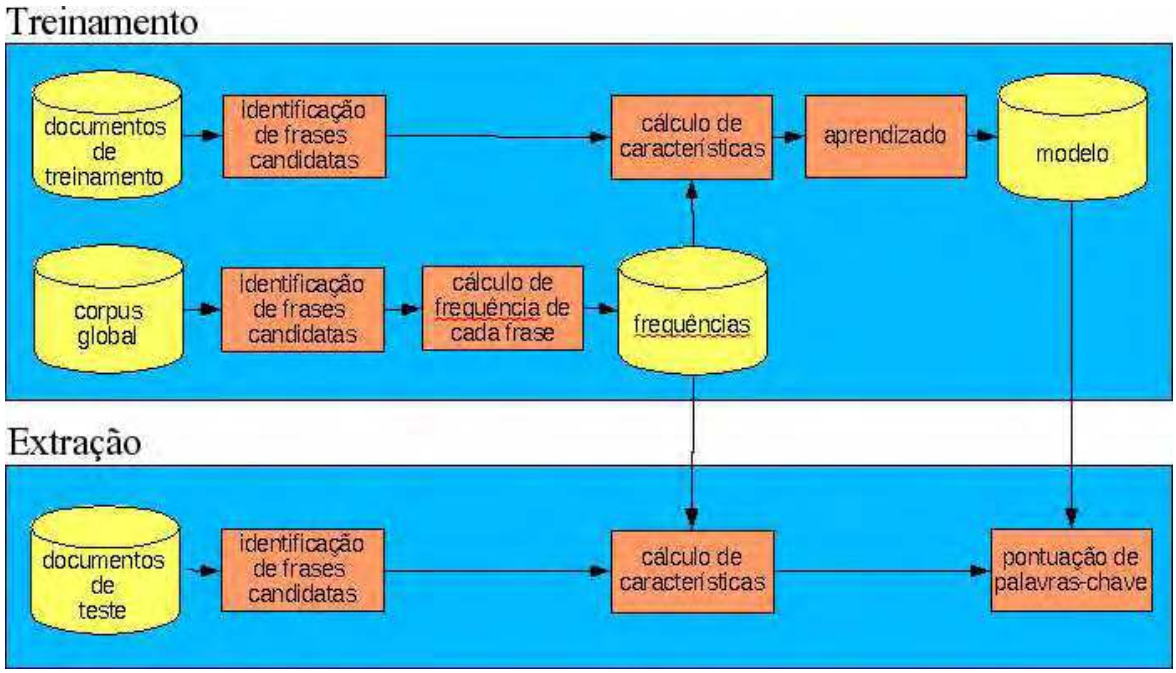

FIGURA 2 - Processos de treinamento e extração. Adaptado de (Witten et al., 1999).

3.4.1.1 Identificação de frases candidatas

O KEA escolhe frases candidatas em três passos. Ele primeiro limpa o texto de entrada, depois identifica as frases candidatas e finalmente faz e aplica o processo de radicalização e *case-folding*[5] nas frases.

3.4.1.1.1 Limpeza da entrada

Arquivos de entrada são filtrados para regularizar o texto e determinar os limites das frases. O fluxo de entrada é dividido em **termos** (seqüências de letras, dígitos e pontos que estão dentro da palavra), e então várias modificações são feitas:

- Sinais de pontuação, colchetes, parênteses, chaves e números são substituídos por limites de frases;
- Apóstrofos são removidos;

5 É o processo de converter todas as letras para maiúsculas ou para minúsculas.

- Palavras hifenizadas são divididas em duas;
- Caracteres restantes que não são válidos para termos são apagados, da mesma forma que qualquer termo que não contenha letras.

O resultado é um conjunto de linhas, cada uma com uma seqüência de termos contendo pelo menos uma letra. Abreviações contendo pontos, como *M12.4* são mantidos como um único termo.

3.4.1.1.2 Obtenção de frases candidatas

O KEA considera todas as subseqüências em cada linha e determina quais destas são frases candidatas adequadas. Foram investigados pelos autores vários métodos para determinação de adequação, tais como olhar para frases substantivas, mas foi encontrado que as seguintes regras são tanto simples quanto eficazes:

- Frases candidatas estão limitadas a um certo tamanho máximo (usualmente três palavras);
- Frases candidatas não podem ser nomes próprios, isto é, palavras únicas que sempre aparecem com uma inicial maiúscula; e
- Frases candidatas não podem começar ou terminar com uma *stopword*.

A lista de *stopwords* usada pelos autores do KEA continha 425 palavras divididas em nove classes sintáticas: conjunções, artigos, partículas, preposições, pronomes, verbos irregulares, adjetivos e advérbios. Segundo (Witten et al., 1999), para a maioria destas classes, todas as palavras listadas num dicionário *online* foram adicionadas à lista. Entretanto, para adjetivos e advérbios, eles introduziram várias subclasses, e palavras das subclasses foram adicionadas somente se elas fizessem parte das 60 palavras mais comuns do conjunto Brown (Brown, 2004) e (Kucera & Francis, 1967) *apud* (Witten et al., 1999). Além disso, só foram adicionadas palavras que ocorriam freqüentemente nestas subclasses.

Todas as seqüências contíguas de palavras em cada linha de entrada são testadas usando as três regras acima, resultando em um conjunto de frases candidatas. De acordo com (Witten et al., 1999), subfrases são freqüentemente candidatas delas próprias. Por exemplo, a linha que pode

ser o método de programação por demonstração gerará método, programação, demonstração, método de programação, programação por demonstração como frases candidatas, por causa de que o e por estão na lista de *stopwords*.

3.4.1.1.3 Case-folding e radicalização

O passo final em determinar frases candidatas é fazer *case-folding* em todas as palavras e aplicar o processo de radicalização.

Quanto à radicalização, o KEA usa o método iterativo de Lovins (Lovins, 1968) *apud* (Witten et al., 1999). Segundo (Witten et al., 1999), isto envolve usar o clássico radicalizador de Lovins para descartar qualquer sufixo, repetindo o processo sobre o radical resultante até que não exista mais mudança. Assim, por exemplo, a frase cut elimination torna-se cut elim.

Os autores usaram as versões reduzidas para comparar a saída do KEA com as palavras-chave dos autores dos artigos. Eles consideraram que uma palavra-chave especificada pelo autor foi identificada com sucesso, se, quando reduzida, ela é a mesma palavra-chave gerada pela máquina, que também foi reduzida. Um exemplo disto é que na Tabela 1 as frases cut-elimination e cut elimination, e proof nets e proof net, são consideradas equivalentes.

Os autores mantém as palavras reduzidas para cada frase, em suas maiúsculas originais, para apresentar ao usuário no caso da frase tornar-se uma palavra-chave. Quando várias maiúsculas diferentes ocorrem, a versão mais freqüente é escolhida.

3.4.1.2 Cálculo das características

Duas características são calculadas para cada frase candidata e usadas no treinamento e extração. Elas são: `TFxIDF`, que mede a freqüência de uma frase num documento comparado à sua raridade em uso geral; e a primeira ocorrência (*first occurrence*), que é a distância do primeiro aparecimento da frase dentro do documento.

3.4.1.2.1 TFxIDF

Esta característica compara a freqüência do uso de uma frase num documento particular com a freqüência da frase no uso geral. Uso geral é representado pelo número de documentos contendo tal frase num grande conjunto de textos. Tal característica da frase indica o quão comum ela é, e frases raras são mais prováveis de serem palavras-chave.

O KEA constrói um arquivo de freqüências para esta finalidade usando um conjunto de 100 documentos. Frases candidatas radicalizadas são geradas a partir de todos os documentos neste conjunto usando o método descrito acima. O arquivo de freqüências armazena cada frase e um contador de número de documentos no qual ela aparece.

Com este arquivo em mão, o `TFxIDF` para a frase `P` no documento `D` é:

`TFxIDF = (freq(P,D)/ size(D)) x` $-\log_2$ `df(P)/ N`, onde

- `freq (P, D)` é o número de vezes `P` ocorridas em `D`
- `size(D)` é o número de palavras em `D`
- `df(P)` é o número de documentos contendo `P` no conjunto global
- `N` é o tamanho do conjunto global

O segundo termo na equação é o `log` da probabilidade desta frase aparecer em qualquer documento do conjunto. O `log` é negado porque a probabilidade é menor do que 1. Se o documento não é parte do conjunto global, `df(P)` e `N` são ambos incrementados por 1 antes do termo ser avaliado, para simular o aparecimento no conjunto.

3.4.1.2.2 Primeira ocorrência

A segunda característica, que indica a primeira ocorrência, é calculada como o número de palavras que precedem o primeiro aparecimento da frase, dividido pelo número de palavras no documento. O resultado é um número entre 0 e 1 que representa quanto do documento precede o primeiro aparecimento da frase.

3.4.1.2.3 Discretização

Ambas as características são números reais e devem ser convertidos para dados discretos para a técnica de Aprendizado de Máquina. Durante o processo de treinamento, uma tabela de discretização para cada característica é derivada dos dados de treinamento. Esta tabela dá um conjunto de faixas numéricas para cada característica, e valores são substituídos pela faixa na qual o valor cai. A discretização é obtida usando o método de discretização supervisionada descrito em (Fayyad & Irani, 1993) *apud* (Witten et al., 1999).

3.4.1.3 Aprendizado: construção do modelo

O passo de treinamento usa um conjunto de documentos para treinamento, no qual as palavras-chave dos autores são conhecidas. Para cada documento de treinamento, frases candidatas são identificadas e os valores das características são calculados como descrito acima.

Para reduzir o tamanho do conjunto de treinamento, os autores descartam qualquer frase que ocorra somente uma vez no documento. Cada frase é então marcada como "é uma palavra-chave" ou "não é uma palavra-chave", usando as verdadeiras palavras-chave daquele documento. Esta característica binária é a característica da classe usada na técnica de aprendizado.

O esquema então gera um modelo que prediz a classe usando os valores das outras duas características. Foi experimentado com um número de diferentes técnicas de Aprendizado de Máquina. O algoritmo KEA adotou a técnica de aprendizado Naïve-Bayes (Domingos & Pazzani, 1997), porque ela é simples e gerou bons resultados. Esta técnica aprende dois conjuntos de pesos numéricos a partir dos valores discretizados das características, um conjunto aplicado a exemplos positivos ("é uma palavra-chave") e o outro a negativos ("não é uma palavra-chave").

3.4.1.4 Extração de novas palavras-chave

Para selecionar palavras-chave de um documento novo, o KEA determina frases candidatas e valores de suas características, então aplicando o modelo construído durante o treinamento. O modelo determina a probabilidade global de cada frase candidata ser uma palavra-

chave, e então uma operação de pós-processamento seleciona o melhor conjunto das palavras-chave.

Quando o modelo Naïve-Bayes é usado numa frase candidata com valores das características `t` (para `TFxIDF`) e `d` (para a distância do primeiro aparecimento), duas quantidades são computadas:

$$P[yes] = (Y / Y + N)\ P_{TFxIDF}[t \mid yes]\ P_{distance}[d \mid yes] \quad (1)$$

$$P[no] = (N / Y + N)\ P_{TFxIDF}[t \mid no]\ P_{distance}[d \mid no] \quad (2)$$

Nestas quantidades, `Y` é o número de exemplos positivos nos arquivos de treinamento (palavras-chave identificadas pelos autores) e `N` é o número de exemplos negativos (frases candidatas que não são palavras-chave). O estimador de Laplace é usado para evitar probabilidades nulas, substituindo nestes casos `Y` e `N` por `Y+1` e `N+1`.

A probabilidade global da frase candidata ser uma palavra-chave pode então ser calculada:

$$P = P[yes] / (P[yes] + P[no]) \quad (3)$$

Frases candidatas são ranqueadas de acordo com este valor, e dois passos de pós-processamento são executados. Primeiro, o valor `TFxIDF` (na sua forma pré-discretizada) é usado como desempate se duas frases têm probabilidade igual (o que é comum por causa da discretização). Segundo, são removidas da lista quaisquer frases que são subfrases de alguma frase com ranque maior. A partir da lista resultante, as primeiras `n` frases são retornadas, onde `n` é o número de palavras-chave solicitadas.

3.4.2 Avaliação do KEA

Segundo (Witten et al., 1999), foi realizada uma avaliação empírica do KEA usando documentos da Biblioteca Digital de Nova Zelândia (NZDL, 2004). O objetivo desta era avaliar a efetividade do KEA, e também investigar os efeitos da variação de diversos parâmetros no

processo de extração. Os autores mediram a qualidade das palavras-chave por contagem do número de coincidências entre a saída do KEA e as palavras-chave que foram originalmente escolhidas pelos autores dos documentos.

As seções seguintes esboçam a metodologia experimental e relata os resultados obtidos pelo KEA, descritos em (Witten et al., 1999).

3.4.2.1 Metodologia

O KEA foi avaliado usando a coleção de Relatórios Técnicos de Ciência da Computação (CSTR) da Biblioteca Digital da Nova Zelândia (NZDL, 2004). A partir de 46000 documentos deste conjunto, foram escolhidos 1800 onde os autores forneceram palavras-chave. A partir desses 1800, foram escolhidos ao acaso um conjunto de 500 documentos, restando 1300 como uma massa da qual seriam selecionados documentos de treinamento. Quanto maior o conjunto de testes, menor a medida de erro, assim os resultados serão aproximadamente os valores esperados para qualquer documento em particular. Finalmente, um conjunto adicional de documentos foi escolhido aleatoriamente a partir do restante da coleção CSTR como o conjunto global, usado para construir o arquivo de freqüência de frases.

Então, foram realizados quatro experimentos para determinar:

- a efetividade global do KEA;
- o efeito de mudar o tamanho e origem do conjunto global;
- o efeito de mudar o número de documentos de treinamento; e
- o desempenho do KEA usando resumos ao invés de todo o texto.

Os resultados de cada um desses experimentos são comentados abaixo. Primeiro, porém, serão descritas as medidas de qualidade, e discutidas as vantagens e desvantagens de usar palavras-chave especificadas pelos autores como um padrão de qualidade.

3.4.2.1.1 Medidas

Os autores avaliaram a efetividade do KEA pela contagem de palavras-chave que foram

também escolhidas pelos autores dos documentos, quando um número fixo ou pré-determinado de palavras-chave são extraídas automaticamente. Os autores usaram esta medida ao invés das métricas mais comuns utilizadas em Recuperação de Informação (precisão e recuperação), por três razões. Primeira, um único valor global é mais facilmente interpretado do que dois valores. Segunda, precisão e recuperação podem ser enganadoras, porque é fácil aumentar a precisão ao custo da recuperação (ao se retornar apenas a frase candidata mais promissora), ou recuperação ao custo da precisão (ao se retornar todas as candidatas). Terceira, esta medida se adequa bem ao comportamento esperado pelo usuário final, que provavelmente apenas solicitará um número fixo de palavras-chave. Caso necessário, porém, a precisão pode ser calculada dividindo a medida pelo número de frases recuperadas.

Os autores escolheram avaliar o KEA de encontro às escolhas dos autores dos documentos por diversas razões: este método de avaliação é simples, pode ser realizado automaticamente e mostra a comparação de diferentes métodos. Contudo, existem várias desvantagens em se usar palavras-chave dos autores como padrão: basicamente porque autores nem sempre escolhem palavras-chave que melhor descrevam o conteúdo de seus artigos. Autores podem escolher frases para direcionar seus trabalhos de alguma forma, ou para aumentar sua probabilidade de ser notado por pesquisadores. Também, palavras-chave são muitas vezes escolhidas apressadamente, apenas depois de que um documento é finalizado. Finalmente, (Witten et al., 1999) comenta que é questionável que os próprios autores sejam os mais qualificados para escolher frases que descrevam seus trabalhos para outros.

Este problema levanta duas questões. Primeira, a variação nas escolhas dos autores torna mais difícil para um método de extração automática operar bem. Segundo, escolhas incorretas do KEA (aquelas que não batem com escolhas dos autores) não são necessariamente palavras-chave ruins. Uma abordagem mais reveladora pode ser usar julgamento humano para avaliar independentemente a qualidade das frases do KEA, sem nem mesmo usar as escolhas originais dos autores. Esta abordagem, todavia, requer recursos consideráveis até mesmo para um único experimento, e então este método foi adiado para estudos futuros.

3.4.2.1.2 Efetividade global

O primeiro experimento avaliou a efetividade global do KEA, quando é feita a extração de até 20 palavras-chave para cada documento de teste. Este experimento usou 50 documentos de treinamento, tirados do conjunto padrão de 500 documentos de teste, e um conjunto global de 100 documentos para calcular as freqüências. Resultados selecionados são mostrados na Tabela 2 e ilustrados na Figura 3. Os números indicam a média do número de coincidências entre as palavras-chave escolhidas pelos autores e as extraídas pelo KEA. Por exemplo, para extrações de 10 palavras-chave por texto, em média 1,39 palavras-chave batiam, ou seja, pelo menos uma das palavras-chave coincidia exatamente.

TABELA 2 - Desempenho global. Adaptado de (Witten et al., 1999).

Palavras-chave extraídas	Média de acertos com palavras-chave dos autores
5	0,93
10	1,39
15	1,68
20	1,88

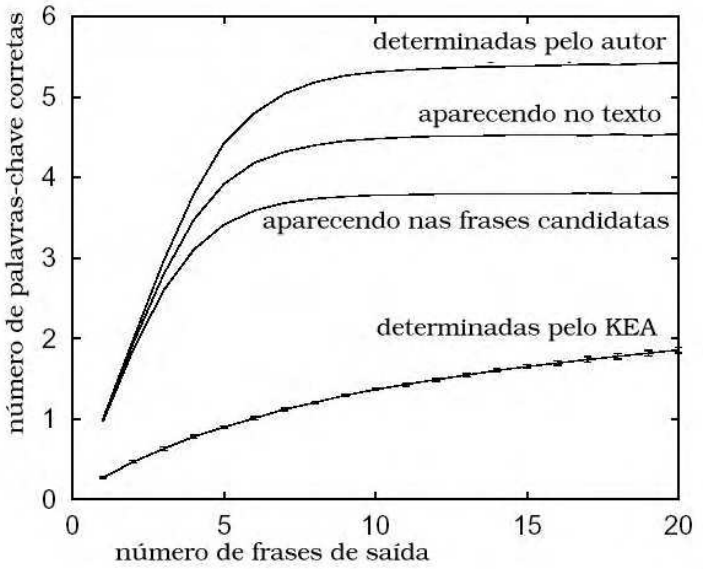

FIGURA 3 - Efetividade global (Witten et al., 1999).

