ABORDAGEM LINGUÍSTICA NA CLASSIFICAÇÃO AUTOMÁTICA DE TEXTOS EM PORTUGUÊS

Yuri Barwick Lannes de Camargo

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

_____

Prof. Jorge Lopes de Souza Leão, Dr.Ing.

_____

Prof. Antônio Carneiro de Mesquita Filho, Dr.d'État

_____

Prof.Geraldo Bonorino Xexéo, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

JUNHO DE 2007

CAMARGO, YURI BARWICK LANNES DE

Abordagem lingüística na classificação de textos em português [Rio de Janeiro] 2007

X, 89 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Elétrica, 2007)

Dissertação – Universidade Federal do Rio de Janeiro, COPPE

1. Classificação automática de textos

I. COPPE II.Título (série)

AGRADECIMENTOS

Ao Professor Jorge Leão, meu orientador, pelas sugestões sempre oportunas e pela paciência na condução deste trabalho.

Aos amigos Henrique, Fabiana, Marcel e Laila que com suas palavras e apoio possibilitaram a chegada até este trabalho.

A Danielle, minha adorada esposa, pela compreensão e amor nos momentos necessários de recolhimento e ausência.

Ao meu filho Gustavo, pela alegria dos seus olhos nos momentos em que o trabalho parecia interminável.

A minha mãe Vera, pelo seu exemplo de perseverança e espírito de luta.

A Marinha do Brasil por ter autorizado a realização deste Curso de Mestrado.

A todos que perguntaram: “E aí, com vai a tese?”

E a Deus pela saúde e fé na vida. Muito Obrigado.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

ABORDAGEM LINGUÍSTICA NA CLASSIFICAÇÃO AUTOMÁTICA DE TEXTOS EM PORTUGUÊS

Yuri Barwick Lannes de Camargo

Junho/2007

Orientador: Jorge Lopes de Souza Leão

Programa: Engenharia Elétrica

Este trabalho aborda uma classificação automática de textos em português com o uso de informações lingüísticas na etapa de pré-processamento dos textos. Para isso são utilizadas duas coleções de textos, sendo uma composta de artigos de jornal (Folha de São Paulo, 1994) e outra composta de textos científicos (Resumo e Título de Teses e Dissertações da COPPE/UFRJ, 2000 a 2006).

Os classificadores utilizados são o Naive Bayes, a Máquina de Vetor Suporte e um classificador baseado em regras de decisão.

Os resultados obtidos comprovam que o Naive Bayes oferece excelentes resultados na classificação de textos e mostra que os outros dois podem ser usados na classificação em combinação com o primeiro, oferecendo resultados ainda melhores.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

LINGUISTIC APPROACH IN AUTOMATIC CLASSIFICATION OF TEXTS IN PORTUGUESE LANGUAGE.

Yuri Barwick Lannes de Camargo

June/2007

Advisor: Jorge Lopes de Souza Leão

Department: Electrical Engineering

This work analyses an automatic classification of texts in portuguese using linguistic information in pre-processing phase of texts. To do this, two collections of texts were used, one using newspaper articles (Folha de São Paulo, 1994) and another using cientific texts (abstract and title of COPPE/UFRJ thesis and dissertations, 2000 to 2006).

The classifiers used were Naïve Bayes, Support Vector Machines and a classifier based on decision rules.

The results testify that Naïve Bayes presents excellents results in text classification and show that other two can be used in classification combinated with the first one presenting even better results.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1 – Introdução

Desde os primórdios da humanidade há a necessidade de armazenar ou registrar o conhecimento, tendo sido criados, até hoje, diversos mecanismos de armazenamento e recuperação destas informações, independente do meio de armazenamento. Assim, com o surgimento do computador, foram desenvolvidos mecanismos automatizados adaptados às atividades de guardar informação, e recuperá-la quando necessário, chamados de Sistemas de Recuperação de Informação (SRI). A área de pesquisa em Recuperação de Informação (RI) surgiu na década de 1950, com os sistemas para localização de livros em bibliotecas convencionais. As pesquisas em RI visam desenvolver metodologias que permitam um melhor desempenho dos SRIs em identificar a necessidade de informação do usuário e recuperar itens relevantes a esta necessidade (YANG, 1997). Contudo, com a quantidade crescente de informação textual armazenada, surge um fenômeno conhecido como sobrecarga de informação (*information overloading*). Em outras palavras, a quantidade crescente de informação disponível torna cada vez mais longa a busca por informações relevantes (YIN, 1996). As técnicas computacionais utilizadas para realizar esta atividade têm que estar adaptadas a este novo tempo, em que o volume de informações é muito grande e cresce basicamente de forma exponencial (RIZZI et al., 2001).

Atualmente, a acessibilidade a grandes coleções de documentos em formato eletrônico enfatizou a necessidade de técnicas de recuperação de informação inteligentes. Técnicas poderosas para automaticamente organizar e pesquisar coleções de documentos textos não estruturados têm sido objeto de pesquisas. Tais pesquisas visam aprimorar estas técnicas permitindo que elas possam lidar com o grande e crescente volume de informação, e oferecer respostas cada vez mais acuradas às necessidades dos usuários que procuram por elas. Estas pesquisas incluem a realização de experimentos diversos, na tentativa de contribuir para o avanço do conhecimento e das técnicas aplicadas nesta área (CORREA, 2002).

1.1 – Motivação

O processo de categorização de documentos foi, durante muito tempo, realizado manualmente por especialistas da área, por meio da leitura parcial ou total dos textos. Embora eficiente para pequenas quantidades de documentos, este método não conseguiu suprir as necessidades de classificação surgidas com o aumento da quantidade de informação disponível eletronicamente, e posteriormente, com o

crescimento desordenado da WWW (*World Wide Web*). Sua maior desvantagem reside no alto custo e na grande quantidade de tempo necessário para a execução da tarefa eficientemente, além da imprecisão em se determinar a classe dos documentos, já que os responsáveis por esta tarefa podem diferir em seus julgamentos sobre um mesmo documento. Assim, sistemas que aprendem automaticamente a categorizar documentos se tornam imprescindíveis (CORREA, 2002).

A disseminação da internet e a popularização dos computadores também trouxeram para os usuários dificuldades em encontrar a informação desejada, mesmo com a evolução dos mecanismos de busca automática, devido ao grande número de documentos recuperados. Em face disso, faz-se necessário a compartimentação desta informação de modo a diminuir o espaço de busca da mesma. Esta compartimentação deve ser supervisionada de modo a garantir que os textos exemplos pertençam de fato às categorias pré-estabelecidas, já que estes textos serão usados para treinamento dos classificadores. Embora seja desejável que o processo de recuperação da informação seja automático, não se pode prescindir da participação dos especialistas nas suas áreas de conhecimento. Esta participação garante que a seleção das amostras de textos para representar cada categoria seja a melhor possível, ou ainda, seja a mais abrangente sem a necessidade de ser enorme.

As pesquisas realizadas até o momento tratam em sua maioria de textos em inglês, mesmo aquelas realizadas por autores brasileiros, restando ainda muito a fazer com relação ao português.

1.2 - Objetivos

Esta tese pretende avaliar a eficácia de alguns dos métodos de classificação automática e dos diferentes procedimentos utilizados neste processo para textos em português.

Os métodos de classificação escolhidos foram o *Naive Bayes* que modela cada distribuição condicional com uma simples função Gaussiana (JOHN & LANGLEY, 1995), o *Support Vector Machines* (SVM), ou Máquina de Vetor Suporte, que mapeia no espaço n-dimensional os pontos mais próximos à superfície de separação dos conjuntos que representam cada classe (VAPNIK, 1995) e um classificador baseado em regras de decisão (MITCHELL, 1997). As coleções (denominadas corpus) utilizadas foram o conjunto de textos extraídos do Jornal Folha de São Paulo, do ano de 1994, elaborado pelo NILC – Núcleo Interinstitucional de Lingüística Computacional. São 855 textos classificados em cinco categorias: esporte, imóveis, informática, política e turismo, sendo 171 arquivos por classe, e o conjunto de textos formados pela junção do título e resumo das teses de pós-graduação (mestrado e

doutorado), dos anos de 2000 até 2006, da área de Engenharia Elétrica da Universidade Federal do Rio de Janeiro – COPPE/UFRJ. São 475 textos classificados nas categorias controle, microeletrônica, processamento de sinais, redes de computadores e sistemas de potência, sendo 95 arquivos por classe.

1.3 – Organização da dissertação

O presente documento está organizado em seis capítulos incluindo este capítulo introdutório, onde é feita uma pequena introdução sobre categorização de textos, acrescentando alguma descrição sobre as coleções e classificadores utilizados.

No capítulo 2 são descritos os sistemas de categorização, incluindo as etapas e as medidas de desempenho a serem aplicadas aos sistemas de recuperação de informação como um todo.

No capítulo 3 são descritos o pré-processamento usual e o lingüístico. É feita uma descrição da análise sintática, que é a componente básica para o pré-processamento lingüístico. Neste capítulo são descritos os modelos de representação de documentos, e os métodos de seleção e extração de características, i.e., os métodos para selecionarem-se os termos que representarão os vetores locais de cada classe (categoria).

No capítulo 4 são descritos os métodos de categorização supervisionada e não-supervisionada, i.e., os classificadores. Além disso, é neste capítulo que será feita a revisão bibliográfica enunciando os trabalhos realizados até o momento e as lacunas existentes que visam ser parcialmente cobertas por este trabalho

No capítulo 5 são descritos as coleções de textos utilizadas, os programas utilizados para as etapas descritas nos capítulos 3 e 4, e ainda, os experimentos, seus resultados e sua análise.

Por fim, no capítulo 6 são feitas as conclusões e proposições para trabalhos futuros.

Capítulo 2 – Categorização de documentos

Neste capítulo serão enunciados conceitos pertinentes a classificação de documentos de texto. Primeiramente será definido o processo de categorização, para após serem enunciadas suas etapas e as medidas de desempenho para avaliação da classificação.

2.1 – Processamento de documentos texto

Algumas tarefas de processamento de texto têm sido de interesse para pesquisadores na área de RI, são elas:

- recuperação de documentos, que consiste em selecionar um subconjunto de documentos a serem retornados e exibidos para o usuário. A seleção é usualmente baseada na consulta requisitada pelo usuário. A consulta pode ser expressa por um conjunto de palavras-chave, selecionadas pelo usuário como representativas da necessidade de informação;
- roteamento ou filtragem de documentos, que envolve a seleção de um subconjunto dos documentos textos disponíveis, de acordo com as necessidades de informação do usuário. Na filtragem de documentos, a necessidade de informação do usuário é tipicamente representada por um perfil do usuário (*user profile*), que pode ser um conjunto de consultas baseadas em palavras-chave especificadas pelo usuário. Entretanto, diferentemente da recuperação de documentos onde a necessidade de informação do usuário é de curto prazo, o perfil do usuário usado na filtragem de documentos representam as necessidades de informação do usuário em longo prazo; e
- categorização de documentos, que é a classificação de documentos em um conjunto de uma ou mais categorias. Na categorização de documentos, as categorias em que os documentos são classificados são predefinidas, tipicamente pelo projetista ou mantenedor do sistema de categorização. Os usuários finais geralmente não são envolvidos no processo de definir as categorias. Uma vez que os documentos estejam categorizados, o usuário pode identificar um conjunto de categorias que podem conter documentos relevantes para suas necessidades, e ignorar documentos em categorias que são provavelmente irrelevantes. Dessa maneira, o espaço de informação que o usuário tem para pesquisar é bastante reduzido, o que acelera o processo de encontrar informação relevante. Nesta dissertação será avaliado o processo de categorização de documentos.

O homem executa a categorização de texto lendo os textos e deduzindo as classes as quais pertencem através de expressões específicas e seus padrões de contexto. A classificação automática simula este processo e reconhece os padrões de classificação como uma combinação de características de texto. Estes padrões devem ser gerais o bastante para ter grande aplicabilidade, mas específicos o suficiente para serem seguros quanto à categorização de uma grande quantidade de textos (MOENS, 2000).

RIZZI et al. (2000) afirmam que a categorização de textos é uma técnica utilizada para classificar um conjunto de documentos em uma ou mais categorias existentes. Ela é geralmente utilizada para classificar mensagens, notícias, resumos e publicações. A categorização também pode ser utilizada para organizar e filtrar informações. Essa capacidade faz com que esta técnica possa ser aplicada em empresas, contribuindo no processo de coleta, análise e distribuição de informações e, conseqüentemente, na gestão e na estratégia competitiva de uma empresa.

A compartimentação propiciada pela classificação permite a organização automática de conteúdos necessária ao desenvolvimento de processos nos quais os executores das etapas não devem ter conhecimento das atividades desenvolvidas por todos, como nas atividades de projetos militares ou de segurança nacional, por exemplo.

Outra aplicação de sistemas de categorização é limitar o espaço de busca para sistemas de recuperação de informação. Além de especificar uma consulta, o usuário pode limitar o escopo da busca especificando um conjunto de categorias a serem pesquisadas. Dependendo do conjunto de categorias, pode haver sobreposição entre diferentes categorias, ou seja, um documento pode pertencer a mais de uma categoria. Em alguns casos, alguns documentos podem não pertencer a nenhuma das categorias definidas; dependendo do sistema de categorização, esses documentos podem ser removidos ou colocados em uma categoria especial. Os documentos processados são então armazenados no banco de dados junto com a lista de categorias atribuídas a cada um deles.

2.2 – Sistemas de categorização

A categorização de textos é a classificação de documentos com respeito a um conjunto de uma ou mais categorias pré-existentes. Uma definição formal para a tarefa de categorização de textos é de difícil formulação mas, neste trabalho, será esta a definição adotada. LEWIS (1992) a define como uma função cujo domínio seria todas as possíveis representações do objeto (no caso, os textos a categorizar), mapeado na pertinência do objeto a uma ou mais classes.

Uma aplicação muito comum da categorização de textos é como uma etapa anterior a indexação de documentos, a fim de facilitar uma posterior recuperação destes. A categorização de documentos pode ser aplicada a textos completos ou a partes destes, e nos formatos mais diversos, tais como:

1. resumos de documentos técnicos,
2. mensagens de correio eletrônico,
3. notícias e, etc.