Na Figura 3, a linha mais baixa mostra o número médio de identificações corretas. As

linhas superiores mostram três limites possíveis de desempenho. A primeira mostra quantas palavras-chave o autor determinou: claramente não é possível para qualquer algoritmo fazer melhor do que isto usando a medida padrão para sucesso. A assíntota mostra que o conjunto teste tem uma média de 5,4 palavras-chave dadas pelos autores para cada documento. A segunda linha de cima para baixo indica o número de palavras-chave que aparecem no texto do documento. Nenhum método de extração de palavras-chave (em oposição ao processo de determinação de palavras-chave) é capaz de identificar palavras-chave que não apareçam no texto. A terceira linha mostra o número de palavras-chave que aparecem entre as frases candidatas.

A Figura 3 então ilustra onde o algoritmo KEA apresenta dificuldades. A diferença entre as duas linhas do meio representa quantas palavras-chave não são selecionadas pelo processo de seleção de frases candidatas. A diferença entre as duas linhas de baixo representa o quanto é possível melhorar o processo de aprendizado em relação a encontrar as palavras-chave dos autores dentre as candidatas.

3.4.2.1.3 Efeito do tamanho e origem do conjunto global

Foi realizada uma série de testes para determinar como o tamanho e a origem do conjunto global afeta o desempenho. Como descrito anteriormente, o conjunto global é usado para construir um arquivo de freqüência de frases.

Para verificar o efeito da origem, foram construídos diferentes conjuntos globais: um conjunto independente de documentos similares, o conjunto de treinamento, os conjuntos de treinamento e de testes, o conjunto teste sozinho e um conjunto de documentos contendo um tipo diferente de textos. No julgamento dos autores, nenhum conjunto global se destacou de forma significativa com melhores resultados.

Para avaliar o efeito do tamanho do conjunto global, o KEA foi avaliado usando conjuntos de tamanhos diferentes. Para este teste, usou-se um conjunto de treinamento de 130 documentos, e o conjunto padrão de 500 documentos de teste. Todos os conjuntos foram formados ao acaso a partir dos documentos da coleção CSTR sem as palavras-chave dos autores. Como mostrado na Tabela 3, há pouco a ser ganho com o aumento do tamanho do conjunto

global além de cerca de 10 documentos, e a partir de 50 documentos não existem melhorias adicionais. Entretanto, o arquivo de freqüência de documentos é crucial para bons resultados; sem estas informações, o desempenho cai drasticamente.

TABELA 3 - Efeito da variação do tamanho do conjunto global (Witten et al., 1999).

Documentos no conjunto	Média de acertos (5 extraídas)	Média de acertos (15 extraídas)
0	?	?
1	0,674	1,307
5	0,738	1,445
10	0,822	1,560
50	0,884	1,644
100	0,868	1,644
1000	0,854	1,590

3.4.2.1.4 Efeito do tamanho do conjunto de treinamento

O terceiro experimento investigou como o número de documentos de treinamento (aqueles com palavras-chave identificadas) afeta o desempenho. Os autores do KEA estavam interessados no problema prático de quantos documentos de treinamento são necessários para se obterem bons resultados. Neste experimento, foi usado um conjunto global de 100 documentos da coleção CSTR, e o conjunto teste padrão. Foi variado o tamanho do conjunto de treinamento de 1 a 130 documentos, e testado o desempenho do KEA com cada conjunto.

TABELA 4 - Efeito do tamanho do conjunto de treinamento (Witten et al., 1999).

Documentos de treinamento	Média de acertos (5 extraídas)	Média de acertos (15 extraídas)
0	0,684	1,266
1	0,717	1,301
5	0,819	1,508
10	0,840	1,542
20	0,869	1,625
50	0,898	1,650
100	0,908	1,673

Os resultados na Tabela 4 mostram que o desempenho melhora constantemente até um

conjunto de aproximadamente 20 documentos, e que pequenos ganhos são obtidos até o conjunto de treinamento conter 50 documentos. Esses resultados indicam que um bom desempenho de extração pode ser obtido com um conjunto relativamente pequeno de documentos de treinamento. Em uma situação do mundo real onde uma coleção sem qualquer palavras-chave deve ser processada, especialistas humanos precisam somente ler e determinar palavras-chave de cerca de 25 documentos para ser possível extrair palavras-chave do resto da coleção.

3.4.2.1.5 Efeito do tamanho do documento

O experimento final considerou se o desempenho do KEA sofre quando somente utiliza resumos de documentos para extrair palavras-chave, e compara com o desempenho sobre o texto inteiro. Este experimento usou os conjuntos padrão de treinamento, teste e conjunto global, exceto que documentos sem sumário foram ignorados (deixando 110 documentos de treinamento e 429 documentos de treino).

A Tabela 5 mostra o número de palavras-chave corretas extraídas usando tanto os documentos resumidos quanto os completos. Como esperado, o KEA extraiu menos palavras-chave dos resumos que para o documento completo.

TABELA 5 - Efeito da variação do tamanho do documento (Witten et al., 1999).

Tamanho do documento	Média de acertos (5 extraídas)	Média de acertos (15 extraídas)
Texto completo	0,909	1,712
Sumários	0,655	1,028

4 ALGORITMOS DE RADICALIZAÇÃO DE PALAVRAS

Este capítulo dedica-se à apresentação de algoritmos propostos na literatura para a radicalização de palavras da língua portuguesa. O primeiro algoritmo descrito corresponde a uma adaptação de um algoritmo desenvolvido inicialmente para a língua inglesa, enquanto as duas técnicas seguintes foram propostas especificamente para a língua portuguesa. Na Seção 4.5 é justificada a escolha do algoritmo de radicalização utilizado neste trabalho.

Antes de detalhar como os algoritmos de radicalização podem ser aplicados à língua portuguesa, é importante abordar as definições e conceitos relacionados com a classificação e estrutura das palavras. Estas definições são feitas na seção seguinte.

4.1 Classe e estrutura das palavras

Segundo (Cunha & Cintra, 2001), uma **língua** é constituída de um conjunto infinito de frases. Cada frase possui uma parte sonora, ou seja a cadeia falada, e uma parte significativa, que correspondente ao seu conteúdo. Uma frase, por sua vez, pode ser dividida em unidades menores de som e significado, as **palavras**, e em unidades ainda menores, que apresentam apenas a parte significante, os **fonemas**. As palavras são, pois, unidades menores que a frase e maiores que o fonema.

De acordo com (Cunha & Cintra, 2001), existem, no entanto, unidades de som e conteúdo menores do que as palavras. Assim, em casas existem duas unidades significativas: casa e s. O primeiro elemento casa também se emprega como palavra isolada ou serve para

formar outras palavras isoladas: casarão, casamento e casinha. Já a forma plural -s que vai aparecer no final de muitas outras palavras (flores, árvores, rios, cristalinas), nunca será usada como palavra individual e autônoma. A essas unidades significativas mínimas dá-se o nome de **morfema**.

Quanto à natureza da significação, os morfemas classificam-se em lexicais e gramaticais. São morfemas lexicais os substantivos, os adjetivos, os verbos e os advérbios de modo. São morfemas gramaticais os artigos, os pronomes, os numerais, as preposições, as conjunções e os demais advérbios, bem como as formas indicadoras de número, gênero, tempo, modo ou aspecto verbal.

Os elementos mórficos ou morfemas dividem-se em: raiz, radical ou tema, vogal temática, desinência, afixo e vogal e consoante de ligação.

4.1.1 Raiz

Em muitas gramáticas, o estudo das raízes é omitido, embora o termo esteja definido na Nomenclatura Gramatical Brasileira (NGB). Raiz não é radical ou tema. **Raiz** é o elemento mórfico mais simples a que pode ser reduzida uma palavra. Obtém-se a raiz pela eliminação dos elementos secundários de formação.

Palavra	**Raiz**
abandonar	bann-
abandono	bann-
abnegar	neg-

É possível que aconteça a coincidência entre a raiz e radical ao mesmo tempo:

Palavra	**Raiz, radical ou tema**
lavar	lav-

Algumas palavras, com a evolução da língua, mantêm apenas uma letra da raiz original, como os exemplos abaixo:

Palavra	Origem latina	Raiz	O que Resta
feito	factu	fac-	f-
feitor	factore	fac-	f-
malfeitor	malefactore	fac-	f-

4.1.2 Radical

Radical é o elemento mórfico que fornece a significação da palavra. É o radical que relaciona as palavras da mesma família e lhes transmite uma base comum de significação. No exemplo abaixo, o radical livr- é comum entre todas as palavras.

Palavra	Radical
livro	livr-
livraria	livr-
livreiro	livr-

Ao radical acrescido de uma vogal temática, isto é, pronto para receber uma desinência (ou um sufixo), denomina-se **tema**. Segundo (Coutinho, 1954) *apud* (De Lucca & Nunes, 2002), é comum que não seja feita a distinção entre tema e radical, sendo ambas utilizadas como sinônimos. Portanto, neste trabalho também não é feita esta distinção.

4.1.3 Vogal temática

Vogal temática é o elemento mórfico que se agrega ao radical de uma palavra para que ela possa receber outros morfemas. Podem ser nominais e verbais.

As vogais temáticas **nominais** referem-se a um substantivo ou a um adjetivo. Por exemplo:

Palavra	Vogal temática
livro	o
música	a

Já as vogais temáticas **verbais** referem-se apenas a verbos, como por exemplo:

Palavra	**Vogal temática**
dançar	a
mexer	e
sair	i

4.1.4 Desinência

Desinência é o nome dado para os elementos mórficos que indicam as flexões das palavras, e divide-se em nominais e verbais.

As desinências **nominais** servem para indicar o gênero e o número dos substantivos, dos adjetivos e de certos pronomes. Por exemplo:

Palavra	**Desinência**	
belas	-a	para indicar o feminino
	-s	para denotar o plural

Por outro lado, as desinências **verbais** servem para indicar o tempo, o número, o modo e a pessoa. Por exemplo:

Palavra	**Desinência**	
aprendestes	-stes	2ª pessoa do plural do pretérito perfeito do indicativo
aprendeu	-u	3ª pessoa do singular do pretérito perfeito do indicativo

4.1.5 Afixo

Afixo é o elemento mórfico que se junta a uma raiz ou radical a fim de modificar geralmente de forma precisa o sentido de uma palavra. Os afixos subdividem-se em sufixos e prefixos. Os **sufixos** se pospõem ao radical; os **prefixos** se antepõem ao radical:

Palavra	Prefixo	Palavra	Sufixo
despreocupar	des-	gatinho	-inho
renovar	re-	nebuloso	-oso
incoerente	in-	felizmente	-mente

4.1.6 Vogal e consoante de ligação

Ainda segundo (Cunha & Cintra, 2001), os elementos mórficos citados acima entram sempre na estrutura do vocábulo com determinado valor significativo externo ou interno. Há, porém, outros que não são significativos, e servem apenas para evitar dissonâncias (hiatos, encontros consonantais) na junção daqueles elementos. Por exemplo, nos vocábulos gasômetro e cafeteira, verifica-se que:

- o primeiro é formado de dois radicais gás- + -metro, ligados pela vogal o, sem valor significativo;
- o segundo é constituído do radical café- + o sufixo -eira, entre os quais aparece a consoante não significativa t para evitar o desagradável hiato -éê-.

A esses sons, empregados para tornar a pronúncia das palavras mais fácil ou eufônica, dá-se o nome de **vogais e consoantes de ligação**.

4.2 Porter Stemmer

O radicalizador de Porter foi desenvolvido por Martin Porter na Universidade de Cambridge em 1980. O radicalizador é baseado na idéia de que sufixos da língua inglesa (aproximadamente 1200) são na maioria das vezes feitos de uma combinação de sufixos menores e mais simples. Este radicalizador tem 5 passos, aplicando regras dentro de cada passo. Em cada passo, se uma regra de sufixo coincide com uma palavra, então as condições associadas àquela regra são testadas sobre o que seria o radical resultante caso aquele sufixo fosse removido. Por exemplo, tal condição poderia ser o número de vogais (que são seguidas por consoantes no radical), seja maior do que um para a regra ser aplicada.

Uma vez que as condições de uma regra são aplicadas, o sufixo é removido e o controle vai para o próximo passo. Se a regra não é aceita, então a próxima regra no passo é testada, até uma outra regra deste passo ser aceita ou até não existirem mais regras, quando passa-se para o passo seguinte. Este processo continua por todos os cinco passos, retornando o radical resultante após a execução do quinto passo.

4.3 PegaStemming

O trabalho de (Chaves, 2003) descreve um estudo e apreciação sobre dois algoritmos de radicalização para a língua portuguesa. Os algoritmos estudados e avaliados foram o PegaStemming (que será tratado como R_1), cuja autoria é de Marco Antonio Insaurriaga Gonzalez, e o Portuguese Stemmer (que será tratado, nesta seção como R_2) e que também será descrito na próxima seção. Não foi encontrado nenhum material adicional sobre o PegaStemming, apenas o trabalho de (Chaves, 2003).

De acordo com (Chaves, 2003), apenas o aspecto de precisão foi levado em consideração, pois para ser feita uma apreciação completa e profunda de um radicalizador é necessário, pelo menos, um conjunto de textos mais amplo. Na análise de precisão dos algoritmos foram utilizadas 500 palavras, sem repetição, retiradas de textos (resumos de dissertações de mestrado em Ciência da Computação da Pós-Graduação da PUCRS) escolhidos aleatoriamente.

No caso das palavras acentuadas, quando processadas corretamente pelos algoritmos, optou-se por considerar correto tanto o radical com acento quanto o radical sem acento. O algoritmo R_1 não faz tratamento para artigos, conjunções e preposições. Portanto, foram incluídas na análise as seguintes classes gramaticais: substantivo, verbo, adjetivo, advérbio, contração de preposição e pronome.

O processo de análise de precisão começou com o procedimento de radicalização manual, que permitiu verificar o radical correto de cada palavra. Com isso obteve-se um parâmetro para medir a precisão dos radicalizadores quando processam automaticamente um texto. Em seguida, foi realizada a execução dos radicalizadores utilizando o conjunto de palavras

selecionadas.

De acordo com (Chaves, 2003), alguns dos resultados da análise foram os seguintes :

- A quantidade total de erros atribuídos à *overstemming* foi bastante próxima para os dois algoritmos, 69 palavras processadas por R_1 e 77 palavras processadas por R_2;
- O algoritmo R_2 apresentou somente 18 erros atribuídos à *understemming*, enquanto que o R_1 apresentou 75;
- Ambos os algoritmos processaram corretamente 57,2% das palavras, ao passo que 13,2% foram processadas equivocadamente. Segundo (Chaves, 2003), esses resultados indicam que houve uma concordância entre os algoritmos de 70,4% das palavras utilizadas no experimento, ou seja, as mesmas palavras foram processadas ou de forma correta ou equivocada;
- O algoritmo R_1 teve seu pior desempenho na categoria gramatical pronome e o R_2 na categoria gramatical contração de preposição, na qual o algoritmo processou de forma equivocada 42% das palavras;
- O algoritmo R_1 apresentou a melhor precisão para a categoria substantivo processando 70% das palavras corretamente e o R_2 obteve a melhor precisão para a categoria advérbio processando 80% das palavras de forma correta;
- O algoritmo R_2 apresentou um desempenho melhor que o algoritmo R_1 em todas as categorias gramaticais. O percentual total de acerto de R_2 foi de 78% contra 63,8% de R_1.

Segundo (Chaves, 2003), este melhor desempenho do R_2 pode ser atribuído ao fato de o mesmo já ter sido testado em um conjunto de palavras maior do que o R_1, conforme (Orengo & Huyck, 2001).

4.4 Portuguese Stemmer

O algoritmo Portuguese Stemmer foi proposto por Viviane Orengo e Christian Huyck em (Orengo & Huyck, 2001). Este algoritmo leva em conta as classes morfológicas, executando uma série de passos de remoção de sufixos conhecidos. Os passos são aplicados na seguinte

seqüência: **1.** Redução do plural, **2.** Redução do feminino, **3.** Redução do advérbio, **4.** Redução do aumentativo e diminutivo, **5.** Redução das formas nominais, **6.** Redução das terminações verbais, **7.** Redução da vogal temática, **8.** Remoção dos acentos. A Figura 4 ilustra a seqüência dos passos.

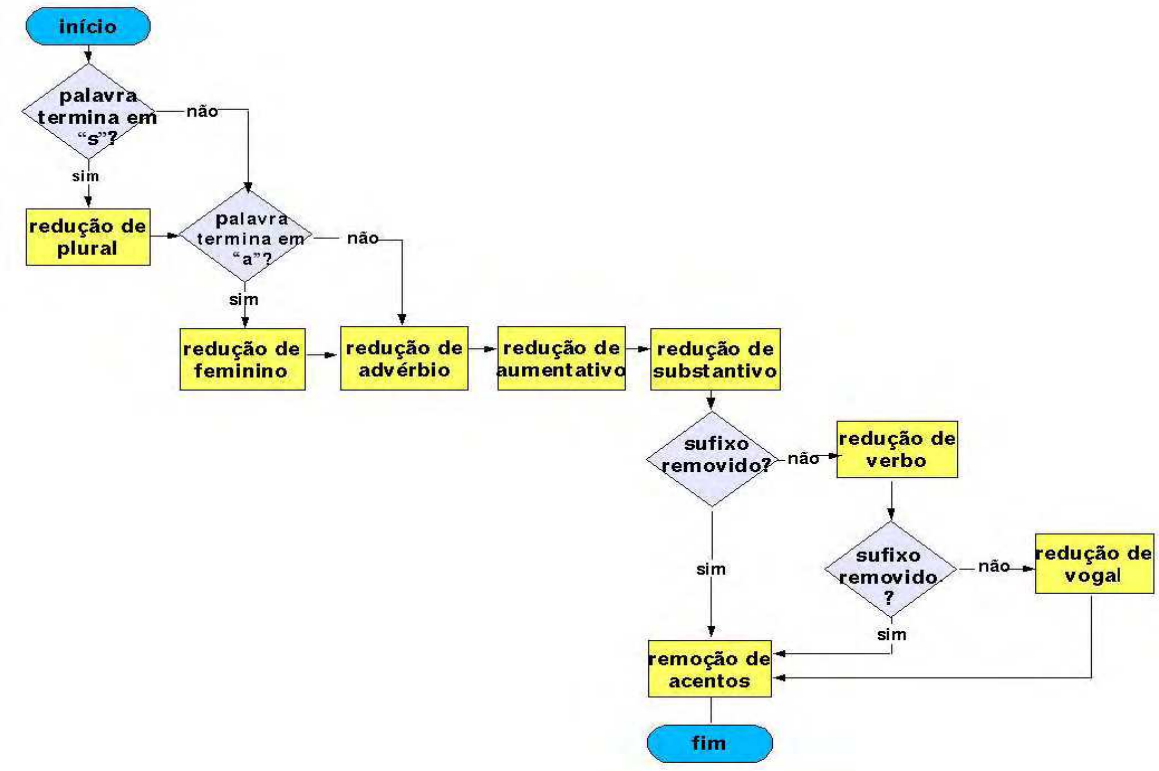

FIGURA 4 - Os passos do Portuguese Stemmer

4.4.1 Os passos do Portuguese Stemmer

Cada passo tem um conjunto de regras. As regras dentro de um passo são examinadas em seqüência e somente uma regra pode ser aplicada em cada passo. O possível sufixo mais longo é sempre removido primeiro por causa da ordem das regras dentro de um dado passo. Por exemplo, o sufixo de plural -es deve ser testado antes do sufixo -s. Tal como descrito em (Orengo & Huyck, 2001), o algoritmo definia 199 regras. Uma implementação deste radicalizador está disponível em (Portuguese Stemmer, 2004).