Os métodos de categorização de textos podem realizar classificação:

1. Binária ou graduada – dependendo de como será expressa a relação de pertinência entre documentos e categorias. Na categorização binária cada documento é classificado como pertence (1) ou não (0) a cada uma das categorias. A categorização graduada ocorre quando cada documento recebe o grau de pertinência em relação a cada uma das categorias (LEWIS, 1992);
2. Simples ou múltipla – dependendo da existência ou não de interseção entre categorias. A categorização múltipla ocorre quando cada documento pode ser associado a mais de uma classe. Na categorização simples um documento só pode ser atribuído a uma categoria.

Tendo sido associado um grau de pertinência a cada uma das classes (categorização graduada), pode-se obter a categorização múltipla atribuindo os documentos às classes em que a probabilidade associada ultrapassa algum limiar pré-estabelecido, ou a categorização simples atribuindo a cada documento a classe de maior probabilidade associada.

A categorização binária é mais comum em pesquisas sobre categorização.

A maior particularidade, ou dificuldade na categorização de textos, assim como nos processos de indexação nos SRI, está na grande quantidade de termos que precisam ser processados, a chamada alta dimensionalidade do espaço de características. Este espaço de características se constitui de termos únicos ou compostos que são extraídos ou adaptados dos textos dos documentos processados. O fato é que podem existir centenas ou milhares destes termos até mesmo para uma coleção de textos de tamanho relativamente pequeno. Um exemplo disto pode ser visto no Capítulo 5, na Tabela 5.5 onde está enumerada a quantidade de termos por coleção utilizada neste trabalho.

2.3 – Etapas da categorização

Um Sistema de Categorização de Textos envolve as fases de indexação e resposta. Na primeira fase, os termos de um documento são analisados considerando a existência de categorias predefinidas. A fase de resposta informa ao usuário se aquele documento processado pertence a nenhuma, uma ou mais daquelas categorias.

Em um projeto de categorização de documentos, três etapas são consideradas (NEVES, 2001):

1. Preparação (ou pré-processamento) dos documentos: consiste na eliminação de caracteres e palavras irrelevantes e na seleção daquelas características mais representativas da coleção de documentos;
2. Representação dos documentos: consiste no processo de representação de um documento de modo que este possa ser compreendido pelo classificador a ser utilizado, ou melhor, na transformação dos textos em vetores numéricos a serem apresentados aos algoritmos de classificação;
3. Classificação: a categorização de texto requer uma especificação na qual se basear (modelo) de modo a decidir a que categoria o documento deverá ser atribuído.

Para a construção de um sistema de categorização automática de documentos, faz-se necessário dois conjuntos de documentos: um de treinamento e um de testes. Este primeiro será cuidadosamente analisado para a construção manual de um classificador, ou utilizado na fase de aprendizagem para a construção automática de classificadores. Estes conjuntos podem ser obtidos pela coleta manual de documentos em algum repositório existente (e.g., WWW, bibliotecas digitais) ou pelo uso de coleções padrões de documentos construídas para esta tarefa (e.g., Reuters, TREC).

A figura 2.1 apresenta uma representação deste processo.

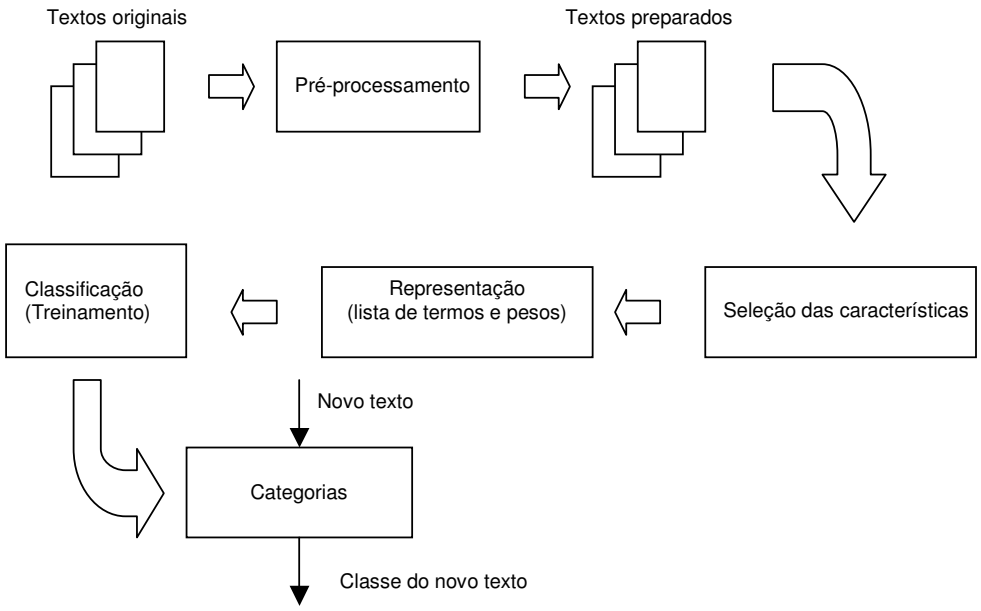

Figura 2.1 – Processo de categorização de textos.

2.4 – Medidas de Desempenho

De modo geral, a eficiência de um SRI, pode ser medida através da análise do grau de Abrangência e Precisão de suas respostas ao usuário, em relação a uma busca realizada por ele.

A Abrangência refere-se à porção das informações relevantes à busca que foram recuperadas. A Abrangência é máxima quando todas as informações relevantes à solicitação são recuperadas. A Precisão refere-se à porção das informações recuperadas que é realmente relevante. A Precisão é máxima quando não são fornecidas informações irrelevantes em resposta à solicitação.

A Abrangência e a Precisão são obtidas através das seguintes relações (RIJSBERGEN, 1979): seja A, o grau de abrangência em relação a uma dada categoria, dado por A = n/Nr onde n é o número de documentos relevantes recuperados e Nr é o número total de documentos relevantes e, seja P, o grau de precisão em relação a uma dada categoria e é dada por P = n/Nt onde Nt é o número total de documentos recuperados. Na figura 2.2 tem-se uma representação gráfica destas relações.

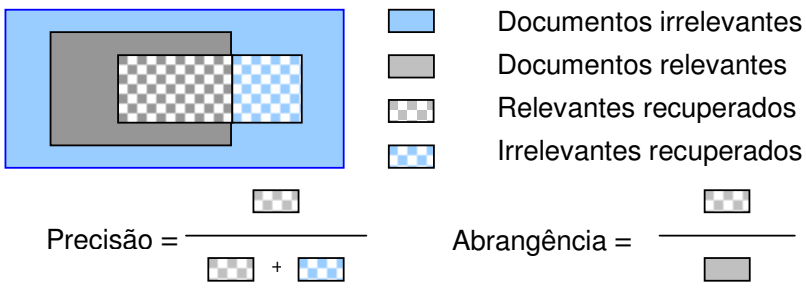

Figura 2.2 – Medidas de eficácia para Sistemas de recuperação da informação.

Outra medida de desempenho refere-se ao valor de erro médio obtido em cada classificação e consiste da relação entre o número de documentos corretamente classificados (C) e o número total de documentos a classificar (N). Assim, o erro tende a zero quando todos os documentos conseguem ser classificados corretamente, i.e., ERRO = C/N. Por motivo de simplificação será está a medida adotada neste trabalho.

A eficiência de um SRI é afetada pela maneira como o usuário representa e identifica seus documentos. Este processo envolve a verificação dos conteúdos ali tratados, que são elaborados através de uma linguagem. A análise automática de conteúdos pode ser feita através das frases e/ou termos que o documento contém. Esta análise pode ser realizada através de duas abordagens (RIZZI et.al., 2000):

- abordagem linguística – aplicando-se métodos semânticos e sintáticos no texto dos documentos;
- abordagem estatística – aplicando-se métodos que incluem seleção e contagem de termos nos documentos. Neste caso, o termo é o meio de acesso ao documento, e a maneira pela qual eles são identificados e diferenciados.

2.5 – Considerações

Neste capítulo foram enumeradas as atividades relativas ao processamento de textos. Dentre elas as principais são a recuperação, o roteamento ou filtragem e a categorização, que é o objeto deste trabalho. Para este trabalho será adotada a seguinte definição do processo de categorização: categorização de textos é a classificação de documentos com respeito a um conjunto de uma ou mais categorias pré-existentes.

Além de enunciadas as etapas envolvidas no processo de categorização (preparação, representação dos documentos e classificação) foram enumerados tipos

de categorização e algumas aplicações, e foram citadas as medidas de desempenho de modo a estabelecer quais os critérios serão usados nos experimentos.

No próximo capítulo será descrita a atividade de pré-processamento dos textos visando à preparação para submetê-los às ferramentas de seleção de termos para posterior codificação e classificação. De maneira simplificada, a preparação dos documentos consiste na leitura das palavras dos textos, o cálculo de ocorrência de cada uma delas e sua posterior transformação em pesos, de modo que estes pesos possam compor um vetor que representa a presença ou ausência da palavra no texto assim como sua importância relativa dentro do texto.

Capítulo 3 - Pré-processamento dos documentos

Neste capítulo serão abordadas as técnicas de pré-processamento dos textos, que consistem das atividades necessárias para transformação dos textos em vetores a serem submetidos aos classificadores. Serão abordados os dois métodos de análise das palavras dos textos e os métodos de seleção das palavras que representarão cada categoria.

3.1 – Pré-processamento usual

O pré-processamento usual dos documentos pode ser dividido em quatro fases principais, que podem estar presentes ou não em um sistema de categorização de documentos, de acordo com as necessidades dos desenvolvedores deste (NEVES, 2001):

- análise léxica,
- eliminação de palavras irrelevantes,
- remoção de afixos das palavras e
- a seleção dos termos de indexação.

Na análise léxica é feita uma adaptação do documento. Essencialmente, eliminam-se os dígitos e os sinais de pontuação e isolam-se os termos e efetua-se a conversão de letras de maiúsculas para minúsculas. São também retirados os conectores textuais e léxicos – conjunções e preposições – por serem consideradas entidades irrelevantes, além dos artigos. Este procedimento visa diminuir a extensão dos documentos. Algumas palavras que aparecem em grande quantidade dos textos são consideradas irrelevantes e denominadas *stopwords* e desta forma também são eliminadas. Consideremos como exemplo a frase “Janeiro começa com grandes liquidações.” Após a análise léxica obteríamos a seguinte estrutura: “janeiro começa com grandes liquidações”, em seguida após a retirada das *stopwords* obteríamos: “janeiro começa grandes liquidações”.

Outra etapa é a eliminação da variação morfológica de um termo, também denominada lematização ou algoritmo de stemming (FRAKES, 1992; KRAAIJ, 1996). Entretanto esta variação depende da tarefa a realizar, na categorização de documentos, por exemplo, a variação morfológica aumenta a discriminação entre documentos (RILOFF, 1995).

A aplicação de um algoritmo de eliminação da variação morfológica visa remover as vogais temáticas, as desinências, os prefixos e os sufixos de um termo. Por exemplo, as palavras *conectar, conectado e conectando* iriam gerar o mesmo

radical *conect*. Desta forma, nossa frase de exemplo resultaria em: ”janeir começ grande liquidaç”.

Apesar destes algoritmos serem aplicados com o intuito de melhorar o desempenho de sistemas de RI, alguns estudos mostraram que a redução de afixos não apresenta melhora significativa no desempenho da tarefa de mineração aplicada (HARMAN, 1991; LENNON et. al,1981).

3.2 – Pré-processamento lingüístico

O pré-processamento utilizando informações linguísticas é feito separando-se as palavras por suas funções sintática e gramatical. Para isso faz-se necessária a análise sintática dos textos.

3.2.1 – Análise sintática

A sintaxe é o componente do sistema lingüístico que determina as relações formais que interligam os constituintes da sentença, atribuindo-lhe uma estrutura. Nessa estrutura encontra-se o sintagma, ou seja, unidade da análise sintática composta de um núcleo (por exemplo, um verbo, um nome, um adjetivo, etc.) e de outros termos que a ele se unem, formando uma locução que entrará na formação da oração. O nome do sintagma depende da classe da palavra que forma seu núcleo, havendo assim sintagma nominal (núcleo substantivo), sintagma verbal (núcleo verbo), sintagma adjetival (núcleo adjetivo), sintagma preposicional (núcleo preposição) e sintagma adverbial (núcleo advérbio). Na teoria gerativa existem sintagmas formados por núcleos mais abstratos, como tempo, concordância, etc (SILVA, 2004).

Nas estruturas frasais observa-se uma hierarquia, sendo que esta pode ser representada por meio de um diagrama arbóreo: no topo encontra-se a unidade maior (a sentença), nos níveis intermediários os elementos sintáticos que formam a sentença (constituintes) e na base da estrutura, os itens lexicais correspondentes (as palavras).

A figura 3.1 ilustra um diagrama arbóreo de uma frase em língua portuguesa. Nesse exemplo a derivação da estrutura da sentença se constitui dos elementos indicados por S (sentença), N (substantivo), SV (sintagma verbal), V (verbo), SP (sintagma preposicional), PRP (preposição), SN (sintagma nominal) e ADJ (adjetivo) que, por sua vez, realizam os itens lexicais da base. Os elementos dos níveis intermediários são a expressão de um conjunto de informações necessário para a composição da sentença.

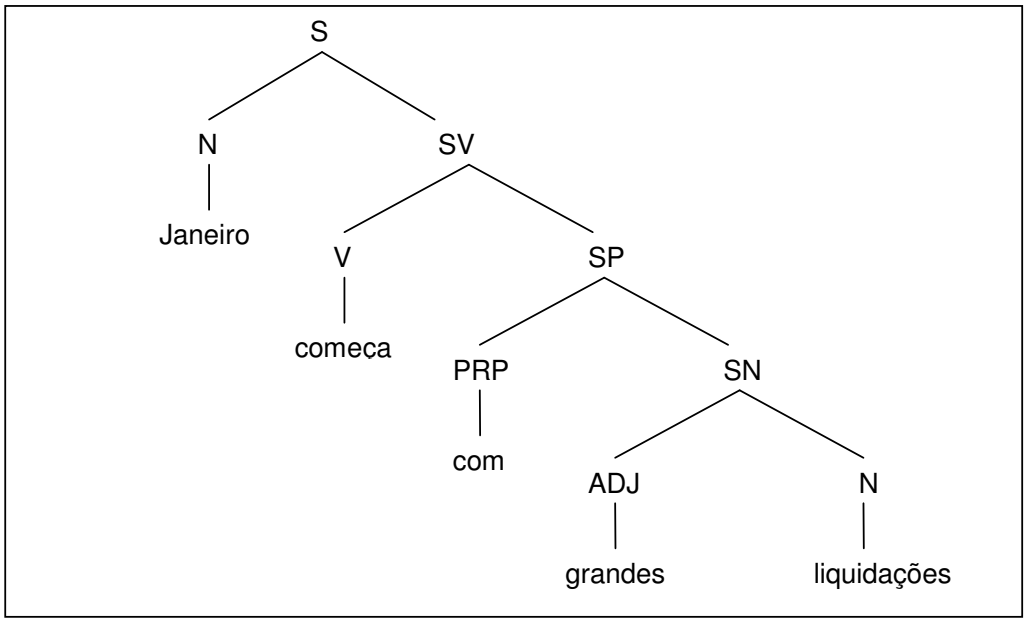

Figura 3.1 – Estrutura arbórea da sentença *Janeiro começa com grandes liquidações*.