Cada regra estabelece:

- O sufixo a ser removido;
- O tamanho mínimo do radical, para evitar remover um sufixo quando o radical é muito curto. Esta medida varia para cada sufixo, e os autores desse radicalizador definiram os valores observando listas de palavras terminadas num dado sufixo. Segundo (Orengo & Huyck, 2001), apesar de não haver suporte lingüístico para este procedimento, ele reduz erros de *overstemming*;
- Um sufixo substituto para ser anexado ao radical, se aplicável;
- Uma lista de exceções, que identificam palavras que terminam no sufixo indicado, mas que não deverão ser reduzidas. As listas foram construídas com o auxílio de um vocabulário de 32000 palavras da língua portuguesa disponível em (Snowball, 2003). Segundo (Orengo & Huyck, 2001), testes com o radicalizador mostrou que listas de exceções reduzem erros de *overstemming* em 5%.

Um exemplo de regra é:

“inho”, 3, “ ”, { “caminho”, “golfinho”, “padrinho”, “sobrinho”, “vizinho” }

Onde -inho é um sufixo que demonstra diminutivo, 3 é o tamanho mínimo do radical, o que evita que a regra seja aplicada em palavras como linho. As palavras entre chaves são justamente as exceções para esta regra, ou seja, são palavras terminadas pelo sufixo -inho, mas que não são diminutivos. Todas as outras palavras que terminam em -inho e são compostas por seis ou mais caracteres serão reduzidas ao radical. Observe que não existe sufixo de substituição para esta regra.

A seguir, serão detalhadas cada uma das regras utilizadas.

4.4.1.1 Redução do plural

Com algumas exceções, a forma plural da língua portuguesa termina em -s. Entretanto, nem todas as palavras terminadas em -s denotam plural, por exemplo, lápis, mais e ademais. Este

passo consiste basicamente em remover o final -s das palavras que não estão listadas na lista de exceções. Contudo, algumas vezes ajustes são necessários, como por exemplo, palavras terminadas em -ns devem ter o sufixo substituído por -m como em bons → bom, palavras terminadas em -ões devem ter o sufixo substituído por -ão como em ações → ação.

4.4.1.2 Redução do feminino

Todos os substantivos e adjetivos na língua portuguesa tem um gênero. Este passo consiste em transformar palavras que estão no gênero feminino em suas correspondentes no gênero masculino. Somente palavras terminadas em -a são testadas neste passo, mas nem todas são transformadas, apenas as que terminam em sufixos mais comuns, por exemplo chinesa → chinês, vilã → vilão.

4.4.1.3 Redução do advérbio

Este é o menor passo de todos, já que existe apenas um sufixo que denota advérbios: -mente. Novamente, nem todas as palavras com esta terminação são advérbios, por exemplo, a palavra experimente termina em -mente, mas não é um advérbio, por isso uma lista de exceções é necessária.

4.4.1.4 Redução do aumentativo e diminutivo

Conforme (Cunha & Cintra, 2001), nem sempre o sufixo aumentativo se junta ao radical de um substantivo. Há derivações feitas sobre adjetivos (ricaço, de rico; sabichão, de sábio) e também sobre radicais verbais (chorão, de chorar; mandão, de mandar).

As palavras têm aumentativo, diminutivo e formas superlativas, por exemplo, casinha = "casa pequena", onde -inha é o sufixo que indica um diminutivo. Esses casos são tratados neste passo.

4.4.1.5 Redução de formas nominais

Este passo testa as palavras contra 61 terminações de substantivos e adjetivos. Por exemplo, palavras terminadas com o sufixo -ista devem ter este sufixo removido como em realista real, palavras terminadas com o sufixo -ismo devem ter este sufixo removido como em realismo real. Observe que são duas palavras diferentes que conservam uma relação de sentido com o mesmo radical. Se um sufixo é removido neste passo, os passos de redução de verbos (Seção 4.4.1.6) e redução da vogal temática (Seção 4.4.1.7) não são executados.

4.4.1.6 Redução das terminações verbais

A língua portuguesa é uma língua muito rica em termos de formas verbais. Enquanto os verbos regulares da língua inglesa têm apenas quatro variações (por exemplo, *talk, talks, talked e talking*), os verbos regulares da língua portuguesa têm mais de 50 formas diferentes (Cunha & Cintra, 2001). Os verbos apresentam as variações de número, de pessoa, de modo, de tempo, de aspecto e de voz. Foi verificado que a estrutura das formas verbais se relacionam pelo radical, parte invariável que lhes dá a base comum de significação. Também foi verificado que a esse radical verbal se junta, em cada forma, uma terminação, da qual participa pelo menos um dos elementos:

- a vogal temática a, característica dos verbos da primeira conjugação:

  cant-**a** cant-**a**-va cant-**a**-ra

- o sufixo temporal (ou modo-temporal), que indica o tempo e o modo:

  cant-a-**va** cant-a-**ra**

- a desinência pessoal (ou número-pessoal), que indica a pessoa e o número:

  cant-o cant-a-va-s cant-á-ra-mos

Neste passo as formas verbais são reduzidas ao seu radical. Por exemplo, todas as formas verbais citadas acima serão reduzidas ao radical cant-.

4.4.1.7 Redução da vogal temática

Este passo consiste em remover a última vogal (a, e, ou o) das palavras que não sofreram o processo de radicalização nos passos de redução de formas nominais e de terminações verbais. Por exemplo, menino não sofreria nenhuma modificação nos passos anteriores, portanto este passo removerá o final -o, assim podendo ser combinado com outras formas diferentes, tais como menina, meninice, meninão e menininho, que também serão convertidas para o radical menin-.

4.4.1.8 Remoção dos acentos

Remover acentos é necessário porque existem casos em que algumas formas diferentes de palavras são acentuadas e algumas não, como em psic**ó**logo e psic**o**logia. Depois desse passo ambas formas seriam combinadas a psicolog. É muito importante que este passo seja feito somente neste ponto e não no início do algoritmo, porque a presença de acentos é significante para algumas regras, como por exemplo, -óis → -ol transforma sóis em sol. Por outro lado, se a regra fosse -ois → -ol, poderia produzir erros como transformar bois em bol.

4.4.2 Dificuldades no processo de radicalização da língua portuguesa

Serão comentadas a seguir, dificuldades encontradas por (Orengo & Huyck, 2001) na aplicação do algoritmo Portuguese Stemmer.

4.4.2.1 Conduta com exceções

Esta foi uma das maiores dificuldades na construção do algoritmo de radicalização, pois quase todas as regras formuladas pelos autores possuíam exceções. Por exemplo, -ão é um sufixo comumente usado para indicar aumentativo, contudo nem todas as palavras terminadas em -ão denotam aumentativo. Portanto, diferentemente do radicalizador de Porter, foram usadas listas de exceções. Segundo (Orengo & Huyck, 2001), se não usassem tais listas, o radicalizador poderia fazer erros de *overstemming* se a regra estivesse presente, e erros de *understemming* se a regra fosse retirada.

4.4.2.2 Palavras com grafia igual e significados diferentes

Existem vários casos desta natureza, muitos envolvendo conjugação de verbos, por exemplo, casais que pode significar "duas pessoas" ou segunda pessoa do plural do verbo casar. O algoritmo Portuguese Stemmer não tem informações sobre as categorias das palavras, portanto os sentidos diferentes dessas palavras não são distinguidos. Para este caso específico, o radicalizador assume o primeiro significado e reduz a palavra para a forma singular casal, isto é devido à segunda pessoa do plural ser pouco utilizada na língua portuguesa moderna.

4.4.2.3 Verbos irregulares

A versão da publicação não trata verbos irregulares, mas de acordo com (Orengo & Huyck, 2001), eles surpreendentemente parecem não afetar os resultados. Os testes mostraram que menos do que 1% de erros ocorrem por causa desta razão.

4.4.2.4 Mudanças no radical

Existem casos em que o processo de flexão muda o radical da palavra. Os casos em que a mudança obedece regras ortográficas, por exemplo, -ns → -m, foram tratados com sucesso. De acordo com (Orengo & Huyck, 2001), para os outros casos, ainda estava sendo procurado um caminho efetivo para lidar com os mesmos. Pelo algoritmo original, palavras como emitir e emissão que estão relacionadas, não são combinadas, a primeira é reduzida para emit- e a segunda para emis-.

4.4.2.5 Tratamento de nomes próprios

Segundo (Orengo & Huyck, 2001), nomes próprios não devem ser reduzidos, pois o problema é justamente identificá-los. Uma lista de nomes próprios não apresentava uma solução ideal por duas razões principais: existem infinitas possibilidades e alguns nomes próprios são compartilhados com nomes de coisas. Por exemplo, Pereira é um sobrenome da língua portuguesa comum, mas também significa "árvore de pêra". Portanto, a implementação do Portuguese Stemmer, na publicação, reduz nomes próprios tal como o radicalizador de Porter.

4.4.3 Avaliação do Portuguese Stemmer

De acordo com (Orengo & Huyck, 2001), para avaliar a eficiência do Portuguese Stemmer, foi realizada uma seqüência de testes. Esses testes usaram um vocabulário de 32000 formas de palavras diferentes obtidas a partir de (Snowball, 2003). Os autores testaram o algoritmo contra a versão na língua portuguesa do radicalizador de Porter aplicado ao mesmo vocabulário.

4.4.3.1 Redução do vocabulário

Um dos propósitos dos autores para a remoção de sufixos era reduzir o tamanho do vocabulário para finalidades de criação de índices. Segundo (Orengo & Huyck, 2001), Porter relata uma redução de aproximadamente um terço do vocabulário, usando seu radicalizador sobre 10000 palavras diferentes na língua inglesa. A versão na língua portuguesa do radicalizador de

Porter reduziu o vocabulário em 44%; isto acontece porque a língua portuguesa possui um número maior de flexões do que a língua inglesa. O Portuguese Stemmer reduziu o vocabulário em 51%.

4.4.3.2 Comparação com a saída prevista

A seguir, está descrito o método usado por (Orengo & Huyck, 2001) para treinar o algoritmo Portuguese Stemmer.

Os autores selecionaram aleatoriamente 2800 palavras a partir do vocabulário e manualmente determinaram para cada palavra o radical correto. Feito isto, os autores testaram os radicais calculados contra a saída esperada e analisaram os erros verificando se novas regras ou exceções seriam necessárias. Contudo, houve um estágio no projeto do algoritmo de radicalização em que a adição de novas regras com o objetivo de evitar erros sobre um caso específico causou imprecisões em outros casos. No final do processo de treinamento, o Portuguese Stemmer conseguiu 98% de precisão no vocabulário de treinamento.

Os autores então decidiram avaliar o desempenho do algoritmo usando um conjunto de palavras que não foram usadas no treinamento, desse modo eles selecionaram outras 1000 palavras e novamente determinaram para cada uma o radical correto. O algoritmo calculou o radical certo 96% das vezes superando a versão na língua portuguesa do radicalizador de Porter, o qual calculou o radical correto 71% das vezes, para o mesmo conjunto de teste. Os autores relataram a consciência de que era um conjunto de teste muito pequeno e que eles pretendiam repetir esse experimento numa amostra maior.

5 ADAPTAÇÃO DO ALGORITMO KEA PARA A LÍNGUA PORTUGUESA

5.1 Criação de uma lista de stopwords da língua portuguesa

Stopwords são termos freqüentes em um texto e que não carregam nenhuma informação de maior relevância. As *stopwords* tipicamente são compostas por palavras das seguintes classes: artigos, preposições, conjunções, pronomes e advérbios.

A remoção de *stopwords* tem como objetivo principal eliminar termos que não são representativos ao documento. Esta etapa também pode ser considerada uma técnica de compressão de textos, pois a eliminação de *stopwords* reduz o número de palavras a serem analisadas no documento e também o número de palavras a serem armazenadas na base de dados.

Para este trabalho, a identificação de *stopwords* nos textos é importante, pois isto possibilita limitar o início e o fim de trechos das frases dos documentos, tal como feito pelo algoritmo KEA (Seção 3.4.1.1.2).

Foi realizada uma pesquisa sobre listas de *stopwords* da língua portuguesa, mas foram encontradas apenas quatro listas, todas com algumas inconsistências e também incompletas. Portanto, depois de julgá-las não adequadas para este trabalho, decidiu-se criar uma lista de *stopwords* da língua portuguesa, justificando todo o processo de construção da mesma.

O primeiro passo foi formar uma lista com as seguintes classes de palavras: artigos, pronomes, advérbios, preposições, conjunções, consoantes e vogais. As classes de palavras foram obtidas de (Cunha & Cintra, 2001).

Alguns autores incluem as interjeições também como *stopwords*, porém optou-se por não incluí-las. Essa decisão foi tomada por que as interjeições (por exemplo, alô, chi, oba, oxalá e eia) são termos de baixa freqüência e que também tem uma grande variabilidade, sendo impraticável listar todos os termos que podem funcionar como interjeições.

Após a lista estar formada, percebeu-se que algumas palavras poderiam também ser substantivos ou adjetivos, mas isso dependeria do contexto. Portanto, decidiu-se fazer consultas de todas as palavras que estavam na primeira versão da lista no dicionário Houaiss (Houaiss, 2004) para esclarecer algumas ambigüidades.

As palavras classificadas em mais de uma categoria foram mantidas na sua classificação mais comum, isto foi feito para não haver palavras classificadas em duas ou três classes. Para fazer esta classificação única seguiu-se a ordem de classificação do dicionário Houaiss. Por exemplo, menos, muito e pouco estão classificados nessa ordem: pronome, advérbio e substantivo, e portanto ficaram na classe dos pronomes. As palavras que foram classificadas primeiramente como adjetivos ou substantivos foram excluídas da lista.

A lista completa de *stopwords* é apresentada no Apêndice A.

A seguir, será descrito em detalhes o processo de seleção das palavras que formam a lista de *stopwords* criada. Todas as citações de exemplos reproduzidas a seguir foram retiradas de (Cunha & Cintra, 2001).

5.1.1 Artigos

A lista é formada pelos artigos definidos e indefinidos e suas formas combinadas com preposições. **Artigos** são as palavras o (com as variações a, os, as) e um (com as variações uma, uns, umas), que se antepõem aos substantivos para indicar:

- que se trata de um ser já conhecido do leitor ou ouvinte, seja por ter sido mencionado antes, seja por ser objeto de um conhecimento de experiência;
- que se trata de um simples representante de uma dada espécie ao qual não se fez menção anterior.

No primeiro caso, diz-se que o artigo é definido; no segundo, indefinido.

5.1.2 Pronomes

A lista também é formada pelos pronomes substantivos e pronomes adjetivos. Os **pronomes** desempenham na frase as funções equivalentes às exercidas pelos substantivos e nomes próprios. Servem, pois:

- para representar um substantivo;
- para acompanhar um substantivo determinando-lhe a extensão do significado.

De acordo com (Cunha & Cintra, 2001), facilmente se distinguem na prática essas duas classes de pronomes, porque os pronomes substantivos aparecem isolados na frase, ao passo que os pronomes adjetivos se empregam sempre junto de um substantivo, com o qual concordam em gênero e número.

Assim, nas frases:

Lembranças a todos **os teus**.
(E. da Cunha, *OC,* II, 646.)

Teus olhos são dois desejos.
(R. Correia, *PCP*, 109.)

A palavra teus é pronome substantivo, na primeira, e pronome adjetivo, na segunda.

Há seis espécies de pronomes: pessoais, possessivos, demonstrativos, relativos, interrogativos e indefinidos. Todas as espécies de pronomes foram acrescentadas à lista. Abaixo, serão comentados os motivos de alguns pronomes terem sido ou não acrescentados à lista.

Os pronomes pessoais oblíquos o (com as variações a, os, as), no (com as variações na, nos, nas) e as suas formas no-lo, no-la, no-los, no-las não foram adicionados nesta classificação porque a classificação mais comum deles é de artigos e todos os artigos foram adicionados à lista anteriormente. Das formas compostas o qual (com as variações a qual, os quais, as quais) dos

pronomes relativos só foram classificadas como pronomes relativos as palavras qual e quais, pois o o (com as variações a, os, as) já foi classificado como artigo.

As palavras tal, mesmo, próprio e semelhante podem também funcionar como pronomes demonstrativos quando:

- Tal é sinônimo de:

a) "este", "esta", "isto", "esse", "essa", "isso", "aquele", "aquela", "aquilo":

Como era possível que nunca tivesse dado por **tal**?

(M.J. de Carvalho, *TM*, 57.)

Tal foi a primeira conclusão do Palha; mas vieram outras hipóteses.

(Machado de Assis, OC, I, 602.)

b) "semelhante":

Houve tudo quando se faz em **tais** ocasiões.

(Machado de Assis, OC, II, 197.)

Tal situação confundia-a fortemente, e fazia diminuir aquele vigor e energia com que a conhecemos.

(A. Assis Júnior, *SM*, I, 198.)

- Mesmo e próprio têm o sentido de "exato", "idêntico", "em pessoa":

Ela não tem amor **próprio**.

Eu não posso viver muito tempo na **mesma** casa, na **mesma** rua, no **mesmo** sítio.

(Luandino Vieira, *JV*, 62.)

Foi a **própria** Carmélia quem me fez o convite.

(C. dos Anjos, *DR,* 161.)

Acontece sempre a **mesma** coisa.

- Semelhante serve de "identidade":

O Lucas reparou nisso e doeu-se intimamente de **semelhante** descuido.

(M. Torga, *CM,* 84.)

Tudo o que disse foi, sem dúvida, convencional, e nem a jovem Aurora podia deixar de recorrer às fórmulas que se usam em **semelhantes** conjunturas.

(C. dos Anjos, *DR, 284*.)

As palavras mesmo (com as variações mesma, mesmos, mesmas), próprio (com as variações própria, próprios, próprias) e semelhante(s) não foram adicionadas à lista pelo fato de que só são pronomes demonstrativos quando expressam os sentidos mostrados acima. Quando foram consultadas em (Houaiss, 2004), tais palavras seriam prioritariamente adjetivos, depois podendo ser substantivos e enfim pronomes pela freqüência de uso.

A palavra tal foi adicionada porque na maioria dos casos ela expressa os sentidos mostrados acima e também porque ela é classificada em (Houaiss, 2004) como pronome, advérbio e por último substantivo.

As seguintes palavras também não foram adicionadas à lista porque foram classificados primeiramente como substantivos e/ou adjetivos. São elas: certo (com as variações certa, certos, certas); vário (com as variações vária, vários, várias); todo (com as variações toda, todos, todas); quanto (com as variações quanta, quantos, quantas); qualquer e quaisquer.

Conforme (Cunha & Cintra, 2001) a palavra tudo é normalmente pronome substantivo, mas tem valor de adjetivo nas seguintes combinações: tudo isto, tudo isso, tudo aquilo, tudo o que, tudo o mais e semelhantes. Mas, como o algoritmo KEA trata as palavras separadamente, essas combinações não existirão, portanto a palavra tudo ficou classificada como pronome.

Há também os pronomes de tratamento. Denomina-se pronomes de tratamento certas palavras e locuções que valem por verdadeiros pronomes pessoais, como: você, o senhor e Vossa Excelência. Os pronomes de tratamento não foram acrescentados porque são formados por duas

palavras. Por exemplo: Vossa Alteza, Vossa Santidade, Vossa Paternidade e Vossa Majestade. Como o algoritmo KEA trata as palavras separadamente, as palavras Alteza, Santidade, Paternidade e Majestade são tratadas sozinhas e quando isso acontece elas passam a ser substantivos e a palavra Vossa é um pronome possessivo, que já está na lista, portanto na lista de *stopwords* não constam os pronomes de tratamento.

5.1.3 Advérbios

Segundo (Cunha & Cintra, 2001), o **advérbio** é fundamentalmente um modificador do verbo. Os advérbios recebem a denominação da circunstância ou de outra idéia acessória que expressam. A Nomenclatura Gramatical Brasileira (NGB) distingue as seguintes espécies: advérbios de afirmação, advérbios de dúvida, advérbios de intensidade, advérbios de lugar, advérbios de modo, advérbios de negação, advérbios de tempo. Há também os advérbios que se empregam nas interrogações diretas e indiretas. Esses são chamados de advérbios interrogativos e são os seguintes: por que (advérbio de causa), onde (advérbio de lugar), como (advérbio de modo) e quando (advérbio de tempo).

Conforme (Cunha & Cintra, 2001) a Nomenclatura Gramatical Portuguesa (NGP) acrescenta à essa lista três outras espécies: advérbios de ordem, advérbios de exclusão e advérbios de designação. Os dois últimos foram incluídos pela NGB e passaram a ter uma classificação à parte. Esta classificação não tem nome especial, mas os autores utilizaram o termo de palavras denotativas. Essas palavras não apresentam as características normais dos advérbios, quais sejam as de modificar o verbo, o adjetivo ou outro advérbio. Ao contrário, estas são palavras que denotam, por exemplo: inclusão, exclusão, designação, realce, retificação e situação.

Contudo, optou-se por não fazer esta distinção, porque na gramática a listagem de palavras dessa categoria e dos advérbios não é exaustiva. Foram identificadas palavras que não estão na gramática, mas que são listadas em dicionários como advérbios, portanto foi criado um conjunto de outros advérbios. Nem todos os advérbios foram adicionados à lista por causa da variedade de palavras da língua portuguesa, de forma que foi considerado impraticável listar e classificar todos os advérbios corretamente.

Os advérbios terminados em -mente não foram adicionados à lista porque existem várias palavras terminadas em -mente que não são advérbios, e que conseqüentemente não deveriam ser tratadas como *stopwords* no processo de extração.

Chegou a ser levantada a idéia de colocar uma regra na implementação do KEA, a qual fizesse com que todas as palavras terminadas em -mente fossem consideradas advérbios e tratadas como *stopwords*. Esta regra poderia trabalhar com uma lista de exceções, na qual as palavras que estivessem nessa lista de exceções não seriam consideradas como *stopwords*. Contudo receou-se que alguns nomes próprios poderiam terminar em -mente e por isso se tornariam *stopwords*. Por exemplo, as palavras Clemente e Valmente, que possuem a terminação -mente não são advérbios e sim nomes próprios. A palavra clemente também pode ser um adjetivo, mas a maior preocupação foi com nomes próprios, pois a transformação de alguns nomes próprios em *stopwords* poderia afetar seriamente o processo de determinação de palavras-chave em um documento.

No momento que uma palavra é considerada uma *stopword* pelo KEA, essa palavra fica impossibilitada de estar no início ou no fim de uma palavra-chave, não sendo descartada e podendo fazer parte de uma palavra-chave como elemento de ligação. Por exemplo, teoria de Newton (a palavra de é uma *stopword*, mas está no meio das palavras, ou seja, está ligando as palavras teoria e Newton).

Também fazem parte da família dos advérbios as locuções adverbiais. Denomina-se **locução adverbial** o conjunto de duas ou mais palavras que funciona como advérbio. De acordo com (Cunha & Cintra, 2001), as locuções adverbiais formam-se da associação de uma preposição com um substantivo, com um adjetivo ou com um advérbio. As locuções adverbiais podem ser:

- de afirmação (ou dúvida): com certeza, por certo, sem dúvida*;*
- de intensidade: de muito, de pouco, de todo;
- de lugar: à direita, à esquerda, à distância, ao lado, de dentro, de cima, de longe, de perto, em cima, para dentro, para onde, por ali, por aqui, por dentro, por fora, por onde, por perto;
- de modo: à toa, à vontade, ao contrário, ao léu, às avessas, às claras, às direitas, às pressas, com gosto, com amor, de bom grado, de cor, de má vontade, de regra, em geral, em silêncio,

em vão, gota a gota, passo a passo, por acaso;

- de negação: de forma alguma, de modo nenhum;
- de tempo: à noite, à tarde, à tardinha, de dia, de manhã, de noite, de quando em quando, de vez em quando, de tempos em tempos, em breve, pela amanhã;

Como os substantivos e adjetivos não são classificados como *stopwords,* as locuções adverbiais foram descartadas da lista. Além disso, como os advérbios foram acrescentados à lista, não foi necessário verificar quais das locuções adverbiais são formadas por uma preposição e um advérbio.

Os advérbios bastante, bem, junto, melhor, pior, quanto, certo não foram acrescentados porque foram classificados primeiramente como substantivos e/ou adjetivos em (Houaiss, 2004) e isto foi usado como regra para descartar qualquer palavra da lista de *stopwords.*

5.1.4 Preposições

Preposições são palavras invariáveis que relacionam dois termos de uma oração, de tal modo que o sentido do primeiro é explicado ou completado pelo segundo. As preposições podem ser:

- simples, quando expressar por um só vocábulo;
- compostas (ou locuções prepositivas), quando constituídas de dois ou mais vocábulos, sendo o último deles uma preposição simples (geralmente de).

Tais preposições se denominam também essenciais, para se distinguirem de certas palavras que, pertencendo normalmente a outras classes, funcionam às vezes como preposições e, por isso, se dizem preposições acidentais. De acordo com (Cunha & Cintra, 2001), são preposições acidentais as seguintes palavras: afora, conforme, consoante, durante, exceto, fora, mediante, menos, não obstante, salvo, segundo, senão, tirante, visto, etc. Essas palavras podem também ser advérbios, pronomes, numerais e adjetivos, e por esta razão as seguintes palavras foram descartadas da lista: conforme, consoante, mediante, obstante, salvo, segundo, tirante, visto.

Segundo (Cunha & Cintra, 2001), na língua portuguesa atual a preposição trás, que indica situação posterior, arcaizou-se. Ela é substituída pelas locuções atrás de e depois de; mais raramente, por sua sinônima após. O sentido originário desta preposição era "além de", que subsiste nos compostos Trás-os-Montes e trasanteontem. Esta preposição foi incluída na lista.

As locuções prepositivas não foram adicionadas à lista porque estas são constituídas de um substantivo ou um adjetivo ou um advérbio e uma preposição simples. Como o KEA separa as palavras, então as locuções prepositivas serão divididas, ou seja, as palavras que formam as locuções prepositivas serão tratadas individualmente. Como as preposições simples e os advérbios já foram adicionados anteriormente e também como as palavras que são classificadas primeiramente como substantivos ou adjetivos não são consideradas *stopwords,* então as locuções prepositivas foram descartadas da lista.

As locuções prepositivas que são formadas por um advérbio e uma preposição passaram a fazer parte da lista no momento em que os advérbios e as preposições foram adicionadas à lista. Foi verificado que todos os advérbios que formam uma locução prepositiva com uma preposição simples estão na lista de advérbios. Eis algumas locuções prepositivas: abaixo de, acerca de, acima de, a despeito de, adiante de, a fim de, além de, antes de, ao lado de, ao redor de, a par de, apesar de, a respeito de, atrás de, através de, de acordo com, debaixo de, de cima de, defronte de, dentro de, depois de, diante de, embaixo de, em cima de, em frente a, em frente de, em lugar de, em redor de, em torno de, em vez de, graças a, junto a, junto de, para baixo de, para cima de, para com, perto de, por baixo de, por causa de, por cima de, por detrás de, por diante de, por entre e por trás de.

5.1.5 Conjunções

Conjunções são os vocábulos gramaticais que servem para relacionar duas orações ou dois termos semelhantes da mesma oração. As conjunções podem ser:

a) Coordenativas, quando relacionam termos ou orações de idêntica função gramatical.

O vento **e** a chuva às vezes chegam juntos.
Ouvi primeiro **e** falai por derradeiro.

b) Subordinativas, quando ligam duas orações, uma das quais determina ou completa o sentido da outra.

Eram duas da manhã **quando** o adolescente chegou em casa.
Pediram-me **que** escolhesse o vinho.

As conjunções coordenativas dividem-se em: aditivas, adversativas, alternativas, conclusivas e explicativas.

As conjunções subordinativas classificam-se em: causais, concessivas, condicionais, finais, temporais, comparativas, consecutivas e integrantes. Conforme (Cunha & Cintra, 2001), a Nomenclatura Gramatical Brasileira distingue ainda, entre as conjunções subordinativas, as conformativas e as proporcionais.

Há numerosas conjunções formadas da partícula que antecedida de advérbios, de preposições e de particípios. São as chamadas locuções conjuntivas. Por exemplo: desde que, antes que, visto que e posto que.

As seguintes conjunções e locuções conjuntivas não foram adicionadas à lista: caso, conforme, ou seja, quer, por conseguinte, uma vez que, visto que, visto como, consoante, mesmo que, posto que, bem que, se bem que, salvo se, dado que, a fim de que, todas as vezes que, cada vez que, de forma que, de maneira que, de modo que, de sorte que, maior do que, menor do que, melhor do que, pior que, tanto quanto, bem como, à medida que, ao passo que, à proporção que, quanto mais… mais, quanto mais… tanto mais, quanto mais... menos, quanto mais... tanto menos, quanto menos... menos, quanto menos... tanto menos, quanto menos... mais e quanto menos... mais.

5.1.6 Consoantes e vogais

Por analogia com o conjunto de *stopwords* da língua inglesa utilizado pelo KEA, incluiu-se também o conjunto de consoantes e vogais na lista final. Apesar de não ser feita a justificativa para tal em (Witten et al., 1999), foi feita a suposição de que caso existam tais letras isoladas, esta não serão termos significativos.

5.2 Alterações efetuadas no radicalizador Portuguese Stemmer

Foi pensado inicialmente implementar diretamente o algoritmo de radicalização proposto em (Orengo & Huyck, 2001). Vale observar que uma parte fundamental do algoritmo se refere à utilização de exceções para regras de redução de sufixos, como descrito anteriormente. Contudo, o artigo original não lista as exceções que foram utilizadas na implementação do mesmo, portanto, nem todas as informações estavam disponíveis para a correta implementação deste radicalizador.

Através de contato direto com os autores, foi obtida uma referência para uma implementação na linguagem C++ do referido algoritmo feito pelos próprios (Rslp, 2004). Com base neste código fonte foi implementado um primeiro protótipo usando a linguagem PHP, para fazerem-se testes interativos.

Neste momento observou-se que a implementação fornecida continha pequenas discrepâncias em relação ao algoritmo descrito no artigo original. Algumas diferenças estava relacionada com as regras, com algumas poucas adições e modificações em relação ao artigo. Entretanto, havia duas alterações mais significativas:

- Primeiro, a ordem dos passos tinha sido modificada, colocando-se o passo de redução dos advérbios antes do passo de redução do feminino. Isto faz sentido, pois como advérbios terminados em -mente são palavras resultantes da composição de outras palavras, é preferível retirar o quanto antes o sufixo -mente para que ele possa ser reduzido em seguida pelo passo de redução do feminino. Por exemplo, a palavra verdadeiramente seria primeiro reduzida para a palavra verdadeira- para depois ser transformada para o masculino verdadeiro-, sendo finalmente reduzida para verdad- pelo passo de redução de sufixos nominais. Caso fosse mantida a seqüência original do algoritmo, tal palavra seria reduzida para verdadeira- apenas.

- A comparação de uma determinada palavra com a lista de exceções de uma dada regra acontecia de duas formas diferentes nesta nova implementação. Para alguns passos era sempre feita a comparação da palavra inteira contra as exceções. Em outros passos a comparação era realizada apenas contra o final da palavra, efetivamente definindo assim conjuntos de exceções que compartilham um dado sufixo. Isso também tem razão de ocorrer, pois às vezes é mais conveniente indicar que todo um conjunto de palavras que terminem com um dado sufixo seja uma exceção para uma determinada regra.

Essas duas modificações em relação ao algoritmo original fazem parte da implementação feita neste trabalho, que se refletiram em resultados satisfatórios na maioria dos casos.

Contudo, para várias palavras testadas o radical gerado não era o mais adequado, ocorrendo tanto problemas de *overstemming* quanto de *understemming*. A leitura feita desses problemas e que coincide com as conclusões de (Chaves, 2003), diz que quanto maior o número de palavras distintas processadas por um algoritmo de radicalização, maior a necessidade de se aumentar as listas de exceções do algoritmo. Além disso, o algoritmo proposto em (Orengo & Huyck, 2001) foi validado com uma lista de apenas 32000 palavras, sendo poucos vocábulos levando-se em conta a língua portuguesa. De fato, a versão mais recente do dicionário Houaiss (Houaiss, 2004) elenca 228000 verbetes.

Portanto, decidiu-se revisar cada uma das regras propostas do algoritmo Portuguese Stemmer. Para tanto, foi analisado o processo de formação morfológica das palavras e observadas as listas de sufixos formalmente definidas em gramáticas da língua portuguesa (Cunha & Cintra, 2001). Além disso, tornou-se necessário a utilização de uma lista mais completa e idealmente mais indicativa de palavras utilizadas em documentos científicos escritos na língua portuguesa.

Como o foco principal deste trabalho é processar dissertações e teses científicas, contou-se com a inestimável colaboração da Biblioteca Digital da UNICAMP (Libdig, 2004) para se ter acesso aos textos completos em formato PDF (*Portable Document Format*). Em um primeiro momento foram focadas as áreas das engenharias, totalizando 919 dissertações e teses.

Para geração de uma massa de palavras de teste foi utilizada a ferramenta `pdftotext` juntamente com comandos padrão de `Unix shell`, para extrair o texto puro (sem formatação

ou fórmulas e gráficos) para simples arquivos `.txt`, um para cada documento. Uma vez feito isto, foi gerado um segundo arquivo para cada texto, contendo o seu histograma individual (ou seja, a contagem de ocorrência de cada uma das palavras).

Todos os histogramas foram então consolidados em um único, obtendo-se assim uma listagem de 602014 termos distintos. É claro que o processo de extração de textos a partir dos documentos originais não é perfeito, ocorrendo a inclusão de termos que consideramos como ruído, sendo tipicamente:

- palavras que ficaram unidas, como por exemplo estudodecaso;
- palavras de outras línguas, como *string* e *conocimiento*;
- ou ainda comandos de formatação ou fórmulas matemáticas que foram incorretamente convertidas em seqüência de caracteres, como por exemplo øøø.

Na geração dos histogramas foram, porém, desconsiderados e retirados todos os algarismos e todos os símbolos de pontuação. É importante notar isso porque todas as palavras hifenizadas foram quebradas em dois ou mais termos. Neste momento não foi feita uma triagem maior dos caracteres alfabéticos (limitando-se aos utilizados na língua portuguesa), pois isto poderia excluir prematuramente nomes próprios e termos estrangeiros. Então, com base nessa grande lista de termos foram utilizados os termos mais freqüentes para cada sufixo das regras de redução, dando origem ao aumento nas listas de exceções.

Em termos comparativos, o algoritmo de Porter tal como adaptado para a língua portuguesa conta com a identificação de 176 sufixos distintos, que poderiam ser considerados como regras separadas. Contudo, este algoritmo não possui nenhuma exceção.

O algoritmo de radicalização Portuguese Stemmer tal como implementado em (Rslp, 2004), apresenta 242 regras, das quais 90 regras possuem exceções. No total existem 325 exceções.

A revisão efetuada no algoritmo de radicalização resultou em 406 regras, das quais 143 contêm listas de exceções. Totalizando 1050 exceções.

Durante o processo de revisão do algoritmo tornou-se necessário definir dois novos tipos

de regras, para aumentar o controle sobre o processo de redução. Para cada uma das três regras definidas foi atribuído um tipo P, N e S.

- Regra P: indica uma regra "positiva", na qual todas as palavras que tenham o dado sufixo são reduzidas, desde que tenham tamanho do radical mínimo especificado e que não estejam listadas na lista de exceções. Por exemplo:

"esa", 4, "ês ", { "*presa", "*defesa", "despesa", "sobremesa", "turquesa"}

Vale observar que nesta regra codificou-se explicitamente exceções terminadas com determinado sufixo com o caracter *. Por exemplo, na regras acima as palavras microempresa e autodefesa não serão reduzidas.

- Regra N: indica uma regra "negativa", na qual somente serão radicalizadas as palavras com sufixo especificado e que estejam numa lista de palavras válidas. Neste caso não é necessário definir o tamanho mínimo do radical.

"oa", "ão ", { "leitoa", "patroa", "leoa"}

Esta regra foi criada pela necessidade de tratar reduções em alguns poucos casos específicos.

- Regra S: indica uma regra de substituição que simplesmente troca uma palavra encontrada pelo radical correspondente.