Para construir a representação em árvore de uma sentença, é preciso conhecer quais as estruturas legais da língua, i.e., a gramática da língua (o conjunto de regras). Com o diagrama arbóreo é possível observar que, sob a aparente diversidade, todas as estruturas possuem uma organização interna que obedece a princípios gerais bem definidos.

Um conjunto de regras gramaticais descreve quais são as estruturas permitidas. Essas regras dizem que determinado símbolo pode ser expandido em árvore pela seqüência de outros símbolos. Cada símbolo é um constituinte da sentença que pode ser composto de uma ou mais palavras.

No nível computacional, para construir o diagrama arbóreo de uma sentença faz-se necessário um processamento sintático, ou analisador sintático, que é alcançado por intermédio da análise morfossintática.

Enquanto o analisador léxico-morfológico trabalha com a estrutura dos termos e com a classificação dos mesmos em diferentes categorias, o analisador sintático trabalha no nível do sintagma, tentando validar o agrupamento de termos que compõem as frases.

O processamento sintático é responsável por construir (ou recuperar) uma estrutura sintática válida para a sentença de entrada. Para tanto é orientado por uma representação da gramática da língua em questão. Em se tratando de uma língua natural, frequentemente se adota uma gramática parcial, que, embora não abranja todas as construções da língua, contempla aquelas construções válidas de interesse

para a aplicação. Assim, evita-se o grande volume de informações gramaticais que pode aumentar demasiadamente a complexidade de sua representação, bem como a complexidade do próprio processo de análise (SILVA, 2004).

Para auxiliar na construção de um processamento sintático, utiliza-se um analisador sintático. Esse, por sua vez, é um procedimento que pesquisa os vários modos de como combinar regras gramaticais, com a finalidade de gerar uma estrutura de árvore que represente a estrutura sintática da sentença analisada. Caso a sentença seja ambígua, o analisador sintático poderá obter todas as possíveis estruturas sintáticas que a representam. No entanto, os analisadores sintáticos podem apresentar variações de acordo com a relação que estabelecem com o usuário, os recursos disponíveis e as estratégias de análise.

O analisador sintático utilizado neste trabalho para extração de informações lingüísticas dos documentos é o *PALAVRAS*, desenvolvido por Eckhard Bick (BICK, 2000) para a língua portuguesa. Ele realiza tarefas como tokenização, processamento léxico-morfológico, análise sintática e faz parte de um grupo de analisadores sintáticos do projeto VISL (Visual Interactive Syntax Learning), do Institute of language and Communication da University of Southern Denmark.

O analisador sintático recebe como entrada o conjunto de sentenças de um corpus e gera a análise sintática das sentenças. A marcação sintática do analisador *PALAVRAS* para a sentença em língua portuguesa “Janeiro começa com grandes liquidações” pode ser vista na figura 3.2 abaixo.

```
SOURCE: live
1. running text
A1
STA:fcl
.
|-SUBJ:n('janeiro' M S)        Janeiro
|-P:v*fin('começar' PR 3S IND) começa
|-ADVL:pp
  |-H:prp('com')       com
  |-P&lt;:np
    |-&gt;N:adj('grande' F P)  grandes
    |-H:n('liquidação' F P)    liquidações

SOURCE: live
1. running text
A1
```

Figura 3.2 – Marcação do analisador sintático PALAVRAS.

Com base nas marcações do analisador sintático um conjunto de programas foi desenvolvido em cooperação com a Universidade de Évora: a ferramenta *Xtractor* – apresentada em GASPERIN et al. (2003). A ferramenta engloba a análise do corpus

por meio do *PALAVRAS*, e o tratamento da saída do *parser* com geração de três arquivos XML.

O primeiro arquivo XML, chamado de *words*, contém as palavras do corpus com elementos <word> e atributos id que representam um identificador único para cada termo do texto.

```
<words>
<word id="word_1">Janeiro</word>
<word id="word_2">começa</word>
<word id="word_3">com</word>
<word id="word_4">grandes</word>
<word id="word_5">liquidações</word>
<word id="word_6">. </word>
</words>
```

Figura 3.3 – Arquivo Words

O segundo arquivo apresenta as informações morfossintáticas das palavras do texto, denominado de POS (*Part-of-speech*), onde o elemento <n> indica um substantivo e o elemento <v> um verbo.

```
<words>
<word id="word_1">
<n canon="Janeiro" gender="M" nimber="S"/>
</word>
<word id="word_2">
<v canon="começar">
<fin tense="PR" person="3S" mode="IND"/>
</v>
</word>
<word id="word_3">
<prp canon="com"/>
</word>
<word id="word_4">
<adj canon="grande" gender="F" number="P"/>
</word>
<word id="word_5">
<n canon="liquidação" gender="F" number="P"/>
<word id="word_6">
</word>
</words>
```

Figura 3.4 – Arquivo Pos (Informações morfossintáticas).

O terceiro arquivo consiste nas estruturas e subestruturas sintáticas das sentenças, representadas por *chunks*. Um *chunk* representa a estrutura interna da sentença e pode conter *sub-chunks*. Estes *sub-chunks* representam todas as possíveis combinações em sequência entre as palavras de uma sentença.

```
<text>
<paragraph id="paragraph_1">
<sentence id="sentence_1" span="word_1..word_6">
<chunk id="chunk_1" ext="sta" form="fcl" span="word_1..word_5">
<chunk id="chunk_2" ext="subj" form="n" span="word_1">
</chunk>
<chunk id="chunk_3" ext="p" form="v_fin" span="word_2">
```

```
</chunk>
<chunk id=”chunk_4” ext=”adv1” form=”pp” span=”word_3..word_5”>
<chunk id=”chunk_5” ext=”h” form=”prp” span=”word_3”
</chunk>
<chunk id=”chunk_6” ext=”p” form=”np” span=”word_4..word_5”>
<chunk id=”chunk_7” ext=”n” form=”adj” span=”word_4”>
</chunk>
<chunk id=”chunk_8” ext=”h” form=”n” span=”word_5”>
</chunk>
</chunk>
</chunk>
</chunk>
</sentence>
</paragraph>
</text>
```

Figura 3.5 – Arquivo Chunks

3.2.2 – Extração de categorias morfossintáticas

Gerados os arquivos no formato XML, a extração de estruturas é realizada aplicando folhas de estilo XSL (*eXtensible Stylesheet Language*) nos arquivos de POS (informações morfossintáticas).

XSL é um conjunto de instruções destinadas à visualização de documentos XML. Desta forma, dado um documento XSL é possível transformar um documento XML em diversos formatos. A linguagem XSL auxilia a identificação dos elementos de um documento XML permitindo a simplificação do processamento de transformação desses elementos em outros formatos de apresentação. Contudo, é possível criar múltiplas representações da mesma informação a partir de vários documentos XSL aplicados a um único documento XML.

Uma folha de estilos é composta por um conjunto de regras, chamados *templates*, ativados no processamento de um documento XML Neste trabalho foram implementadas folhas de estilo para extrair combinações morfossintáticas tais como: substantivo; substantivo e adjetivo; substantivo e nome próprio; substantivo, nome próprio e adjetivo; substantivo e verbo; substantivo, verbo e adjetivo; nome próprio e adjetivo; e verbo assim como em SILVA (2004). Desta forma, obtêm-se dos textos somente as palavras que o representarão, excluindo a análise léxica e a eliminação de palavras irrelevantes.

3.2.3 – Extração de categorias sintáticas

É realizado aplicando-se folhas de estilo XSL aos arquivos de *chunks* extraindo dos mesmos as seguintes estruturas: sujeito, verbo, objeto direto, objeto indireto, agente da passiva e, predicativo do sujeito. As folhas de estilo extraem as palavras dos arquivos de *words*.

3.3 – Modelos de representação de documentos

Os sistemas tradicionais de recuperação de informação, por questões de eficiência e concisão, evitam trabalhar com o texto completo dos documentos, adotando termos ou índices para os processos de representação, classificação e recuperação dos mesmos. Esses termos consistem em palavras isoladas ou grupos de palavras relacionadas, presentes no texto tratado.

A idéia fundamental é que a semântica do documento e a consulta do usuário podem ser expressas por um conjunto de termos de indexação. Os documentos passam a ser representados por vetores, onde os índices correspondem aos termos utilizados e os valores em cada posição representam a importância do termo dentro do documento (CORREA, 2002).

Em uma coleção de documentos, uma palavra que aparece na maioria dos documentos não é um bom termo de indexação, já que não é representativa de nenhum documento em particular. Contudo, uma palavra que aparece em algumas poucas unidades dos documentos desta coleção poderá ser bastante útil para a identificação desses. Portanto, existem palavras que são mais significativas que outras, e por isso, é importante a atribuição de pesos aos termos de modo a se medir a importância dos mesmos para o documento tratado.

Para a tarefa de representação dos documentos e determinação do peso associado a cada termo em um documento, serão apresentados dois modelos: o *booleano* e o vetorial (BAEZA-YATES & RIBEIRO-NETO, 1999).

O modelo *booleano* avalia a presença ou ausência do termo de indexação no documento, portanto, os pesos atribuídos a esses termos são binários, isto é {0,1}. Sendo cada categoria representada por um ou mais vetores binários, correspondentes a disjunção de conjunções de termos encontrados nos documentos nela contidos, não há um casamento parcial entre o documento e a categoria, mas apenas uma decisão binária se o documento pertence ou não a esta. A decisão de quantos vetores *booleanos* serão utilizados para representar uma categoria depende da capacidade em se especificar o conteúdo dos documentos a ela pertencentes apenas por uma conjunção de termos ou por uma disjunção de conjunções de termos. As maiores vantagens do modelo *booleano* são sua simplicidade e por exigir menos espaço de armazenamento. A maior desvantagem é a de não oferecer uma ordem de relevância dos documentos dentro de uma categoria.

O modelo vetorial tenta contornar as limitações do modelo *booleano* utilizando pesos não binários e, consequentemente, permitindo um casamento parcial entre os documentos e as categorias. Segundo este modelo os termos se tornam dimensões e os valores informam a relevância (peso) dos termos. Os pesos atribuídos aos termos

de indexação funcionam como um grau de similaridade entre os vetores documentos, e entre os vetores documentos e os vetores representativos das categorias. Para obtermos a similaridade entre dois vetores, basta aplicar o produto interno dividido pelo produto das normas entre esses dois vetores. Cada categoria pode ser representada como um conjunto de vetores resultantes do somatório dos documentos pertencentes aos respectivos subconjuntos dela. No caso em que não há uma variância muito grande entre os documentos de uma categoria, apenas um vetor resultante do somatório de todos os documentos a ela pertencentes pode ser utilizado para representá-la. Na figura 3.6 é exemplificado o modelo de representação adotado neste trabalho.

Matriz termo x documento	Termo 1	Termo 2	Termo 3	Termo 4	Termo 5	...	Termo n	Classe
Documento 1	0,1	0,1	0,5	0,2	0	0	0	1
Documento 2	0,2	0,3	0	0,2	0	0	0	1
Documento 3	0	0	0,5	0,4	0,6	0	0	2
Documento 4	0	0	0,1	0,2	0,3	0,1	0	2
Documento 5	0	0,1	0	0	0,5	0,6	0,5	3
Documento 6	0	0	0	0,1	0,2	0,2	0,4	3
Documento 7	0	0	0	0	0,6	0,8	0,4	4
...								...
Documento n	0,1	0,1	0,2	0	0	0,5	0,5	n

Figura 3.6 – Representação de documentos pelo modelo vetorial.

O peso de relevância de cada termo para um documento pode ser a simples freqüência de ocorrência do mesmo no texto. Outra forma bastante comum de se determinar esses pesos é pelo método do TFIDF (SALTON & BUCKLEY, 1988), que se baseia no cálculo de dois parâmetros: o *tf* (*term frequency*) e o *idf* (*inverse document frequency*). O *tf* mede a importância de um termo para um documento, enquanto que o *idf* verifica o quão discriminante é o termo. Termos que aparecem em um grande número de documentos não possuem muito poder discriminativo. A composição destes dois parâmetros permite a atribuição de pesos aos termos de acordo com sua importância para cada documento.

3.4 – Métodos de seleção de características

Para representarem-se os textos faz-se necessária a seleção das palavras que os representarão, pois em virtude do tamanho dos textos e de sua quantidade o

número de palavras pode ser intratável pelos algoritmos de classificação. Este processo é conhecido como redução de características.

A redução de características pode ser do tipo:

- global (*global dimensionality reduction*) (APTÉ et al., 1994a) consististindo na escolha dos termos mais importantes para a coleção, ou
- local (*local dimensionality reduction*) (WIENER et al.,1995), consistindo na escolha de um conjunto de termos para cada categoria.

A natureza da redução também pode ser qualificada de duas formas: por meio da seleção ou da extração das características dos documentos da coleção.

Na primeira abordagem, as características selecionadas são um subconjunto das características originais. Esta redução pode ser realizada através da seleção daquelas melhores, de acordo com algum critério. Esta é uma tarefa de certa complexidade, pois se deve selecionar um subconjunto de termos que possa fazer uma discriminação adequada entre as várias categorias, e ao mesmo tempo, ser pequeno o suficiente para que possa ser utilizado pelo classificador. As técnicas mais utilizadas para este fim são baseadas na freqüência de palavras (NG et al., 1997) ou na freqüência de documentos. Técnicas como ganho de informação (medida de entropia) (YANG & PEDERSEN, 1997), o coeficiente de correlação (NG et al., 1997), a técnica do qui-quadrado (NG et al., 1997) e o método de Escore de Relevância (WIENER et al.,1995). Segue descrição detalhada de alguns deles.

- Frequência absoluta (*Term frequency*): quantidade de vezes que um termo ocorre em determinado documento.
- Frequência relativa: é a frequência absoluta dividida pelo número de palavras do documento.
- Frequência do documento: é o número de documentos em que o termo aparece.
- Frequência absoluta/frequência do documento (TFIDF): é a razão entre estas duas medidas.
- Ganho de informação (YANG & PEDERSEN, 1997):

É um critério que define a qualidade de cada termo. Ele mede a quantidade de pequenos pedaços ou partições de informação, obtidos para a predição da categoria através da presença ou ausência de um termo no documento. Este método é comumente utilizado no campo de aprendizagem de máquina e na construção de árvores e regras de decisão.

Os autores afirmam que a categorização de textos, normalmente, possui um espaço dimensional muito grande, alcançando até dezenas de milhares de

características, e é preciso calcular a qualidade do termo de maneira global. A partir de um conjunto de textos de treinamento, para cada termo único é calculado o ganho de informação. Os termos que não alcançarem um limiar pré-definido serão excluídos.

A idéia principal deste método é dividir o conjunto de exemplos em partições ou subconjuntos de exemplos, sendo estes subconjuntos compostos de exemplos de uma mesma classe ou similares. Ao grupo aplica-se o cálculo do ganho. O conjunto vai sendo subdividido repetidamente até que um subconjunto contenha apenas exemplos de uma única classe ou o número de exemplos seja inferior a um limite estabelecido. A conclusão é que o ganho de informação reduz os ruídos conforme o conjunto vai sendo subdividido, de forma que, no final, o último subconjunto será composto apenas por exemplos similares.

Nos estudos de MLADENIC e GROBELNIK (1998), a fórmula apresentada para o cálculo do ganho de informação está baseada em probabilidade e é a seguinte:

$$Ganho\_Info(F) = P(W) \times \sum_i P(Ci|W) \times \log \frac{P(Ci|W)}{P(Ci)} + P(\overline{W}) \times \sum_i P(Ci|\overline{W}) \times \log \frac{P(Ci|\overline{W})}{P(Ci)} \quad (1)$$

onde: $Ganho\_Info(F)$ é o ganho de informação da característica F; F é a característica que representa a palavra W; $P(W)$ é a probabilidade de ocorrer a palavra W; $P(Ci|W)$ é a probabilidade condicional de ocorrer a palavra W na *i-ésima* classe; $P(Ci)$ é a probabilidade da *i-ésima* classe; $P(\overline{W})$ é a probabilidade de não ocorrer a palavra W; $P(Ci|\overline{W})$ é a probabilidade condicional de não ocorrer a palavra W na *i-ésima* classe.

- A entropia cruzada esperada foi utilizada nos estudos de MLADENIC e GROBELNIK (1998), baseando-se no ganho de informação. Ela considera apenas o valor que denota a ocorrência da palavra analisada em um documento em vez da média global como considerado no ganho de informação. Eles afirmam que esta diferença originou melhores resultados de desempenho. A fórmula apresentada para o cálculo é a seguinte:

$$Entropia\_cruzada(F) = P(W) \times \sum_i P(Ci|W) \times \log \frac{P(Ci|W)}{P(Ci)} \quad (2)$$

Os demais termos da fórmula são os mesmos definidos na fórmula para cálculo de ganho de informação.

- Informação mútua e χ^2(qui) estatístico:

A informação mútua e χ^2 estatístico são dois métodos bastante semelhantes e podem ser usados para seleção de características.

YANG & PEDERSEN (1997) dizem que considerando um termo t e uma categoria c, A é o número de vezes que t e c acontecem juntos, B é o número de vezes que t aparece sem c, C é o número de vezes que c aparece sem t, e N é o número total de documentos, então o critério de informação mútua entre t e c é calculado por:

$$I(t,c) = \log \frac{A \times N}{(A+C) \times (A+B)} \quad (3)$$

Dada a situação quando se tem um termo t e uma categoria c, A o número de vezes que t e c acontecem juntos, B é o número de vezes que t aparece sem c, C é o número de vezes que c aparece sem t, D é o número total de vezes que nem c nem t aparecem, e N é o número total de documentos, então o critério χ^2 estatístico entre t e c é calculado por:

$$\chi^2(t,c) = \frac{N \times (A \times D - C \times B)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)} \quad (4)$$

O χ^2 tem valor zero se t e c são independentes. A medida é computada para cada termo e cada categoria, e então combinados para cada termo através da média ou do máximo dos valores encontrados para ele em cada categoria, formando o peso do termo. O peso é o valor que denota a importância de cada termo na distinção entre categorias da coleção; então, os termos com maiores pesos são selecionados como os mais relevantes. A principal diferença entre estes dois métodos é que χ^2 é um valor normalizado.

- Força do termo:

Em YANG e PEDERSEN (1997) e em YANG (1995), está definido que o método força do termo calcula a importância de um termo em um documento baseado na probabilidade deste termo também aparecer em quaisquer outros documentos da coleção.

A idéia é, em uma coleção de documentos, buscar a semelhança entre pares de documentos relacionados. A relação entre os documentos pode ser calculada através da similaridade por cosseno. A força do termo t está na probabilidade de encontrá-lo em um documento relacionado a outro documento, no qual o termo t é encontrado, e é definida por:

$$s(t) = P(t \in y \mid t \in x) \quad (5)$$

sendo x e y dois documentos quaisquer e relacionados de uma coleção e P a probabilidade condicional do termo t pertencer ao documento x dado que t pertence ao documento y.

- Escore de relevância (WIENER et al.,1995)

Esta técnica foi proposta e aplicada inicialmente no estudo de WIENER, PEDERSEN e WEIGEND (1995). SALTON (1983) propôs que a indexação de textos feita a partir de termos com pesos associados, alcança melhores resultados por estar definindo, a partir do peso, o grau de importância que o termo tem dentro do texto. A idéia inicial deste estudo de Salton foi calcular a freqüência com que cada termo aparece no documento. Depois, determinou a freqüência do termo dentro do texto e dentro da coleção, dando origem a técnica de freqüência inversa de documentos, salientando a idéia de que termos com grande capacidade de representação de conteúdo devem possuir alta freqüência no documento e baixa freqüência na coleção. E com isso, ele definiu a técnica do cálculo do peso de relevância do termo.

Os resultados obtidos por Salton utilizando o cálculo do peso de relevância do termo, inspiraram Wiener a propor o escore de relevância para a categorização de textos. A idéia desta técnica está baseada na freqüência de um termo em uma categoria e na sua freqüência nas demais categorias.

A seguir a fórmula para o cálculo do escore de relevância apresentada por WIENER, PEDERSEN e WEIGEND (1995):

$$r_k = \log \frac{\frac{w_{tk}}{d_t} + \frac{1}{6}}{\frac{w_{\bar{t}k}}{d_{\bar{t}}} + \frac{1}{6}} \qquad (6)$$

onde: r_k é o escore de relevância do termo k; w_{tk} é o número de documentos pertencentes a uma dada categoria t que contém o termo k em processo de análise; $w_{\bar{t}k}$ é o número de documentos de outras categorias que contém o termo k em análise; $d_{\bar{t}}$ é o número total de documentos de outras categorias; d_t é o número total de documentos da categoria t; *1/6* é uma constante para evitar divisão por zero.

- Coeficiente de correlação (NG et al., 1997):

Coeficiente de Correlação entre o termo *t* e a classe *c* é definido por:

$$C(t,c) = \frac{(N_{r+} \times N_{n-} - N_{r-} \times N_{n+}) \times \sqrt{N}}{\sqrt{(N_{r+} + N_{r-}) \times (N_{n+} + N_{n-}) \times (N_{r+} + N_{n+}) \times (N_{r-} + N_{n-})}} \qquad (7)$$

onde N_{r+} é o número de documentos relevantes para Cj que contêm o termo *t,* N_{r-} é o número de documentos relevantes para Cj que não contêm *t*, N_{n+} é o número de documentos não relevantes para Cj que contêm t e N_{n-} é o número de documentos não relevantes para Cj que não contêm *t*. Esta medida corresponde a raiz quadrada

do valor obtido pela métrica do χ^2 (Qui-quadrado). O coeficiente de correlação é maior para as palavras que indicam a pertinência de um documento à categoria Cj, enquanto a métrica do χ^2 gera valores maiores não só para este conjunto de palavras mas também para aquelas palavras que indicam a não pertinência à Cj.

A abordagem da seleção de características possui vantagens significativas, tais como sua simplicidade computacional e a interpretabilidade direta do conjunto de características resultante. Entretanto, algumas de suas desvantagens como não reduzir informação redundante (termos correlacionados) e a exclusão de termos individuais não significativos que poderiam ter grande poder discriminativo em combinação com outros, abrem espaço para pesquisas com técnicas de extração de características.

Métodos automáticos de seleção de características incluem a remoção de termos não informativos e a construção de um conjunto de elementos, termos ou radicais, que representam os textos e facilitam a identificação da categoria a que pertencem. É interessante que estes elementos combinem alta representatividade em menor número possível de características.

Após serem identificados os termos mais relevantes aplica-se a filtragem baseada no peso do termo (seleção do peso do termo ou truncagem) que consiste em estabelecer um número máximo de características a serem utilizadas para representar um documento que visa diminuir o número de elementos que compõem o vetor do documento. Tal procedimento não influi negativamente nos resultados, conforme testes realizados por SCHUTZE (1997), além de oferecer um ganho de performance no algoritmo.

Para representação dos conjuntos de documentos de cada classe são elaborados vetores locais compostos dos n termos mais relevantes de acordo com o cálculo de relevância. Assim, os vetores globais que representarão os documentos serão a junção dos vetores locais de cada classe e servirão de índices para os vetores de cada exemplo e as posições correspondentes representarão a importância do termo no documento.

3.5 – Métodos de extração de características

Na segunda abordagem, as características são obtidas pela combinação ou transformação das originais, através da síntese de um subconjunto *n'* de novas características, a partir do conjunto de características *N'* original, de modo a maximizar a eficácia do sistema. Os métodos de extração de características têm como principal objetivo a criação de características artificiais que não sofram dos problemas de polissemia (termo lingüístico que possui vários sentidos) e sinonímia (palavras que possuem o mesmo significado semântico). Uma técnica bastante utilizada para este fim é a Semântica Latente (*Latent Semantic Indexing*) (WIENER et al.,1995, SEBASTIANI, 1999). Este método consiste em organizar uma matriz onde cada coluna corresponde ao histograma da freqüência dos termos obtido em cada documento da coleção e então decompor o espaço expandido pelos vetores coluna em um conjunto ordenado de fatores por um método chamado Decomposição em valores singulares (*singular-value decomposition*-SVD). A decomposição tem a propriedade de que os últimos fatores têm influência mínima sobre a matriz. Os fatores que menos influenciam podem ser descartados, diminuindo a dimensionalidade.

Em KAESTNER ([s.d]) consta que a LSI considera que, se dois documentos A e B não possuem termos comuns entre si, mas ambos possuem termos comuns com um terceiro documento C, então A e B devem ser considerados similares.

A LSI mapeia os documentos em um vetor com menos termos no índice. Este índice passa a ser composto por conceitos em mais alto nível de abstração. A idéia é construir uma matriz de termos por documento na qual seus elementos representam a freqüência de cada termo no documento. A figura 3.7 mostra o processo de mapeamento de (Mij). Segundo Kaestner, a LSI ocorre da seguinte forma: tendo-se uma matriz Mij de dimensões T x N, sendo T o número de termos do índice e N o número de documentos, cada elemento desta matriz está associado a um peso wij relativo ao par [ki, dj], sendo k o termo e d o documento.

A matriz Mij pode ser decomposta em três matrizes através da SVD como segue:

$$(M_{ij})=(U)(D)(V)^t \qquad (8)$$

sendo: (U) a matriz dos autovetores derivada de $(M)(M)^t$; $(V)^t$ a matriz dos autovetores derivada de $(M)^t(M)$; (D) a $r \; x \; r$ matriz diagonal dos valores singulares. Sendo $r = min\ (T,N)$ e os valores singulares referentes às raízes quadradas positivas dos autovalores de $(M)(M)^t$.

A partir desta decomposição, é definida a nova matriz $(A)_k$ com dimensões menores. Para a definição da nova matriz efetua-se o seguinte processo:

- Na matriz (D) são selecionados somente os k maiores singulares formando a matriz $(D)_k$,
- São mantidas as colunas correspondentes das matrizes (U) e $(V)^t$ formando as matrizes $(U)_k$ e $(V)^t_k$,
- A nova matriz $(A)_k$ é dada por:

$$(A)_k = (U)_k(D)_k(V)^t{}_k \qquad (9)$$

onde k é a dimensionalidade do espaço conceitual e k < r. O parâmetro k deve ser grande o suficiente para permitir adesão às características dos dados e pequeno o suficiente para eliminar detalhes não relevantes à representação.

Uma desvantagem da semântica latente está no fato de que termos significativos ao contexto podem ser perdidos na definição do vetor novo.

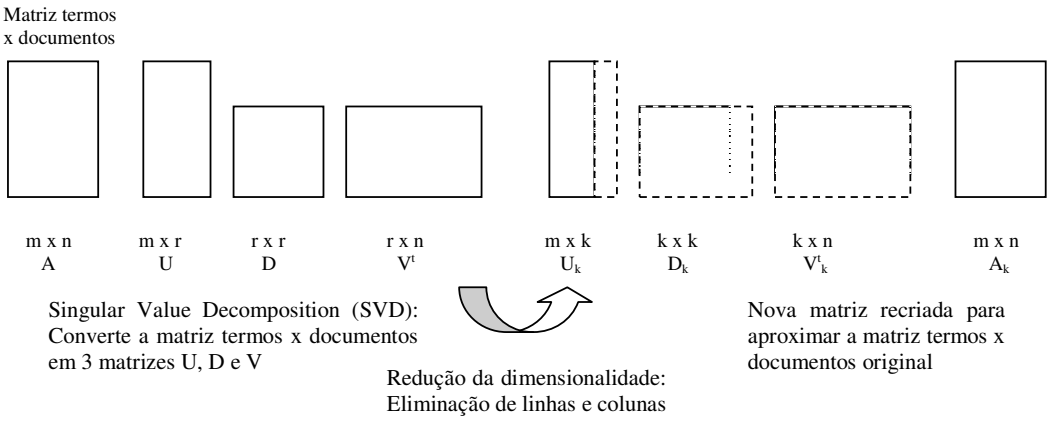

Figura 3.7 – Redução de vetores por LSI.