"sonolência", "sono",

Vale a pena registrar que uma grande dificuldade encontrada no processo de validação das regras foi determinar se uma dada palavra deve ser reduzida ou não, pois dever ser levado em conta não só apenas a sua forma ortográfica, mas também a existência de outras semelhantes e ainda sua etimologia. Por exemplo, sargento que não tem nenhuma relação com o sufixo -ento de formas nominais, pois é derivada da palavra em francês *sergent*, que significa servidor.

O conjunto completo de regras está listado no APÊNDICE B e uma quantificação das melhorias resultantes de sua utilização estão na Seção 6.1.1.

5.3 Detalhes de implementação do algoritmo KEA

O algoritmo KEA foi implementado e disponibilizado em linguagem Java por seus autores, sendo distribuído como Software Livre (NZDL/KEA, 2004). Esta implementação foi feita de forma modular com a separação das funcionalidades em uma série de classes de Java.

Em particular, o algoritmo foi implementado como uma aplicação dos algoritmos para mineração de dados providos por uma biblioteca de métodos chamada WEKA, também disponível como Software Livre e descrita em (Witten & Frank, 2000).

Na parte específica do KEA relacionada com a linguagem dos documentos a serem processados existem duas classes de interesse: a primeira, que lista o conjunto particular de *stopwords* a serem utilizadas no processo de identificação de frases candidatas, e a segunda, que implementa o algoritmo de radicalização a ser empregado tanto na etapa de treinamento quanto na de extração. Estas implementações foram substituídas por novas classes de Java que implementam o suporte para a língua portuguesa proposto neste trabalho, e que se encontram disponíveis em (Dias, 2005).

O pacote original do KEA contém a implementação das classes citadas acima tanto para a língua inglesa quanto para a língua alemã. Estas implementações foram substituídas por novas classes de Java que implementam o suporte para a língua portuguesa proposto ao longo desta dissertação. A implementação feita neste trabalho está disponibilizada sob a mesma licença GPL em (Dias, 2005).

Resultados comparativos estão descritos na Seção 6.1.2.

6 RESULTADOS E CONTRIBUIÇÕES

Neste capítulo relataremos brevemente alguns procedimentos e medidas feitas para avaliar tanto as melhorias efetuadas no algoritmo de radicalização, o Portuguese Stemmer, quanto a adaptação feita no KEA com o objetivo de extrair automaticamente palavras-chave de textos na língua portuguesa.

Em seguida são destacadas as principais contribuições dadas pelo presente trabalho e traçadas conclusões sobre o mesmo.

Finalmente, encerramos este capítulo comentando sobre possíveis melhorias e possibilidades de trabalhos futuros.

6.1 Teste e avaliação do algoritmo KEA adaptado para a língua portuguesa

Nesta seção primeiro avaliamos apenas o algoritmo revisado de radicalização de palavras e depois analisamos o processo inteiro de extração automática de palavras-chave.

6.1.1 Comparação dos algoritmos de radicalização para a língua portuguesa

Diversos experimentos foram feitos para comparar a efetividade do algoritmo de radicalização revisado em relação ao algoritmo original e também ao radicalizador de Porter.

Um primeiro experimento consistiu da criação de sete listas de palavras:

- a primeira lista é formada pelo conjunto de todos os termos distintos constantes no histograma global das 919 teses, resultando em 602014 termos distintos, sobre este conjunto não foi feita nenhuma validação: estes são os termos brutos (incluindo ruído, mas sem algarismos e pontuação). Também não foram impostos limites para o tamanho dos termos, sendo tamanho mínimo de um caracter.
- a segunda lista correspondia aos 32000 termos mais freqüentes da lista bruta que naturalmente conterão uma quantidade bem menor de ruído, já que termos incorretos ou em outras linguagens ocorrerão com uma freqüência muito menor do que palavras da língua portuguesa.
- a terceira lista foi construída com base em uma filtragem da lista bruta de termos limitando às palavras escritas somente em minúsculas (a geração do histograma consolidou em minúsculas todas as palavras que ocorriam com variação de caixa, ou seja, as palavras Amor, AMOR e amor são contados como ocorrências do termo amor). Além disso termos desta lista foram limitados à utilizar somente os acentos válidos na língua portuguesa e seu tamanho máximo definido em 23 caracteres. Esta lista resultou em 262610 termos.
- a quarta lista foi construída com base nos 32000 termos mais freqüentes da terceira lista, portanto representando uma massa de palavras com bem menos ruído.
- a quinta lista contou com termos que somente ocorreram com a inicial em maiúscula e as demais em minúscula, também restritas aos acentos válidos na língua portuguesa e com tamanho máximo de 18 caracteres. Tal construção resultou em uma lista principalmente povoada por nomes próprios e contendo 67216 termos.
- a sexta lista foi formada pelos 32000 termos mais comuns da lista de nomes próprios.
- a sétima lista consistiu do vocabulário de 32000 palavras disponíveis em (Snowball, 2003).

Cada uma dessas listas de termos foi reduzida pelos três radicalizadores e os resultados são mostrados na Tabela 6 abaixo.

TABELA 6 - Comparativo de redução de vocabulário.

	1	2	3	4	5	6	7
lista	bruto	bruto freq.	palavras	palavras freq.	nomes	nomes freq.	snowball
tamanho	602014	32000	262610	32000	67216	32000	32000
Porter	479958 (80%)	18328 (57%)	158803 (60%)	16525 (52%)	62181 (93%)	29839 (93%)	16797 (52%)
Portuguese Stemmer	440466 (73%)	16565 (52%)	144130 (55%)	14738 (46%)	59567 (89%)	28747 (90%)	15216 (48%)
Portuguese Stemmer Revisado	435277 (72%)	15799 (49%)	139477 (53%)	13878 (43%)	59267 (88%)	28624 (89%)	14379 (45%)

Pelos dados da Tabela 6 pode-se observar que o desempenho geral do Portuguese Stemmer revisado é melhor do que o Portuguese Stemmer e o do Porter. Em particular, os resultados são melhores justamente para as listas que contém menos ruídos, ou seja, as que contém os termos mais freqüentes e também o vocabulário Snowball.

Uma outra estimativa feita foi da redução de tamanho em textos normais. Para tanto foram escolhidas aleatoriamente 30 teses de diversas áreas e aplicado os algoritmos de radicalização. Quando utilizado o algoritmo de Porter obteve-se em média uma redução de 12,1%. Já para o Portuguese Stemmer obteve-se uma média de redução de 15,1%. E para o Portuguese Stemmer revisado obteve-se uma média de 15,2%.

O algoritmo de Porter naturalmente apresenta os piores resultados porque não usa listas de exceções e não leva em conta as relações entre as diferentes classes gramaticais da língua portuguesa. As duas versões do Portuguese Stemmer (original e revisada) apresentam resultados relativamente bons frente a palavras de uso comum. Contudo, mesmo com o trabalho feito na revisão das regras e no aumento do número de exceções, o rendimento do radicalizador não aumentou significativamente. De fato, é de se supor que abordagens como esta, que se utilizam de um conjunto limitado de regras e exceções manualmente construídas apresente sempre limitações para radicalizar palavras, em especial aquelas de uso menos freqüente ou que apresentem variações no radical (por exemplo: emitir e emissão). Considera-se que uma abordagem baseada em dicionários, em que é feita uma listagem exaustiva de palavras, suas flexões e respectivos radicais possa apresentar melhores resultados, a despeito do esforço de

construção de tal dicionário.

6.1.2 Avaliação da extração automática de palavras-chave na língua portuguesa

Para avaliar o processo de extração automática de palavras-chave de textos na língua portuguesa foram criados diversos modelos com base em conjuntos de treinamento distintos, variando os parâmetros para a criação dos modelos e também para a composição dos conjuntos de treinamento e extração.

Um primeiro experimento consistiu em selecionar 200 dissertações e teses de áreas distintas (Libdig, 2004), com o objetivo de esclarecer os seguintes pontos:

- Qual o número adequado de documentos a ser utilizado para a composição do conjunto de treinamento, com base no qual é construído o modelo a ser usado posteriormente na extração de palavras-chave?
- Qual o número ideal de palavras-chave para serem extraídas de um dado documento?
- A presença da ficha catalográfica como parte do texto de um documento afeta ou não a extração de palavras-chave?
- Qual é a qualidade das palavras-chave extraídas automaticamente, tanto em relação às palavras-chave associadas pelos próprios autores como em relação à uma inspeção manual do texto?

Analisando as teses observou-se que a grande maioria possuía a ficha catalográfica como parte do documento, pois trabalhou-se em cima do formato definitivo de teses e dissertações da Biblioteca Digital da UNICAMP, que já haviam passado pelo processo de catalogação manual pela equipe da Biblioteca Central. Para garantir a uniformidade dos testes e realmente quantificar a influência da presença da ficha catalográfica, optou-se por trabalhar com 200 das 916 dissertações e teses, todas estas com a respectiva ficha e cujo texto não contivesse muito ruido (resultante do processo de extração de texto a partir dos arquivos originais no formato PDF). Apenas uma parte do total de teses e dissertações foi utilizada porque estimou-se que esses 200 documentos representam uma amostra representativa do conjunto inteiro.

A presença da ficha catalográfica é importante para indicar quais foram as palavras-chave escolhidas pelos autores, e que servirão como parâmetro de avaliação das palavras-chave extraídas automaticamente pelo KEA. É interessante observar que existe grande variação na quantidade de palavras-chave dos autores associadas aos documentos, variando de 3 a 7 para cada documento. Em média as palavras-chave são compostas por três palavras, porém em algumas situações até cinco termos compunham uma palavra-chave.

Os 200 documentos selecionados foram divididos em dois conjuntos denominados I e II. Cada conjunto possuía 100 documentos, que foram utilizados tanto para a criação do modelo quanto para a extração de palavras-chave. Vale dizer que no conjunto I predominou-se textos da área da Educação e no conjunto II textos da área de Engenharia Elétrica.

Dado um conjunto, 40 de seus documentos foram reservados apenas para a construção dos modelos. Os 60 documentos restantes foram divididos em três grupos de 20 documento**s,** denominados A, B e C, para efetuar as extrações em massa. Observou-se que o consumo de memória e tempo de execução do algoritmo KEA era proporcional ao número de documentos envolvidos na extração em massa. Portanto optou-se por um número de 20 documentos para se extrair as palavras-chave por vez. Na prática, a extração de palavras-chave será realizada em apenas um documento por vez.

Para cada conjunto I e II, foram usados os 40 documentos para construir modelos de 10, 20, 30 e 40 documentos com a característica de que um modelo de 20 continha os documentos do modelo de 10 e assim por diante. Isso foi feito para garantir que um modelo com mais documentos contivesse os mesmos dados de modelos com menos documentos.

Para cada grupo de extração (A, B e C) foram realizadas extrações com os quatro modelos de 10, 20, 30 e 40 documentos. Uma vez definido um par de grupo de extração e de modelo, foram extraídas inicialmente cinco palavras-chave e computado o número de acertos, a variância do número de acertos e o tempo de extração. Para esse mesmo par foram extraídas também 10, 15 e 20 palavras-chave.

O número de acertos foi computado com uma média aritmética entre o número de coincidências entre as palavras-chave escolhidas automaticamente para uma tese e as selecionadas pelos próprios autores.

Dado um modelo e um conjunto de documentos para extração, o número de palavras-chave a serem extraídas não influi no tempo de processamento. Isto se deve ao fato do algoritmo KEA precisar primeiro extrair todas as frases candidatas de um documento para depois pontuá-las. Então, as palavras-chave com mais pontuação serão indicadas pelo KEA, portanto não existe diferença em selecionar somente as 5 palavras-chave ou as 10 palavras-chave mais pontuadas. De fato, foi observado na prática que o tempo de execução para extração de 5, 10, 15 ou 20 palavras-chave é praticamente o mesmo para um dado grupo de extração e modelo, de forma que está ilustrado nas Tabelas 7, 8, 9 e 10 apenas um valor de tempo de processamento para facilitar a visualização dos dados.

TABELA 7: Testes do Conjunto I com Fichas Catalográficas. **Ac =** número médio de coincidências entre as palavras-chave extraídas automaticamente e as palavras-chave definidas pelos autores; **Var =** variância do número de coincidências e **Tem** = tempo de processamento.

		Modelo10			Modelo20			Modelo30			Modelo40		
		Ac	Var	Tem	Ac	Var	Tem	Ac	Var	Tem	Ac	Var	Tem
Extração A (6.3MB)	5	0.95	±0.94	3m25s	0.80	±0.83	3m27s	0.80	±0.83	3m29s	0.90	±0.85	3m32s
	10	1.55	±1.39		1.55	±1.10		1.60	±1.10		1.60	±1.14	
	15	1.85	±1.39		1.80	±1.28		1.75	±1.29		1.65	±1.18	
	20	2.20	±1.47		2.25	±1.52		2.20	±1.47		2.15	±1.46	
Extração B (4.3MB)	5	0.85	±1.04	2m33s	1.20	±1.20	2m32s	1.15	±1.23	2m37s	1.20	±1.20	2m37s
	10	1.45	±1.00		1.70	±1.17		1.75	±1.12		1.75	±1.12	
	15	1.75	±1.02		2.10	±1.12		2.15	±1.04		2.15	±1.04	
	20	1.85	±1.04		2.35	±1.18		2.30	±1.08		2.35	±1.09	
Extração C (7.1MB)	5	1.00	±0.73	4m06s	0.90	±0.79	4m28s	0.90	±0.72	3m56s	0.85	±0.81	4m00s
	10	1.60	±1.19		1.60	±1.27		1.55	±1.43		1.50	±1.43	
	15	2.00	±1.30		1.75	±1.37		1.70	±1.42		1.65	±1.39	
	20	2.15	±1.35		1.95	±1.39		1.90	±1.45		1.90	±1.45	

Os resultados apresentados nas Tabelas 7 e 8 foram realizados com documentos completos, ou seja, com as respectivas fichas catalográficas e os das Tabelas 9 e 10 foram realizados com documentos sem fichas catalográficas. Abaixo é mostrado o desempenho do algoritmo KEA sem as fichas catalográficas.

TABELA 8: Testes do Conjunto II com Fichas Catalográficas. **Ac =** número médio de coincidências entre as palavras-chave extraídas automaticamente e as palavras-chave definidas pelos autores; **Var** = variância do número de coincidências e **Tem** = tempo de processamento.

		Modelo10			Modelo20			Modelo30			Modelo40		
		Ac	Var	Tem	Ac	Var	Tem	Ac	Var	Tem	Ac	Var	Tem
Extração A (6.1MB)	5	0.70	±0.73	3m54s	0.70	±0.80	3m53s	0.70	±0.80	3m45s	0.70	±0.73	3m50s
	10	1.15	±0.88		1.00	±0.92		1.05	±0.89		1.20	±0.95	
	15	1.65	±1.23		1.50	±1.24		1.45	±1.19		1.65	±1.18	
	20	1.85	±1.35		1.95	±1.15		1.90	±1.17		2.05	±1.19	
Extração B (4.0MB)	5	0.70	±0.57	1m47s	0.80	±0.62	1m49s	0.80	±0.62	1m55s	0.70	±0.57	1m55s
	10	1.00	±0.92		1.15	±0.88		1.15	±0.81		1.05	±0.89	
	15	1.30	±1.13		1.30	±1.08		1.25	±1.02		1.35	±1.04	
	20	1.45	±1.05		1.60	±1.27		1.45	±1.15		1.50	±1.05	
Extração C (4.8MB)	5	0.35	±0.59	2m41s	0.45	±0.69	2m43s	0.35	±0.59	2m44s	0.45	±0.69	2m43s
	10	1.10	±1.52		1.15	±1.31		1.10	±1.29		1.20	±1.28	
	15	1.40	±1.73		1.60	±1.57		1.55	±1.57		1.50	±1.57	
	20	1.55	±1.90		1.85	±1.76		1.65	±1.63		1.55	±1.73	

TABELA 9: Testes do Conjunto I Sem Fichas Catalográficas. **Ac =** o número médio de coincidências entre as palavras-chave extraídas automaticamente e as palavras-chave definidas pelos autores; **Var** = variância do número de coincidências e **Tem** = tempo de processamento.

		Modelo10			Modelo20			Modelo30			Modelo40		
		Ac	Var	Tem	Ac	Var	Tem	Ac	Var	Tem	Ac	Var	Tem
Extração A (6.3MB)	5	0.55	±0.69	3m24s	0.50	±0.69	3m26s	0.85	±0.88	3m29s	0.85	±0.88	3m31s
	10	1.05	±1.19		0.90	±0.97		1.05	±0.89		1.05	±0.89	
	15	1.25	±1.21		1.10	±1.17		1.25	±1.12		1.15	±1.04	
	20	1.40	±1.23		1.30	±1.13		1.45	±1.28		1.45	±1.23	
Extração B (5.2MB)	5	0.50	±0.61	3m13s	0.50	±0.61	3m14s	0.45	±0.60	3m16s	0.55	±0.69	3m20s
	10	0.70	±0.08		0.70	±0.80		0.70	±0.73		0.75	±0.72	
	15	0.85	±0.81		1.00	±0.97		0.90	±0.85		1.10	±1.02	
	20	1.15	±1.04		1.25	±1.16		1.20	±0.95		1.35	±0.99	
Extração C (7.1MB)	5	0.90	±0.64	3m58s	0.70	±0.66	3m58s	0.90	±0.72	4m00s	0.90	±0.72	4m20s
	10	1.20	±0.77		1.05	±0.69		1.05	±0.83		1.05	±0.83	
	15	1.30	±0.92		1.20	±0.77		1.35	±0.93		1.45	±1.00	
	20	1.45	±0.89		1.25	±0.85		1.45	±1.00		1.55	±1.15	

Uma primeira observação consiste no tempo de extração ser diretamente proporcional ao tamanho total dos documentos extraídos em massa. Isso explica a diferença de tempo de extração entre os grupos de extração A, B e C. Na Tabela 7, o tempo de extração do grupo C é maior do que o tempo de A e o de B. Isso acontece porque a entrada de dados é maior, portanto mais frases candidatas a serem testadas com base no modelo são geradas.

Observou-se também que o tempo de extração é independente do modelo utilizado, ou seja, do número de documentos de treinamento utilizados para construir cada modelo. Apesar dos modelos serem maiores, o tempo para pontuar uma palavra-chave é constante, portanto para o mesmo conjunto de documentos de extração, a análise por modelos diferentes ocorre em tempos praticamente equivalente. Na verdade, o que influencia no tempo de extração é o tamanho dos documentos a serem extraídos e não a complexidade do modelo.

TABELA 10: Testes do Conjunto II sem Fichas Catalográficas. **Ac =** número médio de coincidências entre as palavras-chave extraídas automaticamente e as palavras-chave definidas pelos autores; **Var** = variância do número de coincidências e **Tem** = tempo de processamento.