3.6 – Considerações

Para a apresentação dos textos aos classificadores são necessárias diversas atividades dentre as quais a seleção das palavras e sua conseqüente transformação em vetores numéricos de modo que a classificação passe a ser uma operação matemática. Neste capítulo foram listadas as possíveis maneiras matemáticas de selecionarem-se os termos representativos de cada categoria. Além disso, foram abordados os dois métodos de obtenção das palavras dos textos, o método lingüístico e o método usual. No método lingüístico as palavras são selecionadas pelas suas funções sintáticas e desta forma, é possível selecionar-se dentre as classes gramaticais aquelas que se pretende utilizar para representação dos textos. No método usual as palavras são selecionadas apenas pelo seu conteúdo semântico, sem distinção de suas funções nas sentenças.

Neste trabalho adotou-se o método lingüístico por considerar-se que desta forma agrega-se valor semântico às palavras dos textos e assim, no processo de seleção, as palavras passam a ter mais conteúdo informativo do que se consideradas apenas como seqüências de caracteres.

Os métodos de seleção das palavras escolhidos neste trabalho para representação das classes foram o Escore de relevância, o Coeficiente de correlação e a freqüência do documento. Desta forma, pretende-se comparar métodos que consideram no cálculo de relevância duas informações, a freqüência das palavras na classe e também a freqüência destas mesmas palavras nos textos das outras classes. Espera-se que o resultado dos dois primeiros métodos seja melhor que o terceiro em função desta metodologia, já que os dois primeiros consideram a freqüência das palavras nos textos das não-classes e o terceiro apenas a freqüência das palavras na classe. A redução de características adotada foi do tipo local, i.e., foi selecionado o conjunto mais relevante de termos representativos para cada categoria através dos três métodos de cálculo de relevância retromencionados, este processo é conhecido por truncagem.

No capítulo 5 será avaliado o processo de decomposição por valor unitário na matriz composta pelos vetores representativos de cada texto. Esta operação matemática pretende eliminar os termos menos relevantes escolhidos para representarem-se as categorias e com isso otimizar a seleção dos termos que representarão cada classe.

Capítulo 4 – Métodos de categorização

Neste capítulo serão enumerados os diversos algoritmos de classificação supervisionada que podem ser utilizados na tarefa de categorização de textos. São estes algoritmos que irão receber a representação vetorial dos textos e realizar a classificação.

São enumerados alguns exemplos de categorização não-supervisionada e também de métodos para a combinação de classificadores.

No final deste capítulo é feita a revisão bibliográfica de modo a situar este trabalho no contexto dos resultados obtidos até o momento.

4.1 – Métodos de categorização supervisionada

Os sistemas de categorização de documentos baseados em aprendizagem podem ser criados automaticamente através da apresentação dos exemplos de treinamento para um algoritmo de aprendizagem e posteriormente testados através da classificação dos exemplos de teste. Estes conjuntos de dados podem ser obtidos pela coleta manual e pré-processamento de uma coleção de documentos em algum repositório existente (e.g., Web, bibliotecas digitais) ou pelo uso de coleções padrões construídas para esta tarefa (e.g., Reuters, TREC).

Os métodos de categorização de textos são procedimentos que efetivamente classificam documentos com respeito a um conjunto de nenhuma, uma ou mais categorias existentes. Assim, cada exemplo (padrão) de treinamento e teste deve conter a informação descritiva do objeto (documento) e a codificação das classes (categorias) a que ele pertence.

Os sistemas baseados em aprendizagem passam por três fases distintas:

(1) treinamento – quando é realizada a aprendizagem por meio do conjunto de exemplos de treinamento;

(2) teste – quando se avalia o desempenho final do sistema por meio de um conjunto de exemplos de teste;

(3) e a fase de uso - quando o sistema está pronto para ser utilizado pelo usuário e realizar a tarefa de classificação.

Adota-se um conjunto de documentos específicos para a fase de treinamento e outro conjunto de documentos para a fase de testes, desta forma, a avaliação do sistema no conjunto de teste se torna mais realista, já que o algoritmo é testado em uma coleção independente da coleção que o treinou.

Um fenômeno presente na construção de sistemas baseados em aprendizagem é a saturação (*overfitting*). Este fenômeno se caracteriza quando o modelo gerado pelo algoritmo de aprendizagem obtém um ótimo desempenho na classificação do conjunto de treinamento, mas classifica erroneamente exemplos desconhecidos; isto ocorre porque, durante o treinamento, o algoritmo de aprendizagem passa a se concentrar nas peculiaridades dos exemplos no conjunto de treinamento e perde regularidades necessárias para uma boa generalização (GEMAN et al., 1992). Para cada algoritmo de aprendizagem existem técnicas que visam reduzir o efeito deste fenômeno.

Os algoritmos de aprendizagem mais utilizados na construção de sistemas de categorização de textos são descritos a seguir.

4.1.1 - Árvores de Decisão

A utilização das árvores de decisão na Inteligência Artificial tornou-se popular após o desenvolvimento do algoritmo ID3 (QUINLAN, 1986). Para a construção das árvores de decisão, é utilizado o aprendizado não-incremental a partir de exemplos. O conjunto de exemplos de treinamento é apresentado ao sistema que induz a árvore, construída de cima para baixo (da raiz para as folhas) com base na informação contida nos exemplos. A ordem em que os exemplos são apresentados, não interfere na árvore gerada.

Os nós da árvore correspondem aos atributos utilizados na representação dos objetos, enquanto que os ramos representam valores alternativos predeterminados para estes atributos.

A cada nó, está associado um teste, que pode ser binário ou multi-valorado. Neste último caso, se o atributo puder assumir *n* valores, *n* será o número de galhos provenientes deste nó, um para cada valor. Em alguns casos, a árvore resultante é modificada no final do processo de criação, por meio da poda de alguns dos galhos em folhas, galhos que exercem pouca influência no processo de classificação dos dados. A Figura 4.1 ilustra um exemplo de árvore gerada para a coleção Metais (Reuters-21578). Cada nodo folha é representado de forma retangular e indica a categoria a que pertencem os documentos que o atingem, bem como o número de documentos corretamente categorizados (acertos) e o número de documentos incorretamente categorizados (erros) do conjunto de treinamento. Cada nodo, exceto os nodos folha, contém um teste referente a presença (representado por “> 0”) ou ausência (representado por “<= 0”) de um termo no documento (CORREA, 2002).

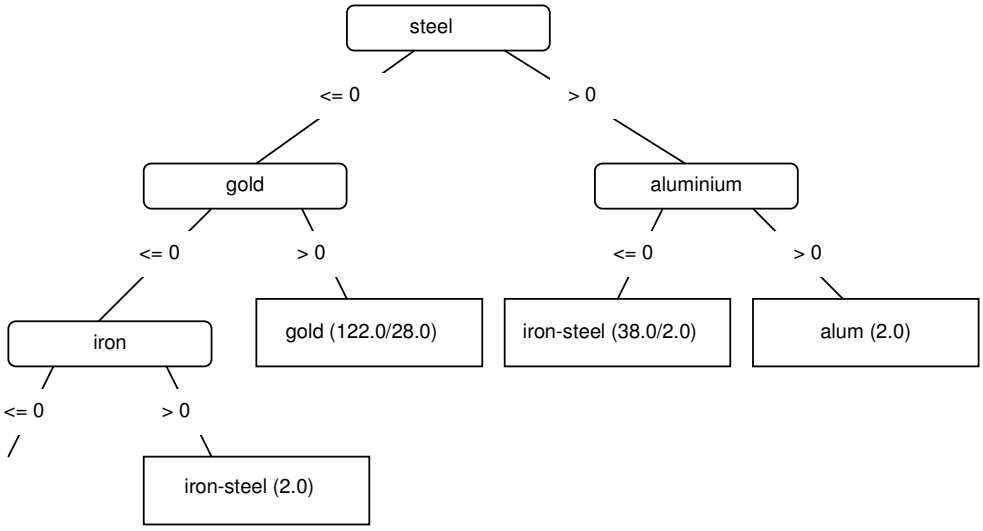

Figura 4.1 - Visão parcial da árvore de decisão.

Para sistemas de categorização de texto, pode-se adotar a construção de uma árvore independente para cada categoria considerada, ou uma única árvore para a coleção. Na utilização de uma árvore por coleção, somente poderá ser realizada categorização simples; a construção de uma árvore por categoria permite a realização de categorização simples ou múltipla. Entretanto, a construção de várias árvores independentes, uma para cada categoria, torna mais complexas as tarefas de aprendizado e categorização para coleções com grande número de categorias.

Após a sua construção, cada árvore poderá ser utilizada para a classificação de documentos descritos em termos dos mesmos atributos usados na sua representação. Isto é feito percorrendo-se a árvore recursivamente, até se chegar à folha que determina a classe a que o documento pertence ou, alternativamente, sua probabilidade de pertencer àquela classe.

O algoritmo de particionamento recursivo é a técnica padrão para a construção de árvores de decisão a partir de exemplos; ele serve de base para sistemas como ID3 (QUINLAN, 1986), C4.5 (QUINLAN, 1993) e CART (STEINBERG & COLLA, 1995). No ID3 e C4.5, a indução das árvores baseia-se na divisão recursiva do conjunto de exemplos de treinamento em subconjuntos mais representativos, utilizando-se a métrica de ganho de informação (*information gain*) (MARON, 1961).

4.1.2 - Regras de Decisão

Outra abordagem simbólica bastante utilizada em tarefas de categorização de documentos é o aprendizado de regras de decisão, devido ao grande poder de expressividade e de legibilidade de regras do tipo Se-Então (MITCHELL, 1997). O principal objetivo de um sistema desta natureza é a obtenção de um conjunto de regras de decisão que classifiquem corretamente os documentos segundo as várias categorias consideradas.

A solução final de um sistema de categorização por aprendizado de regras é expressa sob a forma normal disjuntiva (*disjunctive normal form* - DNF), onde cada categoria é especificada por um conjunto disjunto de regras, compostas de proposições conjuntivas que podem assumir os valores booleanos "falso" ou "verdadeiro". Essas proposições avaliam os valores de alguma característica do documento de entrada, ou verificam o limiar do seu valor corrente.

A Figura 4.2 mostra um subconjunto das regras geradas pelo algoritmo PART (FRANK & WITTEN 1998) para a coleção Metais da Reuters-21578.

Steel > 0 AND
Aluminium <= 0 : iron-steel (38.0/2.0)

Gold > 0 AND
Zinc <= 0 AND
Pound <= 0 AND
Silver <= 0 : gold (67.0/2.0)

Iron <= 0 AND
Gold > 0 AND
Effect <= 0 : gold (52.0/23.0)

Figura 4.2 - Subconjunto das Regras geradas para a categorização.

Se o lado esquerdo da regra (antecedente) resultar verdadeiro, o documento em questão é atribuído à categoria que aparecer do lado direito da regra (conseqüente). Os números que aparecem ao lado dos rótulos das categorias, são respectivamente o número de documentos corretamente categorizados (acertos) e o número de documentos incorretamente categorizados (erros) do conjunto de treinamento.

Diferentemente da árvore de decisão construída para uma coleção, as regras de decisão não são mutuamente exclusivas, de modo que duas ou mais classes podem ser atribuídas simultaneamente a um mesmo exemplo de entrada.

Uma maneira de se efetuar o aprendizado de regras consiste na construção inicial de uma árvore de decisão e na posterior tradução desta para um conjunto equivalente de regras, onde a cada folha da árvore corresponde uma regra correspondente (CORREA,2002). Entretanto, os algoritmos de aprendizado de regras mais utilizados são aqueles que aprendem regras diretamente do conjunto de treinamento, sem a intermediação de uma árvore de decisão. A construção automática de regras se baseia na indução inicial de uma única regra, seguido da retirada daqueles exemplos que satisfaçam a esta regra. Este processo é repetido várias vezes, até que todos os exemplos tenham sido cobertos. Após o processo de indução das regras a partir dos exemplos do conjunto de treinamento, segue-se a fase de ajuste, de modo a eliminar o risco de saturação (*overfitting*), isto é, as regras obtêm um ótimo desempenho para o conjunto de treinamento, mas classificam erroneamente exemplos desconhecidos. Esta estratégia é usualmente denominada de *overfit-and-simplify*, isto é, a geração de um grande conjunto de regras que abranja todos os exemplos (*overfit*) e uma posterior simplificação (*simplify*) deste conjunto.

4.1.3 - Naive Bayes

Existem diversas versões para classificadores de Bayes, e em sua definição mais geral, esta abordagem está baseada em probabilidade condicional (DUDA et.al., 2000). Baseado na probabilidade condicional de palavras específicas aparecerem em um documento, dado que este pertence a uma determinada categoria, esta técnica permite calcular as probabilidades de um novo documento pertencer a cada uma das categorias e atribuir a este às categorias com valores maiores de probabilidade (LEWIS & RINGUETTE, 1994). O classificador também pode informar a probabilidade daquele documento pertencer a uma determinada categoria, dentro de uma estrutura hierárquica (GROBELNIK & MLADENIC, 1998).

A técnica *Naive Bayes* é uma das técnicas de aprendizagem bayesiana mais utilizada para a tarefa de categorização de documentos (MITCHELL, 1997) (McCALLUM & NIGAM, 1998). Aqui, cada um dos exemplos (documentos) do conjunto de treinamento, é descrito por uma série de atributos que indicam a presença ou ausência de termos *<a1, a2, ..., an>*, e a tarefa do classificador será a atribuição da categoria mais provável para cada um dos documentos, por meio de uma função *f* que devolve valores (categorias) pertencentes a um conjunto finito *V*.

O classificador *Naive Bayes* se baseia na suposição simplificada de que os vários atributos dos exemplos de entrada são condicionalmente independentes, dado o valor final da função *f* de saída. Isto é, este classificador considera que a probabilidade de ocorrência de uma conjunção de atributos em um dado exemplo é igual ao produto das probabilidades de ocorrência de cada atributo isoladamente:

$$v_{NB} = \arg\max_{v_j \in V} P(a_1, a_2, ..., a_n / v_j).P(v_j) = \arg\max P(v_j).\prod_i P(a_i / v_j) \qquad (10)$$

onde: v_{NB} é a hipótese (categoria) final atribuída ao documento; v_j é cada um dos possíveis valores (categorias) pertencentes a V; $P(a_1,a_2,...,a_n/v_j)$ é a probabilidade de ocorrência do conjunto de evidências dada a ocorrência da hipótese (categoria); $P(v_j)$ é a probabilidade inicial de ocorrência de cada hipótese (categoria); $P(a_i/v_j)$ é a probabilidade de ocorrência de cada evidência dada a ocorrência de uma hipótese (categoria).

A grande diferença entre o método de *Naive Bayes* e as outras técnicas de aprendizado de máquina é que não existe uma busca explícita no espaço de hipóteses, apenas a definição de uma única hipótese pela contabilização da freqüência das várias combinações de dados do conjunto de treinamento.

Existem diversas versões para esse tipo de classificador, sendo que para a *Naive Bayes* existem dois subtipos, o *multinomial model* e o *multivariate model*.

No *multinomial model* a probabilidade do termo T pertencer à categoria C é calculada através da fórmula:

$$P(T|C) = \frac{\textit{freqüência do termo T nos documentos classificados na categoria C}}{\textit{freqüência do termo T em todos documentos classificados}} \qquad (11)$$

E ainda, dado um documento D, representado por um vetor composto de n termos, $D = (t1,...,tn)$, e uma classe Ci. A probabilidade do documento D pertencer à *i-ésima* categoria Ci, é definida através do produtório das probabilidades dos n termos t desse documento pertencerem a essa categoria C, como mostra a fórmula a seguir:

$$P(D \mid C_i) = \prod_{j=1}^{j=n} P(t_j \mid C_i) \qquad (12)$$

No entanto, há um inconveniente por se tratar de um produtório. Quando a probabilidade de um ou mais termos pertencerem à classe for zero, a probabilidade do documento de pertencer a essa classe também será zero. Para contornar esse problema, a solução proposta é adicionar uma constante maior que zero a cada probabilidade de termo encontrada (JACKSON & MOULINIER, 2002).

Já o *multivariate model* usa um vetor de componentes binários que indicam, para cada termo do vocabulário, se ele acontece ou não no documento. Não é

registrada a freqüência com que o termo aparece em documentos novos, apenas sua presença ou ausência.

4.1.4 - Support Vector Machines (SVM)

O método Máquina de Vetor Suporte ou *Support Vector Machines* (SVM) é uma técnica de aprendizado introduzida por VAPNIK (1995) para resolver problemas de reconhecimento de padrões com duas classes. Ele é baseado no princípio de minimização do risco estrutural. O método é definido sobre um espaço vetorial onde o problema é encontrar uma superfície de decisão que "melhor" separa os dados em duas classes, buscando a maximização da margem existente entre as fronteiras das nuvens de pontos formados por estas classes.

Para a classificação de textos, cada documento é representado por um vetor atributo-valor, em que cada palavra (atributo) corresponde a uma característica.

Basicamente, a idéia de uma máquina de vetor suporte depende de duas operações matemáticas (HAYKIN, 2000):

- O mapeamento não-linear de um vetor de entrada para um espaço de características de alta dimensionalidade, que é oculto da entrada e da saída;
- A construção de um hiperplano ótimo para separar as características descobertas no passo anterior.

4.1.4.1 – Hiperplano ótimo para padrões linearmente separáveis

Quando os dados são linearmente separáveis, pode-se construir um hiperplano que separa os exemplos positivos dos negativos. A função de decisão usada nesses casos é $f(x) = sign((w, x) + b)$ na qual w representa o vetor de pesos da rede, x um vetor de características e b, o bias.

Os pontos x que pertencem ao hiperplano satisfazem a equação $wx+b=0$, onde w é normal ao hiperplano e $|b|/||w||$ é a distância perpendicular do hiperplano até a origem e $||w||$ é a norma euclidiana de w (MORAES, 2004).

Para o caso de dados linearmente separáveis, o algoritmo simplesmente procura pelo hiperplano de separação com a maior margem.

4.1.4.2 – Hiperplano ótimo para padrões não-separáveis

Segundo HAYKIN (2000), quando um determinado dado viola a condição $y_i(x_i.w + b) - 1 \geq 0$, diz-se que a margem de separação entre as classes é suave. Nesse Caso, o dado pode estar dentro da região de separação, no lado correto ou incorreto da superfície de decisão.

4.1.5 - Modelos de Regressão

O *Linear Least Squares Fit* (LLSF), é um modelo de regressão proposto por CHUTE e YANG (1994) cujo processo de treinamento é realizado utilizando um conjunto de documentos e suas respectivas categorias, representados em forma de pares de vetores de entrada e saída. Pela aplicação do LLSF, é obtida a matriz de coeficientes de regressão palavra-categoria. Esta matriz estabelece o mapeamento entre um documento qualquer e os vetores de categorias com pesos. Uma lista ordenada pelos pesos das categorias existentes define a categoria a qual pertence o documento de teste (YANG & LIU, 1999).

4.1.6 - Redes Neurais

Redes Neurais Artificiais são sistemas distribuídos altamente paralelos compostos por simples unidades de processamento que simulam o comportamento de um neurônio biológico, dispostas em uma ou mais camadas. Cada conexão entre dois neurônios possui um peso. Estes pesos guardam o conhecimento de uma rede neural e são usados para definir a influência de cada entrada recebida por um neurônio na sua respectiva saída. Ajustando-se os seus pesos, a rede neural assimila padrões e é capaz de fazer generalizações, isto é, produzir saídas consistentes para entradas não apresentadas anteriormente. As Redes Neurais Artificiais são classificadores por natureza. Uma rede neural básica geralmente possui os seguintes elementos em sua estrutura:

- Uma camada de entrada, composta de várias unidades (neurônios artificiais), de acordo com o número de características utilizadas na representação dos objetos (textos) considerados;
- Uma ou mais camadas intermediárias compostas por alguns neurônios responsáveis pela modelagem de relações não lineares entre as unidades de entrada e saída;
- Uma camada de saída que fornece a resposta do sistema (por exemplo, a probabilidade de uma dada categoria ser atribuída ao documento processado).
- Ligações entre as várias camadas (pesos), responsáveis pela propagação dos sinais entre essas, que são aprendidos durante a fase de treinamento do sistema.

O neurônio é o elemento processador da rede neural. Cada neurônio gera uma saída a partir da combinação de sinais de entrada recebidos de outros neurônios, com os quais está conectado, ou a partir de sinais externos. A saída de um neurônio é, na

maior parte dos modelos, o resultado de uma função de ativação aplicada a soma ponderada de suas entradas.

A topologia de uma rede é descrita por um grafo de nodos (neurônios) e conexões (pesos). Ela é descrita pelo número de camadas da rede, o número de neurônios em cada camada, e o tipo de conexão entre nodos.

A fase de treinamento de redes neurais corresponde a determinação dos pesos das conexões entre os neurônios. Elas possuem a capacidade de aprender por exemplos e fazer interpolações e extrapolações do que aprenderam. No aprendizado conexionista, não se procura obter regras como a abordagem simbólica da Inteligência Artificial, mas sim determinar a intensidade de conexões entre neurônios.

A arquitetura da rede consiste na especificação de sua topologia, a função de ativação utilizada em cada neurônio, o algoritmo de aprendizado, e parâmetros livres associados com a rede.

Existem três diferentes tipos de algoritmos de aprendizagem que podem ser utilizados para o treinamento de uma rede neural:

1. **Aprendizagem supervisionada**: no conjunto de treinamento estão contidos os valores das unidades de entrada e das unidades de saída para cada exemplo. Os pesos da rede devem ser ajustados até que os valores das unidades de saída da rede estejam bem próximos dos valores fornecidos na entrada (ex. MLP - *Multi-layer Perceptron* (BEALE & JACKSON, 1990), RBF - *Radial Basis Function* (MITCHELL, 1997));
2. **Aprendizagem não supervisionada**: no conjunto de treinamento estão contidos apenas os valores das unidades de entrada e a rede deve executar um agrupamento (*clustering*) ou associação, para aprender as classes presentes no conjunto de treinamento (ex. *Selforganizing Maps* (HAYKIN, 2000), Hopfield (BEALE & JACKSON, 1990);
3. **Aprendizagem por reforço**: embora no conjunto de treinamento estejam contidos apenas os valores das unidades de entrada, a cada ciclo do processamento da rede, algumas dicas sobre a saída desejada são fornecidas, ao invés de sua resposta direta.

Nas redes MLP (BEALE & JACKSON, 1990) o sinal de saída de cada neurônio é o resultado da aplicação da função de ativação (também chamada de função de transferência) sobre a soma ponderada dos sinais de entrada. O estado de um neurônio é representado pelo seu sinal de saída. A Figura 4.3 mostra o modelo do neurônio e de uma rede típica.

As funções de ativação mais utilizadas são a sigmóide logística e a tangente hiperbólica. O tipo de função de ativação utilizada influencia na velocidade do treinamento, e na performance da rede.

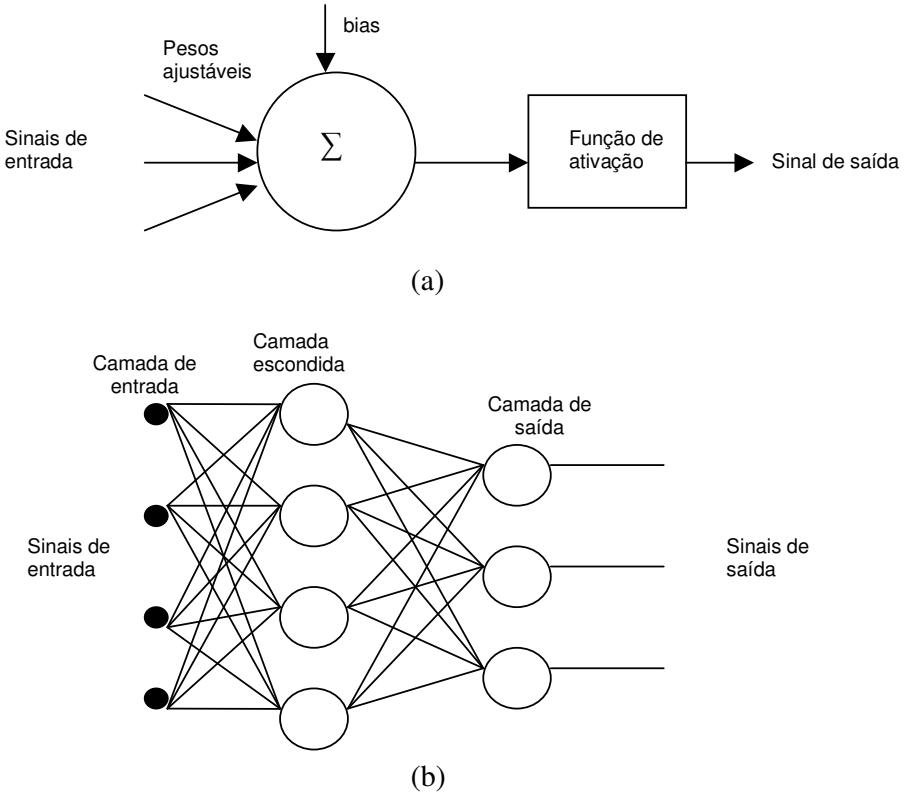

FIGURA 4.3 - Modelo do neurônio (a) e topologia de rede neural (b).

Nas redes MLP, os neurônios estão conectados de forma parcial. O sentido do fluxo de dados ocorre da camada de entrada para a camada de saída, ou seja, alimentação para frente (*feed-forward*). Os neurônios estão divididos em camadas, e só há conexão de um neurônio com neurônios da camada anterior.

As redes neurais MLP podem apresentar uma ou mais camadas de neurônios. As redes de camada única só são capazes de aprender problemas linearmente separáveis (BRAGA et al., 2000). A inserção de camadas escondidas (ou intermediárias) com funções de ativação não lineares permite a modelagem de relações não lineares entre as entradas e saídas (WIENER et al.,1995). As saídas da rede neural são estimadas em termo da probabilidade condicional da pertinência à classe dado o vetor de características do documento.

Uma das técnicas mais utilizadas para diminuir o erro em redes neurais é o algoritmo de *Backpropagation (*MITCHELL, 1997). Este algoritmo apresenta duas fases no seu ciclo de aprendizagem, sendo a primeira responsável pela propagação dos valores de entrada através da rede, e a segunda responsável pelo ajuste da saída por meio da mudança dos pesos da rede.

Este procedimento de aprendizagem é iterativo, com os pesos sendo ajustados à medida que se apresentem os padrões (exemplos) de treinamento. Empregando a técnica de gradiente descendente, o algoritmo do *backpropagation* tenta minimizar a erro quadrático médio entre a saída fornecida pela rede neural e o valor desejado (esperado) para esta saída.

O algoritmo *Backpropagation* possui como parâmetro ajustável a taxa de aprendizado, que indica a intensidade com que os pesos da rede serão ajustados. Valores diferentes podem levar a grandes diferenças tanto no tempo de treinamento quanto na generalização obtida. Geralmente se utiliza taxa de aprendizado pequena.

Uma variação do algoritmo *Backpropagation* adiciona mais um parâmetro ajustável, o *momentum* (RUMELHART & McCLELLAND, 1986), que aumenta a velocidade de aprendizado e reduz o perigo de instabilidade (oscilação dos pesos).

Os critérios de parada mais utilizados no treinamento das redes neurais são:

1. Encerrar o treinamento após certo número de iterações ou épocas;

2. Encerrar o treinamento após o erro quadrático médio ficar abaixo de uma constante;

3. Encerrar o treinamento após a perda de generalização atingir um limiar (PRECHELT, 1994). Este método é conhecido como *early stopping* e necessita de parte do conjunto de treinamento para formar o conjunto de validação. Para medir a generalização da rede durante o treinamento é utilizada validação cruzada (*cross-validation*) com o conjunto de validação, este conjunto de exemplos não será usado no ajuste dos pesos. A configuração final da rede no fim do treinamento é aquela que obteve o menor erro de validação durante o mesmo.