		Modelo10			Modelo20			Modelo30			Modelo40		
		Ac	Var	Tem	Ac	Var	Tem	Ac	Var	Tem	Ac	Var	Tem
Extração A (6.0MB)	5	0.65	±0.75	3m42s	0.65	±0.67	3m44s	0.65	±0.67	3m51s	0.55	±0.69	3m54s
	10	0.90	±0.91		0.85	±0.75		1.10	±0.92		0.70	±0.73	
	15	1.00	±0.92		0.85	±0.75		1.10	±0.92		1.00	±0.86	
	20	1.05	±1.00		0.95	±0.76		1.10	±0.85		1.15	±0.88	
Extração B (3.9MB)	5	0.40	±0.68	1m52s	0.55	±0.51	1m53s	0.35	±0.59	1m55s	0.40	±0.50	1m56s
	10	0.50	±0.83		0.60	±0.50		0.65	±0.67		0.55	±0.51	
	15	0.50	±0.83		0.80	±0.70		0.70	±0.73		0.70	±0.66	
	20	0.55	±0.83		0.80	±0.70		1.05	±0.83		0.75	±0.64	
Extração C (4.8MB)	5	0.3	±0.66	2m36s	0.20	±0.52	2m37s	0.30	±0.66	2m44s	0.25	±0.55	2m43s
	10	0.45	±0.89		0.50	±0.89		0.50	±0.89		0.5	±0.83	
	15	0.55	±1.00		0.85	±1.09		0.65	±0.88		0.8	±0.95	
	20	0.6	±0.99		0.90	±1.17		0.90	±1.17		0.9	±1.12	

Os modelos que tiveram maior precisão (maior número de acertos) foram primeiramente o modelo 20 seguido do modelo 10. Os modelos 30 e 40 ficaram abaixo dos dois modelos anteriores, porém com resultados equivalentes entre si. Isso é semelhante ao resultado descrito em (Witten et al., 1999), o qual diz que modelos com poucos documentos são bons,

apresentando resultados significativos.

Após a análise das Tabelas 7, 8, 9 e 10, verificou-se que o modelo 20 parece ser o ideal e que os modelos 30 e 40 não apresentaram nenhum ganho em relação ao modelo 20. Nota-se que apresentam até um decréscimo, o que indica que aumentar o número de documentos de treinamento para a construção dos modelos não traz melhorias em termos de precisão de resposta.

Como regra geral quanto mais palavras-chave extraídas, melhor é a precisão. Constatou-se que extrair 5 palavras-chave é insuficiente, que 10 ainda não traz resultados satisfatórios e que aumentando de 10 para 15 há uma melhoria significativa. De 15 para 20 também há uma melhora, só que um pouco menor do que de 10 para 15. Em resumo, considera-se que uma extração com mais de 15 palavras-chave seja adequada, mas é provável que o ganho incremental de precisão para quantidades maiores seja proporcionalmente menor. Por outro lado, como foi observado anteriormente, o tempo de execução para se extrair 5, 10, 15 ou 20 palavras-chave é praticamente o mesmo.

Outra comparação envolve os resultados mostrados nas Tabelas acima, entre a extração de palavras-chave de documentos com ficha e documentos sem ficha catalográfica. Fica evidente que ocorre precisão maior quando estão presentes as fichas. Isso se deve ao fato de que, nas dissertações e teses que contém fichas catalográficas, existe a ocorrência das palavras-chave no início do texto, o que faz com que as palavras-chave indicadas pelo autor tenha uma influência alta na pontuação final. Já em documentos sem ficha catalográfica, as palavras-chave escolhidas pelos autores podem ocorrer mais afastados do início do texto. Portanto, pela própria concepção do KEA, já era esperado de início esse resultado. Contudo para uso prático, a não existência de uma ficha não limita a sua utilização, apenas reduz a sua precisão com relação a palavras-chave definidas pelos autores

Outra comparação realizada foi entre os resultados dos conjuntos I e II tanto com ficha e sem ficha catalográfica. Observa-se que houve uma redução significativa de acertos no conjunto II. Uma provável explicação é de que os documentos do conjunto I, em sua maioria documentos da área de Educação, possuem palavras-chave mais simples, ou seja, com menos quantidade de termos e compostas de palavras de uso mais freqüente. Em contrapartida, os documentos do conjunto II, em sua maioria da área de Engenharia Elétrica, possuem palavras-chave mais longas

e com termos de uso mais específico, o que resultou em um resultado de precisão inferior.

Quando comparados com as mesmas medidas feitas em (Witten et al., 1999), tanto para extração de 5 quanto de 15 palavras-chave, a precisão obtida nestes testes são compatíveis com os valores obtidos para a língua inglesa, o que mostra que os resultados para a língua portuguesa são aceitáveis, mesmo quando obtidos a partir de um conjunto de documentos bastante diverso.

Finalmente, foi feita uma breve inspeção nas palavras-chave geradas automaticamente para três dissertações, cujas palavras-chave geradas automaticamente puderam ser validadas pelos próprios autores. Para cada documentos foram feitas extrações em quatro situações: com e sem ficha catalográfica, aplicadas contra modelos treinados com 20 documentos, os mesmos gerados anteriormente para o conjunto I e o conjunto II. Para cada situação foram extraídas 15 palavras-chave que foram julgadas relevantes ou não-relevantes pelo próprio autor. Os resultados estão tabulados nas Tabela 11, 12 e 13, onde as palavras selecionadas como representativas pelos próprios autores estão em negrito.

TABELA 11: Resultado da extração de palavras-chave da dissertação Análise Topológica de Modelo Implícito (Malheiros, 2002).

Com Ficha Catalográfica		Sem Ficha Catalográfica	
Conjunto I	Conjunto II	Conjunto I	Conjunto II
pontos críticos	**primitivas**	**primitivas**	**primitivas**
primitivas	**pontos críticos**	**pontos críticos**	superfície
superfície	implícita	implícita	**superfície implícita**
superfície implícita	**superfície implícita**	**superfície implícita**	**conjuntos de nível**
regiões de influência	**regiões de influência**	**regiões de influência**	**pontos críticos**
conjuntos de nível	limiar	limiar	**regiões de influência**
função implícita	**conjuntos de nível**	**conjuntos de nível**	**função implícita**
malha	**topologia**	**topologia**	malha
modelo implícito	**modelo implícito**	**Implicit Surfaces**	**modelo implícito**
topologia	**função implícita**	duas primitivas	topológica
limiar	malha	esqueleto	esféricas
topológica	**algoritmo**	**modelo implícito**	**primitivas esféricas**
esféricas	configuração	**função implícita**	**suporte local**
críticos da função	topológica	malha	Luiz Henrique
numérica	Surfaces	configuração	Henrique de Figueiredo

TABELA 12: Resultado da extração de palavras-chave da dissertação Leitura Significativa no Ensino Superior: A Busca da Formação Integral do Universitário (Martins, 2005).

Com Ficha Catalográfica		Sem Ficha Catalográfica	
Conjunto I	Conjunto II	Conjunto I	Conjunto II
leitura significativa	**leitura significativa**	**leitura significativa**	**leitura significativa**
universitários	**universitários**	**universitários**	**universitários**
UNIVATES	UNIVATES	UNIVATES	aula
ensino superior	ensino	Vale do Taquari	UNIVATES
educação	**educação**	aprender	Taquari
Porto Alegre	**ensino superior**	aula	**ler**
estudante da UNIVATES	**estudante da UNIVATES**	**ler**	Português
docente	acredito	Português	Vale do Taquari
acredito	Porto Alegre	aulas de Português	ensino
missão	docente	escrita	**educação**
Martins	conviver	op	**ensino superior**
palavras	cursos	sala de aula	acredito
formação integral	missão	**estudante universitário**	Porto Alegre
conviver	professor	**ensino superior**	cursos
compreensão	Martins	pedagógica	missão

De acordo com (Witten et al., 1999), as palavras-chave determinadas pelo autor e as palavras-chave extraídas automaticamente são bastante similares, mas não é muito difícil adivinhar quais são as palavras-chave dos autores. Pode-se verificar nas Tabelas 11, 12 e 13 que o KEA escolhe diversas palavras-chave boas, mas também escolhe algumas que são improváveis dos autores usarem, por exemplo, na Tabela 12, as palavras compreensão, acredito e especialmente op. Apesar destas anomalias, as palavras-chave extraídas automaticamente fornecem uma descrição adequada das três dissertações. No caso em que nenhuma palavra-chave especificada pelo autor estiver disponível, as escolhas do KEA poderiam ser um recurso valioso para alguém encontrar estas três dissertações na primeira busca.

TABELA 13: Resultado da extração de palavras-chave desta dissertação.

Com Ficha Catalográfica		Sem Ficha Catalográfica	
Conjunto I	Conjunto II	Conjunto I	Conjunto II
língua portuguesa	**palavras-chave**	**algoritmo**	**palavras-chave**
palavras-chave	**língua portuguesa**	**língua portuguesa**	**língua portuguesa**
extração automática	**extração automática**	documentos	**KEA**
automática de palavras-chave	automática de palavras-chave	palavras	documentos
dissertações e teses	dissertações e teses	**palavras-chave**	advérbios
Engenharia Elétrica	área das Engenharias	**radical**	**radical**
área das Engenharias	palavras-chave na língua	**stopwords**	**stopwords**
palavras-chave na língua	Mauro Sérgio Miskulin	**algoritmo de radicalização**	preposições
Mauro Sérgio Miskulin	TESES DA ÁREA	**extração automática**	pronomes
Computação	SP Prof	advérbios	Portuguese
assunto	Membro Externo	KEA	**Stemmer**
Disponível	querida	pronomes	**Portuguese Stemmer**
TESES DA ÁREA	amor	preposições	frases
SP Prof	Disponível	**Stemmer**	**radicalizador**
Membro Externo	**Computação**	frases	frases candidatas

É interessante observar que mesmo devido à grande variabilidade de construção dos modelos, em todas as situações, ou seja, para cada uma das dissertações e nas quatro extrações efetuadas, o algoritmo KEA foi capaz de identificar palavras-chave relevantes. Mas ainda nota-se a ausência de palavras de uso comum, o que reflete a importância da medida de raridade das palavras. Por outro lado, é possível notar a alteração entre os modelos com e sem ficha catalográfica, indicando também a importância da segunda medida que é a distância em que as palavras ocorrem no texto.

Para finalizar, na Tabela 13, observou-se que os resultados foram ainda melhores com modelos sem ficha catalográfica, o que sugere que a presença ou não da ficha catalográfica em média não tem um efeito tão grande na seleção de palavras-chave adequadas.

6.2 Principais contribuições

Uma das principais contribuições desta dissertação foi a proposta de um algoritmo de extração automática de palavras-chave para a língua portuguesa e a avaliação da sua aplicação em dissertações e teses, em sua maioria da área das Engenharias, obtidas da Biblioteca Digital da UNICAMP.

Uma outra contribuição foi a revisão do algoritmo de radicalização Portuguese Stemmer. Com isto foi possível obter um pequeno acréscimo na precisão da radicalização e na redução dos erros de *overstemming* e *understemming* quando aplicado ao processo de radicalização de palavras da língua portuguesa.

Também foi elaborada uma lista de *stopwords* contendo 336 termos da língua portuguesa, e cuja construção foi totalmente justificada.

Estima-se que este estudo pode ser considerado relevante na área de Processamento de Linguagem Natural, pois ainda há poucos trabalhos relacionados à extração automática de palavras-chave de textos para a língua portuguesa.

6.3 Trabalhos futuros

Certamente, ainda é possível fazer grandes melhorias na área de radicalização de palavras na língua portuguesa, em particular vinculando-se os algoritmos de radicalização a dicionários manualmente construídos que armazenem os radicais específicos de cada palavra.

Outro campo que merece atenção mais profunda se refere à exploração de informações de domínios específicos de conhecimento para geração de modelos mais precisos de extração de palavras-chave (Frank et al., 1999).

Espera-se futuramente integrar este algoritmo de extração automática com forma de aprimorar os mecanismos de busca e de categorização de documentos para os sistemas Nou-Rau e Rau-Tu. Finalmente, uma possível área de aprofundamento de estudo seria utilizar também técnicas de análise sintática e semântica como forma de aprimorar a extração de palavras-chave.

REFERÊNCIAS BIBLIOGRÁFICAS

AIRES, R. V. X. et al. Combining Multiple Classifiers to Improve Part of Speech Tagging: A Case Study for Brazilian Portuguese. In: SBIA'2000, Atibaia: [s.n.], 2000. p. 20-22.

AMERICAN. **American Heritage Dictionary, The**. 2. ed. Boston: Houghton Mifflin, 1991. 1564 p.

ARAMPATZIS, A. T. et al. Linguistically-motivated Information Retrieval. In: **Encyclopedia of Library and Information Science**. Nova York: Marcel Dekker, v. 69, 2000. p. 201-222.

BAEZA-YATES, R.; RIBEIRO-NETO, B. **Modern Information Retrieval**. Wokingham: Addison-Wesley, 1999. 513 p.

BICK, E. Structural lexical heuristics in the automatic analysis of portuguese. In: **Proceedings of the 11th Nordic Conference on Computational Linguistics (Nodalida'98)**. København: [s.n.], 1998. p. 44-56.

BOD, R. **Enriching Linguistics with Statistics**: Performance Models of Natural Language. Tese de Doutorado. Institute for Logic, Language and Computation (ILLC), University of Amsterdam. Holanda: Academiche Pers, 1995. 143 p.

BRANDOW, R.; KITZE, K.; RAU, L. R. Automatic condensation of eletronic publications by Sentence Selection. In: **Information Processing and Management**. [S.l.]: [s.n.], v. 31, n. 5, 1994. p. 675-689.

BROWN. Disponível em: <http://helmer.aksis.uib.no/icame/brown/bcm.html>. Acesso em: 08 out. 2004.

CHAVES, M. S. Um estudo e apreciação sobre algoritmos de stemming. In: **IX JORNADAS IBEROAMERICANAS DE INFORMÁTICA**. Cartagena de Indias, 2003.

COUTINHO, I. L. **Gramática Histórica**. Rio de Janeiro: Livraria Acadêmica, 1954.

CUNHA, C.; CINTRA, L. F. L. **Nova Gramática do Português Contemporâneo**. 3 ed. Rio de Janeiro: Nova Fronteira, 2001. 748 p.

DE LUCCA, J. L.; NUNES, M. G. V. Lematização versus Stemming. **Série de Relatórios Técnicos do NILC - ICM-USP**, 2002. 16 p.

DIAS. Disponível em: <http://ensino.univates.br/~mald/>. Acesso em: 17 mai. 2005.

DOMINGOS, P.; PAZZANI, M. On the optimality of the simple Bayesian classifier under zero-one loss. In: **Journal Machine Learning**. [S.l.]: [s.n.], v. 29, n. 2-3, 1997. p. 103-130.

DUMAIS, S. T. et al. Inductive Learning Algorithms and Representation for Text Categorization. In: **Proceedings of the 7th international conference on Information and Knowledge Management**. [S.l.]: [s.n.], 1998. p. 148-155.

FAYYAD, U. M.; IRANI, K. B. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In: **Proceedings of the 13th International Joint Conference on Artificial Inteligence**. [S.l.]: [s.n.], 1993. p. 1022-1027.

FRANK, E. et al. Domain-specific keyphrase extraction. In: **Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence**. São Francisco: Morgan Kaufmann Publishers, 1999. p. 668-673.

FRANTZ, V.; SHAPIRO, J.; VOISKUNSKII, V. **Automated Information Retrieval: Theory and Methods**. San Diego: Academic Press, 1997. 365 p.

GAUCH, S.; FUTRELE, R. Experiments in Automatic Word Class and Word Sense Identification for Information Retrieval. In: **Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval**. [S.l.]: [s.n.], 1994. p. 425-434.

GONZALEZ, M.; LIMA, V. L. S. Recuperação de Informação e Processamento da Linguagem Natural. In: XXIII Congresso da Sociedade Brasileira de Computação. **Anais da III Jornada de Mini-Cursos de Inteligência Artificial**. Campinas: [s.n.], v. III, 2003. p. 347-395.

HARMAN, D. How effective is suffixing? In: **Journal of the American Society for Information Science**. [S.l.]: [s.n.], v. 42, n. 1, 1991. p. 7-15.

HOUAISS. Disponível em: <http://houaiss.uol.com.br/busca.jhtm>. Acesso em: 09 out. 2004.

HTDIG. Disponível em: <http://htdig.org>. Acesso em 20 fev. 2004.

HULL, D. A. Stemming Algorithms: A Case Study for Detailed Evaluation. In: **Journal of the American Society for Information Science**. [S.l.]: [s.n.], v. 47, n.1, 1996. p. 70-84.

JACKSON, P.; MOULINIER, I. Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization. Amsterdan: John Benjamins Publishing Company. 2002, v. 5, 225 p.

JOHNSON, F. C. et al. The application of linguistic processing to automatic abstract generation. In: **Journal of Document and Text Management**. [S.l.]: [s.n.], v. 1, n. 3, 1993. p. 15-42.

KRENN, B.; SAMUELSON, C. The linguist's guide to statistics. University of Saarland, 1997. 172 p. Disponível em: <http://www.coli.uni-sb.de/~krenn/edu.html>. Acesso em: 28 set. 2004.

KROVETZ, R. Viewing morphology as an inference process. In: **Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval**. Nova York: ACM Press, 1993. p. 191-202.

KUCERA, H.; FRANCIS, W. N. Computational analysis of present-day American English. Providence: Brown University Press, 1967.

KUPIEC, J.; PEDERSEN, J. CHEN, F. A trainable document summarizer. In: **Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**. Nova York: ACM Press, 1995. p. 68-73.

LIBDIG. Disponível em: <http://libdigi.unicamp.br/>. Acesso em: 12 set. 2004.

LOVINS, J. B. Development of a Stemming Algorithm. In **Mechanical Translation and Computational** Linguistics. [S.l.]: [s.n.], 1968. 11, p. 22-31.

MANDALA, R.; TOKUNAGA, T.; TANAKA, H. Complemeting WordNet with Roget's and corpus-based thesauri for information retrieval. In: **Proceedings of the 9th conference on European chapter of the Association for Computation Linguistics**. Morristown: Association for Computation Linguistics, 1999. p. 94-101.

MALHEIROS, M.G. **Análise Topológica de Modelo Implícito**. Dissertação de Mestrado. Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, 2002. 90 p.

MARTINS, S. M. **Leitura Significa no Ensino Superior: A Busca da Formação Integral do Universitário**. Dissertação de Mestrado. Faculdade de Educação, Pontifícia Universidade Católica do Rio Grande do Sul, 2005. 132 p.

MITCHELL, T. **Machine Learning**. [S.l.]: McGraw Hill, 1997. 414 p.

MOENS, M. F. **Automatic Indexing and Abstracting of Document Texts**. Boston: Kluwer Academic Publishers, 2000. 265 p.