O *overfitting* ocorre quando após certo ciclo do treinamento, a rede, em vez de melhorar, começa a piorar a sua taxa de acertos para padrões diferentes daqueles utilizados para o ajuste dos pesos. Diz-se então que a rede memorizou os padrões de treinamento, gravando suas peculiaridades e ruídos, e perdeu em generalização. As medidas mais utilizadas para reduzir a ocorrência de *overfitting* são: parar o treinamento quando o erro no conjunto de validação começar a crescer e podar os pesos da rede (CORREA, 2002).

A rede MLP possui boa capacidade de generalização, classificando corretamente padrões não-utilizados no treinamento ou com ruído (BRAGA et al., 2000). A generalização ocorre através da detecção de características relevantes nos padrões de entrada. Assim, padrões desconhecidos são atribuídos a classes cujos padrões apresentam características semelhantes. Esta característica motiva a utilização destas redes na categorização de documentos.

4.1.7 – Espaço vetorial e Similaridade por cosseno

O modelo de indexação por espaço vetorial para RI considera o documento representado por um vetor de termos e seus respectivos pesos associados. O vetor possui a forma {(termo1, peso1),(termo2, peso2),...,(termo n, peso n)}, onde n representa a n-ésima dupla termo-peso representante do documento. Calcula a similaridade ou diferença entre dois vetores através da medida de cosseno entre eles. LOSEE (1998) diz que o ângulo entre dois vetores pode ser apresentado em um espaço euclidiano. Esta técnica pode ser utilizada para categorização de textos por serem eles representados por vetores. Através do cálculo do cosseno do ângulo entre os vetores é possível definir a similaridade entre eles. SALTON e BUCKLEY (1987) propuseram a seguinte fórmula para cálculo de similaridade entre vetores normalizados:

$$Similaridade(Q,D) = \frac{\sum_{k=1}^{n} w_{qk} \times w_{dk}}{\sqrt{\sum_{k=1}^{n} (w_{qk})^2 \times \sum_{k=1}^{n} (w_{dk})^2}} \qquad (13)$$

onde: Q é o vetor de termos da consulta (vetor de termos do texto a ser categorizado), D é o vetor de termos do documento (vetor de termos da categoria), w_{qk} são os pesos dos termos da consulta, ou seja, do texto analisado, w_{dk} são os pesos dos termos do documento, ou seja, da categoria.

4.1.8 – Algoritmo de Widrow-Hoff

Em MOENS (2000), encontra-se a descrição do algoritmo de Widrow-Hoff. Este algoritmo utiliza um vetor de pesos que é atualizado a cada novo exemplo a partir do vetor antigo. Inicialmente, um vetor de pesos é definido aleatoriamente. A cada passo, um novo vetor de pesos w_{i+1} é calculado a partir do vetor de pesos antigos w_i, utilizando um exemplo de treinamento x_i.

O componente j-ésimo do vetor de pesos w_{i+1} é calculado a partir da seguinte fórmula apresentada em MOENS(2000):

$$w_{i+1,,j} = w_{i,j} - 2 \times \eta \times (w_i \times x_i \times - y_i) \times x_{i,j} \qquad (14)$$

onde: w_i é o vetor de peso antigo; $w_{i,j}$ é o valor do j-ésimo componente do vetor wi; η é a taxa de aprendizado que controla quanto o vetor de peso w_i mudará e quanta influência cada exemplo novo deve ter ($\eta > 0$); x_i é o vetor treinamento; y_i é um valor que indica se a classe C_k é classificada ao exemplo de treinamento x_i (normalmente 1 ou 0); x_{ij} é o valor do j-ésimo componente do vetor x_i.

4.1.9 – Modelos Difusos

A categorização de documentos trata da representação de textos através de termos e, por isso, enfrenta situações ambíguas para definir a relevância dos termos no que se refere à representação de uma categoria.

O uso da lógica difusa proposta por Lotif A. Zadeh, na categorização de textos, vem ao encontro da solução do problema da ambigüidade, pois a mesma se propõe a tratar situações imprecisas, oferecendo melhores resultados através do cálculo da pertinência de um elemento a um conjunto.

Através desta técnica, é possível definir o quanto um termo é importante, alem de definir se ele é relevante ou irrelevante para uma categoria.

Em CROSS apud RIZZI(1999), é apresentada a formalização da extensão de um modelo probabilístico para um conjunto difuso, no qual os operadores *booleanos* são ampliados para tratar a incerteza.

Os conjuntos que representam os documentos são compostos pelas duplas {termo, peso}, sendo o peso um valor difuso definido entre zero e um. Este valor indica a importância do termo, quanto mais próximo do valor um, mais relevante é o termo.

A partir da atribuição da relevância dos termos em relação ao documento, os sistemas difusos baseiam-se na idéia de similaridade, permitindo que os resultados ofereçam não apenas classificações exatas de um documento com relação a uma classe, mas também classificações parciais, sendo atribuído a cada classe um grau de pertinência ou de relevância com relação ao documento analisado.

4.1.10 – Algoritmo de Rocchio

O algoritmo de ROCCHIO (1971) foi inicialmente definido para o processo de relevância e *feedback* em RI, mas também é aplicado à categorização de textos. Ele primeiramente constrói um vetor, chamado de vetor protótipo, para representar cada categoria; no caso da categorização de textos, um vetor de conceitos da categoria. Para efetuar a categorização é feito um cálculo de similaridade ou de distância entre o vetor de conceitos do texto e o vetor protótipo da categoria. Considerando o grau de similaridade dentro de um limiar pré-estabelecido, o texto será classificado ou não na categoria testada.

A construção do vetor protótipo da categoria se dá por um conjunto de textos para treinamento. São atribuídos pesos positivos aos vetores pertencentes à categoria e pesos negativos aos vetores não pertencentes. A partir da média ponderada entre esses pesos é definido o vetor protótipo da categoria. A expressão para efetuar esse cálculo é apresentada a seguir:

$$\beta \times \frac{1}{\eta_{relCk}} \times \sum_{relCk} crença - \gamma \times \frac{1}{\eta_{nãorelCk}} \times \sum_{nãorelCk} crença \qquad (15)$$

onde: crença é o peso do termo; η_{relCk} é a quantidade de textos exemplo relevantes para a classe C_k; $\eta_{nãorelCk}$ é a quantidade de textos exemplo irrelevantes para a classe C_k; β é o peso negativo dos textos relevantes para a classe C_k; γ é o peso negativo dos textos irrelevantes para a classe C_k.

4.1.11 – Similaridade Difusa

A similaridade tem por objetivo definir o quão semelhantes são dois vetores representativos. Em WIVES (1999), é apresentada uma forma de cálculo de similaridade baseada em funções difusas.

Baseado nas idéias de OLIVEIRA (1996), LOH (2001) define uma fórmula para calcular similaridade que leva em conta as diferenças e as semelhanças entre os documentos, utilizando operadores difusos adequados as situações. Abaixo é dada a fórmula:

$$gs(X,Y) = \frac{\sum_{h=1}^{k} gi_h(a,b)}{N} \qquad (16)$$

onde: gs é o grau de similaridade entre documentos X e Y; gi é o grau de igualdade entre pesos do termo h (peso *a* no documento X e peso b no documento Y); h é um índice para os termos comuns aos dois documentos; k é o número total de termos comuns aos dois documentos; N é o número total de termos nos dois documentos sem contagem repetida.

A partir da aplicação desta fórmula, cada vez que um termo é encontrado em ambos os documentos, um valor é acumulado. Esse valor vai definir o grau de similaridade entre os textos. O valor que deve ser acumulado é dado pelo grau de igualdade entre os pesos. Este valor é calculado pela seguinte fórmula, apresentada por PEDRYCZ apud WIVES (1999):

$$gi(a,b) = 0{,}5 \times [(a \rightarrow b) \wedge (b \rightarrow a) + (\overline{a} \rightarrow \overline{b}) \wedge (\overline{b} \rightarrow \overline{a})] \qquad (17)$$

A utilização do grau de igualdade é necessária, pois, mesmo que os termos sejam iguais, eles podem ter pesos diferentes entre os documentos analisados. Estes pesos podem ter sido calculados pelas fórmulas de freqüência de termo. Então, quando um termo aparece em ambos os documentos com pesos muito diferentes, a igualdade diminui, com pesos semelhantes, a igualdade aumenta.

O resultado deste processo será um valor entre zero e um, como todo resultado difuso. Quanto mais próximo de zero, menos similares serão os documentos e quanto mais próximos a um, mais similares.

A fase de definição de categorias é o final do processo de categorização. Para análise de textos novos, após a preparação do texto e seleção das características, é efetuada a comparação de seu índice com o índice das categorias pré-existentes, através de um dos métodos descritos, e assim a categorização está completa.

4.1.12 – Neurofuzzy

Um sistema *neurofuzzy* é um sistema que usa um algoritmo de aprendizado derivado ou inspirado em teoria de redes neurais para determinar seus parâmetros, conjuntos e regras *fuzzy*, através do processamento de exemplos de dados.

Sistemas *neurofuzzy* modernos são normalmente representados como redes neurais multicamadas. Contudo, *fuzzificações* de arquiteturas de redes neurais são também consideradas.

Em redes *neurofuzzy* os pesos da conexão e a propagação e funções de ativação diferem de redes neurais comuns. Todavia, existem várias abordagens diferentes. O termo sistema *neurofuzzy* é usado para abordagens que apresentam as seguintes propriedades (JANG & SUN, 1995):

- um sistema *neurofuzzy* está baseado em um sistema *fuzzy* que é treinado por um algoritmo de aprendizagem derivado de uma teoria de redes neurais. O procedimento de aprendizagem (heurística) opera em informações locais, e causa apenas modificações locais no fundamento de sistema *fuzzy*;
- um sistema *neurofuzzy* pode ser visto como uma rede neural de 3 camadas. A primeira camada representa as variáveis de entrada, a camada do meio (escondida) representa as regras *fuzzy* e a terceira camada representa as variáveis de saída. Conjuntos *fuzzy* são codificados como pesos de conexões. Não é necessário representar um sistema *fuzzy* como este para aplicar o algoritmo de aprendizagem. De qualquer forma, pode ser conveniente, pois representa o fluxo de dados do processamento de entrada e aprendizagem no modelo. Algumas vezes uma arquitetura de 5 camadas é usada, onde os conjuntos *fuzzy* são representados em unidades da segunda e quarta camadas;

- um sistema *neurofuzzy* pode ser sempre interpretado como um sistema de regras *fuzzy*. Nem todos os modelos *neurofuzzy* especificam procedimentos de aprendizagem para criação de regras *fuzzy*;
- o procedimento de aprendizagem de um sistema *neurofuzzy* torna as propriedades semânticas de um sistema *fuzzy* em descrição. Isto resulta em restrições em possíveis modificações aplicáveis aos parâmetros do sistema. Nem todas as abordagens *neurofuzzy* possuem esta propriedade;
- um sistema *neurofuzzy* aproxima uma função n-dimensional que é parcialmente definida pelo treinamento dos dados. As regras *fuzzy* codificadas no sistema representam exemplos vagos, e podem ser vistas como protótipos de treinamento de dados. Um sistema *neurofuzzy* não deve ser visto como um tipo de sistema *expert*, e não tem nada a fazer com a lógica *fuzzy* no sentido limitado.

Sistemas *neurofuzzy* combinam os atributos positivos destas abordagens produzindo sistemas *fuzzy* com habilidade de aprender e se adaptar ao mundo real. Estes sistemas são ideais para aplicações como:

- modelagem e controle de processos dinâmicos em tempo real;
- classificação; e
- vigilância de condição de sensor.

Um sistema *neurofuzzy* consiste de componentes convencionais de um sistema *fuzzy*, exceto que a computação de cada estágio é realizada por uma camada de neurônios ocultos e a capacidade de aprendizagem da rede neural possibilita o melhoramento do sistema de conhecimento. Diferentes arquiteturas de sistemas *neurofuzzy* estão disponíveis.

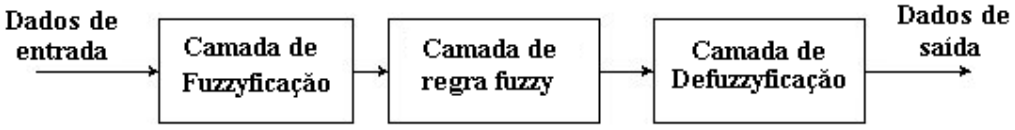

Figura 4.4: Diagrama Esquemático de um Sistema Neurofuzzy.

Uma possível arquitetura de um sistema híbrido *neurofuzzy* é mostrada na Figura 4.4. O sistema contém três diferentes camadas, sendo elas:

- Camada de *Fuzzificação*;

- Camada de Inferência (Regras *Fuzzy*); e
- Camada de *Defuzzificação*.

Na Camada de *Fuzzificação* cada neurônio representa uma função *membership* de entrada do antecedente de uma regra *fuzzy* (Figura 4.5). Na Camada de Inferência as regras são ativadas e os valores ao final de cada regra representam o peso inicial da regra, e serão ajustados ao seu próprio nível ao final do treinamento. Na Camada de *Defuzzificação* cada neurônio representa uma proposição conseqüente e suas funções *membership* podem ser implementadas combinando uma ou duas funções sigmóides e lineares. O peso de cada saída representa aqui o centro de gravidade de cada saída da função *membership*. Após adquirir a saída correspondente o ajuste é feito na conexão do peso e nas funções *membership* visando compensar o erro e produzir um novo controle de sinal.