NOU-RAU. Disponível em: <http://www.rau-tu.unicamp.br/nou-rau/>. Acesso em: 08 out. 2004.

NRC. Disponível em: <http://www.extractor.com/Extractor7Details.htm>. Acesso em 02 out. 2004.

NZDL. Biblioteca Digital da Nova Zelândia. Disponível em: <http://www.sadl.uleth.ca/nz/cgi-bin/library>. Acesso em: 28 set. 2004.

NZDL/KEA. Disponível em: <http://www.nzdl.org/Kea/>. Acesso em 09 out. 2004.

ORENGO, V. M.; HUYCK, C. R. A Stemming Algorithim for The Portuguese Language. In: **Proceedings of the SPIRE Conference**. Laguna de San Raphael: [s.n.], 2001, p. 13-15.

PORTUGUESE STEMMER. Disponível em: <http://gold.mdx.ac.uk/~viviane1/rslp.html>. Acesso em: 15 ago. 2004.

OXFORD. **Oxford Dictionary and Thesaurus, The**. American Edition. Oxford: Oxford University Press, 1996.

PEREIRA, M. B.; NUNES, M. G. V. Algoritmos de Extração de Palavras-Chave de Textos em Português. **Série de Relatórios do NILC - ICM-USP**. NILC-TR-01-6, 2001. 16 p.

PORTER, M. F. **An Algorithm for Suffix Stripping**. Program. [S.l.]: [s.n.], v. 14, n. 3, 1980. p. 130-137.

RAU-TU. Disponível em: <http://www.rau-tu.unicamp.br/>. Acesso em: 12 abr. 2004.

RSLP. Disponível em: <http://www.gold.mdx.ac.uk/~viviane1/rslp.html>. Acesso em 20 abr. 2004.

SAINT-DIZIER, P. On the Polymorphic Behavior of Word-senses. In: **Lingüística Computacional: Investigação Fundamental e Aplicações**. Lisboa: Colibri, 1999. p. 209-56.

SCAPINI, I. K. et al. Relações entre Itens Lexicais. Fundamentos de um Dicionário Remissivo. In: Primeiro Encontro do CELSUL. **Anais**, Florianópolis: [s.n.], 1995. p. 327-334.

SCHOLTES, J. C. Neural Networks in Natural Language Processing and Information Retrieval. PhD thesis, Institute for Logic, Language and Computation (ILLC).University of Amsterdam, 1993.

SNOWBALL. Disponível em: <http://snowball.tartarus.org>. Acesso em 20 nov. 2003.

SPARCK-JONES, K.; WILLET, P. Readings in Information Retrieval. São Francisco: Morgan Kaufmann, 1997.

TURNEY, P. D. Extraction of keyphrases from text: evaluation of four algorithms. Technical Report. National Research Council of Canada, 1997.

TURNEY, P. D. Learning to extract keyphrases from text. **Technical Report**. National Research Council of Canada, 1999.

VIEIRA, R. Textual co-reference annotation: a study on definite descriptions. **Anais do VIII Congresso da Sociedade Argentina de Linguística**. Mar del Plata, Argentina: [s.n.], 2000.

ZUCHINI, M. H. **Aplicações de Mapas Auto-Organizáveis em Mineração de Dados e Recuperação de Informação**. Tese de Mestrado. Universidade Estadual de Campinas (UNICAMP). Faculdade de Engenharia Elétrica e de Computacção (FEEC), 2003.

WHITLEY, D. The Genitor algorithm and selective pressure: Why Rank-Based Allocation of

Reproductive Trials is Best. In: **Proceedings 3th International Conference on Genetic Algorithms**. [S.l.]: Morgan Kaufmann, 1989. p. 116-121. Disponível em: <http://www.cs.colostate.edu/~genitor/Pubs.html>. Acesso em: 02 out. 2004.

WITTEN I. H. et al. KEA: Practical automatic keyphrase extraction. In: **Proceedings of the Fourth ACM Conference on Digital Libraries**. [S.l.]: [s.n.], 1999. p. 254-255.

WITTEN, I. H.; FRANK, E. **Data Mining:** Practical Machine Learning Tools and Techniques with Java. São Francisco: Academic Press, 2000.

ÍNDICE DE CITAÇÃO DE AUTORES

GLOSSÁRIO

- **Advérbios** – são palavras que são fundamentalmente um modificador do verbo.
- **Afixo** – é o elemento mórfico que se junta a uma raiz ou radical a fim de modificar geralmente de forma precisa o sentido de uma palavra.
- **Aprendizado de Máquina** – é uma subárea de pesquisa em Inteligência Artificial que estuda métodos computacionais para adquirir novos conhecimentos, novas habilidades e novos meios de organizar o conhecimento já existente.
- **Artigos** – são as palavras o, a, os, as e um, uma, uns, umas que se antepõem aos substantivos para indicar que se trata de um termo definido ou indefinido.
- **Classificação** – é a etapa onde o modelo construído na etapa de treinamento, é utilizado para avaliar automaticamente novos conjuntos de dados, classificando-os sem intervenção humana.
- **Conflação** – é o ato de fusão ou combinação para igualar variantes morfológicas de palavras.
- **Conjunções** – são os vocábulos gramaticais que servem para relacionar duas orações ou dois termos semelhantes da mesma oração.
- **Descoberta de Conhecimento em Textos (*Text Mining*)** – é o processo de extrair padrões ou conhecimento, interessantes e não-triviais, a partir de documentos textuais.
- **Desinência** – é o elemento mórfico que indica as flexões das palavras, e divide-se em nominais e verbais.

- **Determinação de palavras-chave** – procura selecionar as frases de um vocabulário controlado que melhor descreva um documento.
- **Extração de Informação** – difere de Recuperação de Informação. O foco é encontrar informação útil dentro de documentos e não em encontrar documentos.
- **Extração de palavras-chave** – esta abordagem não usa um vocabulário controlado e sim escolhe palavras-chave do próprio texto.
- **Fonemas** – são unidades ainda menores que as palavras e que apresentam apenas a parte significante.
- **Lema** ou **forma canônica** – é a palavra reduzida à forma canônica.
- **Léxico** – significa uma relação de palavras com suas categorias gramaticais e seus significados.
- **Língua** – é constituída de um conjunto infinito de frases.
- **Morfemas** – são unidades de som e conteúdo menores do que as palavras.
- ***Overstemming*** – ocorre quando a parte removida da palavra não é um sufixo, mas parte do seu radical.
- **Palavras** – são unidades menores de som e significado que formam frases. As palavras são unidades menores que a frase e maiores que o fonema.
- **Precisão** (*precision*) – fornece a relação entre o número de documentos recuperados relevantes e o total de documentos obtidos.
- **Prefixo** – é o elemento mórfico que antepõem ao radical.
- **Preposições** – são palavras invariáveis que relacionam dois termos de uma oração, de tal modo que o sentido do primeiro é explicado ou completado pelo segundo.
- **Processamento de Linguagem Natural** – trata computacionalmente os diversos aspectos da comunicação humana, como sons, palavras, sentenças e discursos, considerando formatos e referências, estruturas e significados, contextos e usos. É normalmente usado para descrever a

função de softwares ou componentes de hardware em um sistema computacional que analisam ou sintetizam linguagem falada ou escrita.

- **Pronomes** – são palavras que desempenham na frase as funções equivalentes às exercidas pelos substantivos e nomes próprios. Servem para representar um substantivo ou ara acompanhar um substantivo determinando-lhe a extensão do significado.
- **Radical** – é o elemento mórfico que fornece a significação da palavra.
- **Radicalização** – é o processo de combinar as formas diferentes de uma palavra numa representação comum, o radical.
- **Raiz** – é o elemento mórfico mais simples a que pode ser reduzida uma palavra.
- **Recuperação** (*recall*) – fornece a relação entre o número de documentos recuperados relevantes e o total de documentos relevantes que existem.
- **Recuperação de Informação** – é a aplicação de tecnologia computacional para a aquisição, organização, armazenamento, recuperação e distribuição de informação.
- **Redução à forma canônica** – é o ato de representar as palavras através do infinitivo dos verbos e masculino singular dos substantivos e adjetivos.
- ***Stopwords*** – são palavras freqüentes em um texto e que não representam nenhuma informação de maior relevância para o texto.
- **Sufixo** – é o elemento mórfico que pospõem ao radical.
- **Tema** – é o radical acrescido de uma vogal temática, isto é, pronto para receber uma desinência (ou um sufixo).
- **Treinamento** – é a etapa onde o sistema é alimentado com um conjunto de testes previamente classificados e organizados.
- ***Understemming*** – ocorre quando um sufixo não é removido completamente.
- **Usuários** – são pessoas que usam os recursos da Informática em geral.
- **Vogais e consoantes de ligação –** são sons empregados para tornar a pronúncia das palavras

mais fácil ou eufônica.

- **Vogal temática** – é o elemento mórfico que se agrega ao radical de uma palavra para que ela possa receber outros morfemas.

APÊNDICE A – Lista de stopwords da língua portuguesa

• Artigos

Artigos definidos				Artigos indefinidos			
o	a	os	as	um	uma	uns	umas
Formas combinadas com preposições A, DE, EM e POR (PER)				**Formas combinadas com preposições EM e DE**			
ao	à	aos	às	num	numa	nuns	numas
do	da	dos	das	dum	duma	duns	dumas
no	na	nos	nas	-			
pelo	pela	pelos	pelas				

• Pronomes

Pronomes pessoais retos							
eu	tu	ele	ela	nós	vós	eles	elas
Contração das preposições DE e EM com ELE(S) e ELA(S)							
dele	dela	deles	delas	nele	nela	neles	nelas

Pronomes pessoais oblíquos não reflexivos			
Átonos		**Tônicos**	
me	–	mim, comigo	nós, conosco
te	vos	ti, contigo	vós, convosco
lhe	lhes	ele, ela	eles, elas
Formas do pronome oblíquo			
lo	la	los	las

Pronomes pessoais reflexivos e recíprocos			
se	si	consigo	–

Pronomes possessivos							
Um possuidor				**Vários possuidores**			
meu	minha	meus	minhas	nosso	nossa	nossos	nossas

Pronomes possessivos							
teu	tua	teus	tuas	vosso	vossa	vossos	vossas
seu	sua	seus	suas	seu	sua	seus	suas

Pronomes demonstrativos				
Variáveis				**Invariáveis**
este	esta	estes	estas	isto
esse	essa	esses	essas	isso
aquele	aquela	aqueles	aquelas	aquilo
Formas combinadas com preposições DE, EM e A				
deste	desta	destes	destas	disto
neste	nesta	nestes	nestas	nisto
desse	dessa	desses	dessas	disso
nesse	nessa	nesses	nessas	nisso
daquele	daquela	daqueles	daquelas	daquilo
naquele	naquela	naqueles	naquelas	naquilo
àquele	àquela	àqueles	àquelas	àquilo

Pronomes relativos				
Variáveis				**Invariáveis**
qual	qual	quais	quais	que
cujo	cuja	cujos	cujas	quem

Pronomes interrogativos			
Variáveis		**Invariáveis**	
qual	quais	que	quem

Pronomes indefinidos				
Variáveis				**Invariáveis**
Masculino		**Feminino**		
algum	alguns	alguma	algumas	alguém
nenhum	nenhuns	nenhuma	nenhumas	ninguém
outro	outros	outra	outras	outrem

Pronomes indefinidos				
muito	muitos	muita	muitas	nada
pouco	poucos	pouca	poucas	cada
–	tantos	tanta	tantas	tudo
–				algo

· Advérbios

Advérbios de afirmação			
sim	decerto	deveras	–

Advérbios de dúvida			
acaso	porventura	quiçá	talvez

Advérbios de intensidade			
assaz	demais	mais	quase
tanto	tão	–	

Advérbios de lugar			
abaixo	acima	adiante	aí
além	ali	aquém	aqui
atrás	através	cá	defronte
dentro	detrás	fora	lá
longe	perto	onde	aonde
donde	adentro	afora	algures
alhures	nenhures	embaixo	debaixo
diante	–		

Advérbios de modo			
assim	adrede	debalde	depressa
devagar	mal	–	

Advérbios de negação	
não	tampouco

Advérbios de tempo			
agora	ainda	amanhã	anteontem
antes	cedo	depois	então
hoje	já	jamais	logo
nunca	ontem	outrora	sempre
tarde	amiúde	entrementes	enfim
Outros advérbios			
acerca	acinte	afinal	aliás
apenas	apesar	inclusive	eis
somente	também	entanto	entretanto
ora	contanto	quando	–

• Preposições

Preposições			
Simples			**Acidentais**
a	com	em	durante
por	ante	contra	exceto
entre	per	após	–
de	para	sem	
até	desde	perante	
sob	sobre	trás	

• Conjunções

Conjunções			
Coordenativas		**Subordinativas**	**Proporcionais**
contudo	e	como	enquanto

Conjunções			
mas	nem	conquanto	–
ou	pois	embora	
porquanto	porque	–	
portanto	porém		
todavia	–		

- **Consoantes e vogais**

Consoantes e vogais						
a	b	c	d	e	f	g
h	i	j	k	l	m	n
o	p	q	r	s	t	u
v	w	x	y	z	-	

APÊNDICE B – Lista de regras do radicalizador Portuguese Stemmer

PLURAL

Regra P	ns	1	m	íons, elétrons, prótons, nêutrons, fótons, epsilons
Regra P	ães	1	ão	mães
Regra P	ões	2	ão	
Regra P	ais	1	al	cais, mais, pais, demais, ademais, jamais, anais
Regra P	éis	1	el	réis
Regra P	eis	3	el	fósseis, táteis, répteis, fáceis, frágeis, têxteis, férteis, inférteis, voláteis, inúteis, dóceis, indóceis, estéreis, hábeis, inábeis, portáteis, débeis, mísseis, fúteis, vibráteis, verocímeis, projéteis
Regra P	eis	2	il	
Regra P	óis	3	ol	heróis
Regra P	uis	2	ul	caquis, sanguis, croquis, sambaquis
Regra S	álcoois			álcool
Regra P	is	2	il	lápis, cais, mais, crúcis, biquínis, pois, depois, dois, oásis, pais, demais, ademais, jamais, anais, reis, leis, práxis, quis, tênis, sífilis, pênis, bois, grátis, oásis, brócolis, pélvis, júris, álcalis, zumbis, púbis, clitóris, bílis, bisturis, íris, táxis, alíbis, guris, chassis, abacaxis, caquis, sanguis, croquis, sambaquis
Regra P	les	2	l	simples, deles, aqueles, daqueles, controles, àqueles, neles, naqueles, vales, peles, isósceles, móbiles, hipérboles, sístoles, metrópoles
Regra P	nes	4	n	perenes, autóctones, microfones, cabines, aborígenes, telefones, gramofones
Regra P	eses	4	ês	*teses, *gêneses, dioceses
Regra P	ses	1	s	*pses, *sses, *fases, *frases, *tases, *oses, *teses, *enses, *gêneses, dioceses, análises, bases, crises
Regra P	res	1	r	*bres, *cres, *dres, *fres, *gres, *tres, *vres, árvores, softwares, hardwares, pires, escores, torres, hectares, alferes, alhures, víveres, títeres, alqueires, porres
Regra P	zes	2	z	fezes, deslizes, varizes, bronzes
Regra P	s	1	""	aliás, pires, lápis, cais, mais, mas, menos, férias, fezes, pêsames, crúcis, gás, atrás, trás, detrás, moisés, através, convés, invés, *ês, país, após, ambas, ambos, messias, oásis, ônibus, dois, duas, três, depois, revés, seis, dezesseis, atlas, alvíssaras, anais, antolhos, calendas, cãs, condolências, exéquias, fastos, núpcias, mês, olheiras, primícias, víveres, viés, demais, ademais, jamais, réis, grátis, brócolis, pélvis, púbis, clitóris, ânus, bílis, íris, isósceles, simples, parênteses, apenas, vós, nós, antes, pós, deus, cós, status, caos, ônus, vírus, tônus, bônus, versus, campus, stress, corpus, través

ADVERBIOS

Regra P	mente	4	""	movimente, *argumente, fragmente, implemente, incremente, decremente, experimente, complemente, *regulamente, instrumente, cumprimente, arregimente, fundamente, suplemente, sedimente, parlamente, documente, *compartimente, atormente, *alimente, ornamente, regimente, pavimente, sacramente

FEMININO

Regra P	dora	3	dor	
Regra P	sora	3	sor	
Regra P	tora	2	tor	fatora
Regra S	senhora			senhor
Regra P	ona	3	ão	*crona, *erona, *iona, *lona, *nona, desabona, abandona, telefona, sanfona, antígona, mamona, japona, corona, poltrona, matrona, manjerona, dipirona, destrona, persona, unísona, *cortisona, *metasona, monótona, maratona, *cetona, detona, *oxítona, apaixona, azeitona, *zona
Regra S	baronesa			barão
Regra S	duquesa			duque
Regra S	princesa			príncipe
Regra P	esa	4	ês	*presa, *defesa, despesa, sobremesa, turquesa
Regra P	ã	2	ão	manhã, amanhã, maçã, xamã, afã, clã, *ímã, divã, sutiã, titã, tucumã, marzipã
Regra P	ésima	2	ésimo	
Regra P	íssima	3	íssimo	
Regra P	érrima	3	érrimo	
Regra P	iva	4	ivo	gengiva
Regra P	eira	3	eiro	madeira, cadeira, ribeira, bandeira, esteira, peneira, ladeira, derradeira, requeira, caveira, lareira
Regra P	izada	4	izado	
Regra P	ada	1	ado	*camada, pitada, entrada, década, cada, nada, jornada, palmada, batelada, enxada, congada, espada, risada, saraivada, tonelada, marmelada, goiabada, lombada, camarada, trovoada, chuvarada, toada, ossada, macacada, granada, gônada, alvorada, chibatada, salada, peixada, lâmpada, meninada, molecada, mulherada, criançada, olimpíada, colherada, facada, panelada, cilindrada, escada, cabeçada, cachorrada, joelhada, bofetada, barrigada, narigada, manada, lambada, jangada, porrada, dentada, cilada, arcada, moçada, polegada, garotada, papelada, pomada, piazada, balada, almofada, rapaziada, feijoada, ninhada, bolada, boiada
Regra P	enta	3	ento	quarenta, cinqüenta, sessenta, setenta, oitenta, noventa, pimenta, placenta, tormenta, parenta, polenta, magenta
Regra N	oa		ão	leitoa, patroa, leoa
Regra N	tisa		ta	poetisa, profetisa
Regra S	sacerdotisa			sacerdote

Regra N	triz		tor	atriz, imperatriz
Regra S	sílfide			silfo
Regra S	diaconisa			diácono
Regra S	égua			cavalo
Regra S	galinha			galo
Regra S	maestrina			maestro
Regra S	monja			monge
Regra S	rainha			rei

AUMENTATIVO

Regra P	zona	3	""	macrozona, microzona, subzona, biozona, *butazona
Regra P	bilíssimo	3	vel	
Regra S	antiquíssimo			antigo
Regra P	quíssimo	2	co	
Regra P	díssimo	3	do	grandíssimo
Regra S	amicíssimo			amigo
Regra S	dulcíssimo			doce
Regra P	císsimo	4	z	
Regra S	grandessíssimo			grande
Regra S	longuíssimo			longo
Regra P	íssimo	3	""	
Regra S	magérrimo			magro
Regra S	paupérrimo			pobre
Regra P	érrimo	3	""	
Regra P	alhão	3	""	batalhão, trabalhão, trapalhão, medalhão
Regra P	aça	3	""	ameaça, abraça, *faça, cabaça, *laça, espaça, embaça, desgraça, carcaça, cachaça, carapaça, rechaça, trapaça, *embaraça, despedaça, estilhaça, arruaça, *mordaça, linhaça, esvoaça, congraça, arregaça
Regra P	aço	3	""	*espaço, pedaço, abraço, almaço, *braço, palhaço, *embaraço, cangaço, rechaço, retraço, *faço, mormaço
Regra P	uça	4	""	
Regra P	ázio	3	""	topázio
Regra P	arraz	4	""	
Regra P	arra	3	""	esbarra, cigarra, bizarra, fanfarra, guitarra
Regra P	orra	3	""	
Regra P	anzil	4	""	
Regra P	aréu	3	""	
Regra P	astro	4	""	
Regra P	asta	4	""	contrasta, desgasta, desbasta
Regra P	asto	4	""	*plasto, *blasto
Regra P	zarrão	3	""	
Regra P	rrão	4	""	empurrão, macarrão, chimarrão
Regra P	zão	2	""	*razão, vazão, prizão, coalizão
Regra S	casarão			casa
Regra S	asneirão			asno
Regra S	toleirão			tolo
Regra S	vozeirão			voz
Regra S	narigão			nariz

Regra P	ão	3	""	camarão, chimarrão, canção, coração, embrião, grotão, glutão, ficção, fogão, feição, furacão, gamão, lampião, *leão, macacão, nação, órfão, orgão, patrão, portão, quinhão, rincão, tração, falcão, espião, mamão, folião, cordão, aptidão, campeão, colchão, limão, leilão, melão, barão, milhão, bilhão, fusão, cristão, ilusão, estação, senão
Regra S	fornalha			forno
Regra S	gentalha			gente
Regra S	muralha			muro
Regra S	queixada			queixo

DIMINUTIVO

Regra P	quinha	2	ca	mesquinha
Regra P	quinho	2	co	mesquinho, parquinho, molequinho, bosquinho, chequinho
Regra P	guinha	2	ga	linguinha
Regra P	guinho	2	go	sanguinho
Regra P	zinha	2	""	*vizinha, cozinha, vozinha, luzinha, belezinha, brazinha
Regra P	zinho	2	""	*vizinho, cozinho, rapazinho, quinzinho, comezinho, gizinho
Regra P	cinha	2	ça	docinha
Regra P	cinho	2	ço	focinho, toicinho, toucinho, docinho, ancinho
Regra S	asinha			asa
Regra S	feinha			feia
Regra S	joinha			jóia
Regra S	meinha			meia
Regra S	sainha			saia
Regra S	veinha			veia
Regra P	inha	3	a	*caminha, mantinha, continha, farinha, marinha, espinha, detinha, *linha, sobrinha, obtinha, convinha, advinha, campainha, ladainha, engatinha, intervinha, andorinha, mesquinha, *vizinha, cozinha
Regra S	aninho			ano
Regra S	radinho			radio
Regra P	inho	3	o	caminho, carinho, sobrinho, marinho, cadinho, espinho, redemoinho, advinho, engatinho, pergaminho, encaminho, torvelinho, cominho, golfinho, mesquinho, *vizinho, cozinho, comezinho, focinho, toicinho, toucinho, ancinho
Regra S	pequenina			pequena
Regra S	pequenino			pequeno
Regra S	boletim			boleto
Regra S	botequim			bar
Regra S	camarim			camara
Regra S	espadim			espada
Regra S	festim			festa
Regra S	folhetim			folha

Regra S	fortim			forte
Regra S	fedelha			feder
Regra S	fedelho			feder
Regra S	grupelho			grupo
Regra S	rapazelho			rapaz
Regra S	animalejo			animal
Regra S	festejo			festa
Regra S	gracejo			graça
Regra S	lugarejo			lugar
Regra S	quitalejo			quintal
Regra S	sertaneja			sertão
Regra S	sertanejo			sertão
Regra S	vilarejo			vila
Regra S	vasilha			vaso
Regra P	ilha	4	""	compartilha, maravilha, desempilha, desvencilha, fervilha, engatilha, lentilha
Regra P	ilho	4	""	compartilho, maravilho, desempilho, desvencilho, fervilho, engatilho
Regra S	fogacho			fogo
Regra S	penacho			pena
Regra S	populaçho			povo
Regra S	riacho			rio
Regra S	barbicha			barba
Regra S	governicho			governo
Regra S	capucha			capa
Regra S	casucha			casa
Regra S	capucho			capa
Regra S	cartucho			carta
Regra S	gorducho			gordo
Regra S	papelucho			papel
Regra S	pequerrucho			pequeno
Regra S	casebre			casa
Regra S	boteco			bar
Regra S	filmeco			filme
Regra S	soneca			sono
Regra S	folheca			folha
Regra S	jornaleco			jornal
Regra S	livreco			livro
Regra S	burrico			burro
Regra S	amorico			amor
Regra S	namorico			namoro
Regra S	ruela			rua
Regra S	viela			via
Regra S	cidadela			cidade
Regra S	mordidela			morder
Regra S	olhadela			olho
Regra S	piscadela			piscar
Regra S	sacudidela			sacudir

Regra P	onete	3	""	
Regra P	uete	3	""	
Regra N	ete		""	balancete, barrilete, bracelete, cacetete, canivete, capacete, cartazete, cavalete, claquete, clarinete, corpete, estilete, florete, lembrete, martelete, palacete, patinete, ramalhete, rolete, tablete, trompete, verbete, artiguete, malandrete
Regra S	disquete			disco
Regra S	gabinete			cabine
Regra S	charrete			carro
Regra P	oneta	3	""	
Regra N	queta		c	barqueta, fabriqueta, plaqueta
Regra N	eta		""	bicicleta, camiseta, faceta, motocicleta, caderneta, caixeta, saleta, carreta, chupeta, vareta, prancheta, mureta, maleta, corneta, clarineta, chaveta, barqueta, saleta, papeleta, maneta, fabriqueta, canaleta, trombeta, estatueta, marreta, historieta, filipeta, costeleta
Regra S	tabuleta			tábua
Regra N	eto		""	livreto, carreto, poemeto, verseto, esboceto
Regra S	libreto			livro
Regra S	florzita			flor
Regra S	jardinzito			jardim
Regra S	pequetita			pequena
Regra N	ita		""	blusita, camisita, carmelita, fulanita, israelita, salita, saudita, senhorita
Regra S	mosquito			mosca
Regra S	palito			pau
Regra S	pequetito			pequeno
Regra N	ito		""	cabrito, modelito, negrito, erudito, rapazito
Regra S	velhota			velha
Regra N	ote		""	filhote, meninote, grandote, pequenote, molecote, fracote, serrote, velhote, sacerdote, cabeçote, caixote, malote, camarote
Regra N	isca		""	mourisca, talisca
Regra N	isco		""	mourisco, chuvisco, levantisco, pedrisco
Regra N	usca		""	velhusca
Regra N	usco		""	velhusco, chamusco
Regra N	ola		""	fazendola, rapazola, marola, bandeirola, camisola, gabarola, mariola, casinhola, ventarola, pianola, cachola
Regra N	úncula		""	questiúncula
Regra N	únculo		""	homúnculo, pedúnculo
Regra N	úscula		""	maiúscula, minúscula
Regra N	úsculo		""	corpúsculo, opúsculo, maiúsculo, minúsculo
Regra N	áculo		""	vernáculo, sustentáculo, receptáculo, habitáculo, tabernáculo, tentáculo
Regra N	érculo		""	tubérculo
Regra N	ícula		""	febrícula, gotícula, partícula, película, radícula, matrícula, quadrícula, vesícula

Regra N	ículo		""	montículo, vermículo, versículo, ventrículo, folículo, fascículo, testículo, cubículo, funículo, pedículo, ossículo, montículo, canalículo
Regra N	ula		""	nótula, rótula, molécula
Regra N	ulo		""	glóbulo, grânulo, módulo, nódulo, régulo

NUMERAL

Regra P	ésimo	2	""	

FORMAS NOMINAIS

Regra P	bilizado	2	v	
Regra P	alizado	4	""	
Regra P	atizado	4	""	
Regra P	tizado	4	""	
Regra P	izado	4	""	
Regra P	ado	2	""	prado, grado, veado, brado, alado, estado, senado, sábado, fígado, bêbado, bocado, rosado, enfado, côvado, mercado, machado, quadrado, deputado, telhado, eldorado, feriado
Regra P	guento	4	g	
Regra P	quento	4	c	
Regra N	ulento		""	corpulento, turbulento, fraudulento, truculento, purulento, virulento
Regra S	sangrento			sangue
Regra S	sonolento			sono
Regra P	ento	3	""	*mento, *vento, *tento, *sento, enfrento, talento, acrescento, sargento, desalento, relento, rebento, arrebento
Regra P	ativo	4	""	
Regra P	tivo	4	""	
Regra S	defensivo			defender
Regra P	ivo	4	""	arquivo, passivo, massivo, ostensivo, defensivo, remissivo, convivo, cursivo, compassivo, lascivo, elusivo
Regra P	encialista	5	""	
Regra P	alista	5	""	
Regra P	icionista	4	""	
Regra P	cionista	4	""	
Regra P	ionista	4	""	
Regra N	tista		c	cientista, renascentista, separatista, corporatista, preventista
Regra S	estatista			estado
Regra P	ista	3	""	
Regra P	izagem	6	""	
Regra P	agem	3	""	mensagem, *vantagem, paisagem, interagem, homenagem, bagagem, *imagem, massagem, garagem, chantagem, estalagem, fuselagem, carenagem
Regra N	agem		ag	mensagem, *vantagem, paisagem, interagem, homenagem, bagagem, *imagem, massagem, garagem, chantagem, estalagem, fuselagem, carenagem
Regra P	amento	3	""	filamento, firmamento, departamento
Regra P	imento	3	""	detrimento, pavimento, condimento

Regra P	ido	3	""	líquido, marido, fluido, *válido, sólido, híbrido, rígido, *óxido, libido, nítido, tímido, bandido, comprido, valido, *ácido, pálido, liquido, *lúcido, límpido, prurido, mórbido, estúpido, convido, decido, plácido, esplêndido, sórdido, grávido, insípido, explêndido, cálido, vívido, sustenido, pútrido, lívido, lânguido, fúlgido, flácido, anidrido, tórrido, *árido, lépido, intrépido, impávido, esquálido, tépido, pérfido, frígido, bólido, cândido, duvido
Regra P	ído	3	""	*aldeído
Regra P	ador	3	""	
Regra P	edor	3	""	
Regra P	idor	3	""	
Regra P	dor	2	d	condor
Regra P	ssor	3	ss	
Regra P	sor	4	s	
Regra P	tor	3	""	leitor, doutor, escritor, monitor, reitor, pastor, gestor, agricultor, benfeitor, consultor
Regra P	or	2	""	maior, menor, melhor, redor, rigor, tambor, tumor, pastor, interior, favor, autor
Regra N	esco		""	burlesco, principesco, parentesco, gigantesco, romanesco
Regra P	esco	4	""	
Regra P	atória	5	""	
Regra P	oria	4	""	categoria
Regra P	ário	3	""	voluntário, salário, aniversário, *lionário, armário
Regra P	atório	3	""	
Regra P	rio	5	""	voluntário, aniversário, compulsório, *lionário, *stério
Regra P	ério	6	""	
Regra P	abilidade	5	""	
Regra P	ividade	5	""	
Regra P	idade	4	""	autoridade, comunidade
Regra P	ionar	5	""	
Regra P	ional	4	""	
Regra P	ência	3	""	
Regra P	ância	4	""	ambulância
Regra P	edouro	3	""	
Regra P	queiro	3	c	
Regra P	adeiro	4	""	desfiladeiro
Regra P	eiro	3	""	desfiladeiro, pioneiro, mosteiro
Regra P	uoso	3	""	
Regra P	oso	3	""	precioso
Regra P	alizaç	5	""	
Regra P	atizaç	5	""	
Regra P	tizaç	5	""	
Regra P	izaç	5	""	organizaç
Regra P	aç	3	""	equaç, relaç
Regra P	iç	3	""	eleiç
Regra P	ês	4	""	
Regra P	eza	3	""	
Regra P	ez	4	""	

Regra P	ante	2	""	gigante, elefante, adiante, possante, instante, restaurante
Regra P	ástico	4	""	eclesiástico
Regra P	alístico	3	""	
Regra P	áutico	4	""	
Regra P	êutico	4	""	
Regra P	ático	4	""	alopático
Regra P	tico	3	""	político, eclesiástico, diagnóstico, prático, doméstico, diagnóstico, idêntico, alopático, artístico, autêntico, eclético, crítico, critico
Regra P	ico	4	""	*tico, público, explico
Regra P	encial	5	""	
Regra P	auta	5	""	
Regra P	quice	4	c	
Regra P	ice	4	""	cúmplice
Regra P	íaco	3	""	
Regra P	ente	4	""	freqüente, alimente, acrescente, permanente, aparente
Regra P	ense	5	""	
Regra P	inal	3	""	
Regra P	ano	4	""	
Regra P	ável	2	""	afável, razoável, potável, vulnerável
Regra P	ível	3	""	possível
Regra P	vel	5	""	possível, vulnerável
Regra P	bil	3	vel	
Regra P	ura	4	""	imatura, acupuntura, costura
Regra P	ural	4	""	
Regra P	ual	3	""	bissexual, virtual, visual, pontual
Regra P	ial	3	""	
Regra P	al	4	""	afinal, animal, estatal, bissexual, desleal, fiscal, formal, pessoal, liberal, postal, virtual, visual, pontual, sideral, sucursal
Regra P	alismo	4	""	
Regra P	ivismo	4	""	
Regra P	ismo	3	""	cinismo
Regra P	quid	3	co	
Regra P	id	3	""	solid, multid, partid, marid, partid

TERMINAÇOES VERBAIS

Regra P	aríamo	2	""	
Regra P	ássemo	2	""	
Regra P	eríamo	2	""	
Regra P	êssemo	2	""	
Regra P	iríamo	3	""	
Regra P	íssemo	3	""	
Regra P	áramo	2	""	
Regra P	árei	2	""	
Regra P	aremo	2	""	
Regra P	ariam	2	""	
Regra P	aríei	2	""	
Regra P	ássei	2	""	
Regra P	assem	2	""	
Regra P	ávamo	2	""	

Regra P	êramo	3	""	
Regra P	eremo	3	""	
Regra P	eriam	3	""	
Regra P	eríei	3	""	
Regra P	êssei	3	""	
Regra P	essem	3	""	
Regra P	íramo	3	""	
Regra P	iremo	3	""	
Regra P	iriam	3	""	
Regra P	iríei	3	""	
Regra P	íssei	3	""	
Regra P	issem	3	""	
Regra P	ando	2	""	
Regra P	endo	3	""	
Regra P	indo	3	""	
Regra P	ondo	3	""	
Regra P	aram	2	""	
Regra P	arão	2	""	
Regra P	arde	2	""	
Regra P	arei	2	""	
Regra P	arem	2	""	
Regra P	aria	2	""	
Regra P	armo	2	""	
Regra P	asse	2	""	
Regra P	aste	2	""	
Regra P	avam	2	""	agravam
Regra P	ável	2	""	
Regra P	eram	3	""	
Regra P	erão	3	""	
Regra P	erde	3	""	
Regra P	erei	3	""	
Regra P	êrei	3	""	
Regra P	erem	3	""	
Regra P	eria	3	""	
Regra P	ermo	3	""	
Regra P	esse	3	""	
Regra P	este	3	""	faroeste, agreste
Regra P	íamo	3	""	
Regra P	iram	3	""	
Regra P	íram	3	""	
Regra P	irão	2	""	
Regra P	irde	2	""	
Regra P	irei	3	""	admirei
Regra P	irem	3	""	adquirem
Regra P	iria	3	""	
Regra P	irmo	3	""	
Regra P	isse	3	""	
Regra P	iste	4	""	
Regra P	iava	4	""	ampliava
Regra P	amo	2	""	
Regra P	iona	3	""	
Regra P	ara	2	""	arara, prepara
Regra P	ará	2	""	alvará
Regra P	are	2	""	prepare

Regra P	ava	2	""	agrava
Regra P	emo	2	""	
Regra P	era	3	""	acelera, espera
Regra P	erá	3	""	
Regra P	ere	3	""	espere
Regra P	iam	3	""	enfiam, ampliam, elogiam, ensaiam
Regra P	íei	3	""	
Regra P	imo	3	""	reprimo, intimo, íntimo, *nimo, queimo, *ximo
Regra P	ira	3	""	fronteira, sátira
Regra P	ído	3	""	
Regra P	irá	3	""	
Regra P	tizar	4	""	alfabetizar
Regra P	izar	5	""	organizar
Regra P	itar	5	""	acreditar, explicitar, estreitar
Regra P	ire	3	""	adquire
Regra P	omo	3	""	
Regra P	ai	2	""	
Regra P	am	2	""	
Regra P	ear	4	""	alardear, nuclear
Regra P	ar	2	""	azar, bazar, patamar
Regra P	uei	3	""	
Regra P	uía	5	u	
Regra P	ei	3	""	
Regra P	guem	3	g	
Regra P	em	2	""	*alem, virgem
Regra P	er	2	""	éter, pier
Regra P	eu	3	""	chapeu
Regra P	ia	3	""	estória, fatia, *acia, praia, elogia, mania, lábia, aprecia, polícia, arredia, cheia, *ásia
Regra P	ir	3	""	freir
Regra P	iu	3	""	
Regra P	eou	5	""	
Regra P	ou	3	""	
Regra P	i	3	""	

VOGAL TEMATICA

Regra P	bil	2	vel	
Regra P	gue	2	g	gangue, jegue
Regra P	á	3	""	
Regra P	ê	3	""	bebê
Regra P	a	3	""	ásia
Regra P	e	3	""	
Regra P	o	3	""	*ão