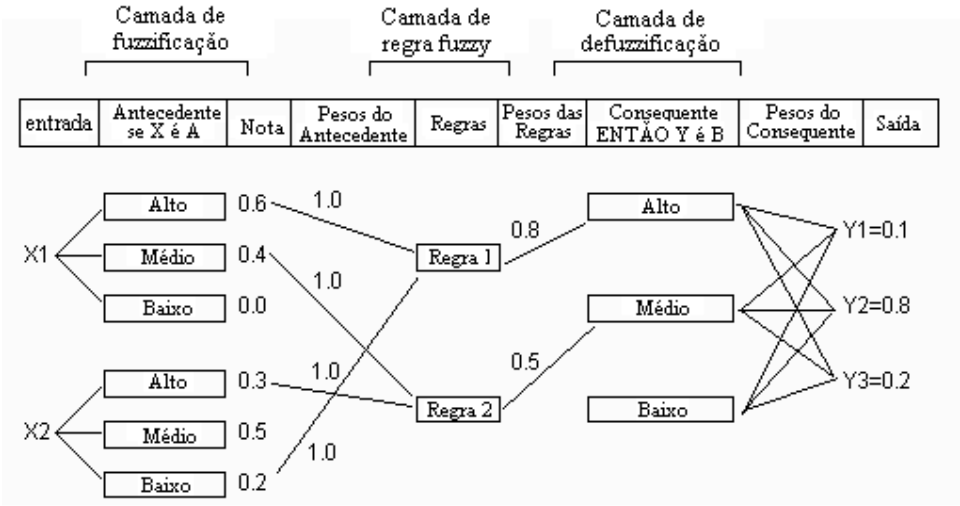

Figura 4.5. Estrutura de um Sistema Neurofuzzy**.**

4.2 – Métodos de categorização não-supervisionada

Embora esta dissertação trate de categorização supervisionada de textos, serão listados apenas alguns exemplos de métodos de categorização não-supervisionada, ou seja, a classificação é feita sem o conhecimento prévio da classe de cada uma das amostras. Desta forma, é realizado um agrupamento (*clustering*) das amostras em torno de pontos médios em torno dos quais se agrupam em maior grau as amostras. Após este processo, serão rotulados os agrupamentos de acordo com as características de cada um deles. Este tipo de categorização é realizada para grandes

conjuntos de dados de modo a organizá-los e facilitar o tratamento de cada um deles separadamente.

4.2.1 - K Vizinhos mais Próximos

Este método considera que, dado um documento de entrada qualquer, são ordenados seus *k* vizinhos mais próximos (*k-nearest neighbour*) entre os documentos de treinamento, e são usadas as categorias dos vizinhos melhores colocados (mais similares) para atribuir a(s) categoria(s) ao documento de entrada (YANG & LIU, 1999). De forma mais específica, o nível de similaridade de cada documento vizinho ao documento a ser classificado, é usado como peso das categorias as quais pertencem. Se alguns dos *k* vizinhos mais próximos são de uma mesma categoria, então é atribuído a esta categoria o somatório dos respectivos pesos. Esta soma resultante é usada como pontuação da possibilidade do documento de teste pertencer àquela categoria. Uma lista ordenada é elaborada pela classificação dos escores dos pesos das possíveis categorias as quais o documento pode pertencer. Por esta lista é feita a categorização do documento.

Diferente de outros tipos de aprendizagem, por este método não há uma construção de uma descrição geral e explícita de uma situação (modelo). Os documentos pré-classificados (ou instâncias) são armazenados em memória e a classificação de um novo exemplo é baseada na classificação dos exemplos armazenados mais próximos (ou similares). O processo de aprendizagem consiste unicamente no armazenamento de novas instâncias e o processamento é efetuado apenas após a apresentação de um exemplo não classificado (CORREA, 2002).

4.2.2 - Redes Self-Organizing Maps (SOM)

A rede SOM (KOHONEN, 1997) é um modelo de aprendizado não supervisionado que produz um gráfico de similaridade dos dados de entrada. Ele aproxima um número ilimitado de itens de dados (padrões) por um conjunto finito de vetores modelos. Estes vetores modelos se encontram associados a unidades (neurônios), arranjadas regularmente em uma grid bidimensional denominada mapa, assim possuindo uma única camada. Tais vetores modelos são vetores de características, da mesma dimensão dos padrões de treinamento, que são gerados por um processo de aprendizado que automaticamente os ordena na grid de acordo com as suas similaridades mútuas. A SOM pode ser primeiramente computada usando um conjunto representativo de dados de entrada antigos e novos itens de entrada podem ser mapeados em modelos mais similares sem recomputação do mapeamento inteiro.

Do ponto de vista estatístico, a SOM pode ser interpretada como uma projeção não-linear da função densidade de probabilidade de dados de entrada de alta dimensionalidade em um gráfico bidimensional.

Durante o treinamento, a SOM forma uma rede elástica de vetores modelos (ou unidades) que se molda na nuvem formada pelos dados de entrada. Pontos do espaço de entrada próximos uns dos outros são mapeados em unidades próximas no mapa. Assim, o mapa SOM pode ser interpretado como um mapeamento que preserva a topologia do espaço de entrada em uma grid 2-D de unidades. Em nenhum momento a informação sobre a que categoria pertence um documento é utilizado no treinamento das redes SOM.

A saída das unidades quando um padrão (sinais de entrada) é apresentado ao mapa, é a distância entre o vetor de entrada que o representa e o vetor modelo associado a cada uma (CORREA, 2002).

A SOM é treinada originalmente de forma iterativa. Em cada passo do treinamento, um vetor x é randomicamente escolhido do conjunto de dados de entrada. As distâncias Euclidianas entre x e todos os vetores modelo são computadas. A unidade do mapa cujo vetor modelo melhor aproxima o vetor entrada é denominada unidade vencedora. Então, os vetores modelos da unidade vencedora e as unidades topologicamente na vizinhança são atualizados, movidos para a região do espaço próximo do vetor entrada.

O algoritmo SOM é aplicável a grande conjunto de dados. A complexidade computacional é linearmente proporcional ao número de vetores de entrada. Por outro lado, a complexidade é quadraticamente proporcional ao número de unidades no mapa. A memória necessária durante o treinamento basicamente é gasta no armazenamento dos vetores modelos e vetores de entrada.

O número de unidades no mapa é que determina a acurácia e capacidade de generalização dos mapas auto-organizáveis. Durante o treinamento, a SOM forma uma rede elástica que se molda na nuvem formada pelos dados de entrada. Pontos do espaço de entrada próximos uns dos outros são mapeados em unidades próximas no mapa. Assim, o mapa SOM pode ser interpretado como um mapeamento que preserva a topologia do espaço de entrada em uma grid 2-D de unidades (CORREA, 2002).

A capacidade da SOM em organizar os padrões de entrada de forma que padrões similares encontram-se mapeados em unidades próximas, a torna muito útil na organização de grandes coleções de dados em geral, incluindo coleções de documentos.

Baseando-se na hipótese de que características textuais elementares dos documentos que abordam tópicos similares são estatisticamente similares, a SOM já tem sido usada para a tarefa de classificação de documentos.

Após o treinamento a SOM passa por um processo de rotulação onde é atribuída a cada unidade a categoria da maioria dos documentos do conjunto de treinamento nela mapeados.

4.3 – *Ensembles* de Classificadores

Uma das áreas em aprendizado de máquina supervisionado estuda métodos de construção de *ensembles* de classificadores. Um *ensemble* é um conjunto de classificadores cujas decisões individuais são combinadas de alguma forma para classificar um novo caso. Um resultado interessante é que *ensembles* de classificadores podem ser mais precisos que os classificadores individuais que compõem o *ensemble* (BERNARDINI, 2002).

Uma condição para que um *ensemble* de classificadores seja mais preciso que seus componentes é que os classificadores que compõem o *ensemble* sejam distintos (HANSEN & SALAMON, 1990). Um classificador preciso é um classificador que faz a predição da classe de um novo exemplo x com uma margem de erro menor do que simplesmente adivinhar o valor de y dado x. Dois classificadores são distintos se cometem erros diferentes em novos conjuntos de exemplos.

Para melhor compreender essa condição, considera-se um *ensemble* de três classificadores h_1, h_2 e h_3 e um novo exemplo x. Esse exemplo será classificado por cada classificador com uma das classes do conjunto discreto de classes $\{C_1,...,C_{NCl}\}$.

Seja $h_1(x)$ a classificação atribuída a esse novo exemplo x pelo classificador h_1, $h_2(x)$ pelo classificador h_2 e $h_3(x)$ pelo classificador h_3. Se os três são idênticos então quando $h_1(x)$ está errado, $h_2(x)$ e $h_3(x)$ também estão. Entretanto, se os erros cometidos pelos classificadores forem não correlacionados, então quando $h_1(x)$ está errado, $h_2(x)$ e $h_3(x)$ podem estar corretos, de forma que o voto majoritário pode classificar corretamente o exemplo x. Em geral, dado um *ensemble* composto por L classificadores $h_1,...,h_L$, para cada novo exemplo x a ser classificado por esses classificadores, tem-se uma série de L ensaios. Considerando que cada um desses ensaios é independente e que cada ensaio é um sucesso na classificação de x com probabilidade p ou uma falha com probabilidade $1 - p$, então a probabilidade do número de sucessos ser l em L ensaios é dada por

$$P(Z = l) = \binom{L}{l} p^l (1-p)^{L-l} \quad (18)$$

Mais precisamente, se as taxas de erro de L classificadores $h_1,...,h_L$ são todas iguais a $p < 0,5$ e se os erros são independentes, então a probabilidade do voto majoritário estar errado, ou seja, a probabilidade do *ensemble* ter mais de 50% dos classificadores que falharam é dada por

$$P(Z > \frac{L}{2}) = 1 - \sum_{l=0}^{\frac{L}{2}} \binom{L}{l} p^l (1-p)^{L-l} \quad (19)$$

que corresponde a área sob o gráfico da distribuição binomial na qual mais que $L/2$ hipóteses estão erradas.

Em BERNARDINI (2002) pode ser observado que quanto maior o valor da taxa de erro das hipóteses, maior é a taxa de erro do *ensemble*, mas ainda assim a taxa do *ensemble* é menor que a taxa de erro de cada hipótese que o compõe. Observa-se também que quanto maior o número de hipóteses, menor é a taxa de erro do *ensemble*. A chave para o sucesso dos métodos de criação de *ensembles* está em construir classificadores individuais com taxas de erro abaixo de 0,5.

4.3.1 – Métodos de combinação de classificadores

Dado um conjunto (*ensemble*) de classificadores, muitos métodos têm sido explorados para se combinar as decisões individuais de cada um deles. Para problemas de classificação, esses métodos podem ser divididos em Votação sem peso, Votação com peso e *Stacking*.

A votação sem peso é o método mais simples de combinação de classificadores. Pode parecer que esquemas de votação mais inteligentes poderiam ser melhores, mas a experiência mostra que a votação sem peso é bastante robusta (BERNARDINI, 2002).

Na votação com peso são atribuídos pesos proporcionais para cada classificador medindo a precisão de cada um deles no conjunto de treinamento ou de teste.

No *stacking* combinam-se diferentes hipóteses induzidas por algoritmos de aprendizado de máquina. A meta deste método é encontrar uma boa combinação deste conjunto de hipóteses.

4.4 – Revisão Bibliográfica

A indução de árvores de decisão tem sido usada na tarefa de categorização de textos com bons resultados. Exemplos de aplicações são LEWIS & RINGUETTE (1994) e JOACHIMS (1998) para os documentos da empresa de notícias *Reuters* (os dois primeiros utilizaram a coleção Reuters-22173, e o último, a coleção *Reuters*-21578).

Em JOACHIMS (1998) foi utilizado o algoritmo C4.5 na construção da árvore de decisão, e o desempenho obtido foi comparado com os obtidos por outros classificadores. Em YANG (1999) compara-se o desempenho da aplicação de árvores de decisão em atividades de categorização de documentos nas várias versões da coleção *Reuters*.

Como exemplo de sistemas de regras de decisão, pode-se citar: o SWAP-1 aplicado na categorização de documentos da coleção *Reuters*-22173 em (APTÉ et al., 1994a) e (APTÉ et al., 1994b). Em (APTÉ et al., 1994a) foram criadas regras de indução para representar categorias, a partir de exemplos de treinamento. O classificador gerado pelo algoritmo SWAP-1 obteve desempenho superior aos obtidos em experimentos anteriores utilizando árvore de decisão (LEWIS & RINGUETTE, 1994) e probabilidade Bayesiana (LEWIS, 1992). Em APTÉ et al. (1994b) o Swap-1 é usado para construir um classificador para coleção *Reuters* em alemão que possui organização hierárquica.

A técnica *Naive Bayes* foi utilizada satisfatoriamente na construção do sistema CORA (McCALLUM et al., 1999) para a categorização de artigos científicos e nos experimentos de GROBELNIK & MLADENIC (1998) para a categorização de páginas Web na hierarquia de categoria do Yahoo. O sistema Cora coleta na Internet documentos nos formatos *postscript* (ps) e *portable document format* (pdf) e os insere numa hierarquia de tópicos da área da Ciência da Computação.

Resultados da avaliação de classificadores *Naive Bayes* para a coleção Reuters podem ser encontrados em LEWIS & RINGUETTE (1994), McCALLUM & NIGAM (1998) e YANG & LIU (1999); e para documentos da área médica (LARKEY & CROFT, 1996).

Existem várias versões do classificador Naive Bayes. Estudos mais recentes (McCALLUM & NIGAM, 1998) (YANG & LIU, 1999), têm indicado que o modelo *multinominal mixture* obtém melhor performance na categorização de documentos que algumas versões geralmente utilizadas de Naive Bayes, entre elas o modelo *multivariate Bernoulli*. SCHNEIDER (2003) realizou um estudo de comparação entre as duas variações do modelo Naive Bayes na classificação de *spam* e uma de suas conclusões é que o *multinomial model* pode alcançar uma maior precisão que o

multivariate model. Os testes foram feitos com duas coleções de mensagens: a Ling-Spam e a PUI, composta de 618 mensagens legítimas e 481 mensagens *spam*, ambas em língua inglesa.

Em COOPER et al. (1994) parâmetros obtidos na aplicação de regressão logística são utilizados em uma metodologia para recuperação de textos. No trabalho de YANG & LIU (1999) este método é utilizado para categorizar documentos da coleção Reuters-21578.

Uma das aplicações de redes neurais mais reconhecidas para a tarefa de categorização de texto está descrita no trabalho de WIENER et al. (1995) e WIENER (1995) para a coleção Reuters- 22173.

Entre os trabalhos realizados na utilização de redes MLP em classificação se destacam os trabalhos de WIENER et al. (1995) e NORMAN et al. (1996) sobre a coleção Reuters-22173, e RIZZI et al. (2001), YANG & LIU (1999) e CORREA (2002) sobre a coleção Reuters-21578.

Em SILVA (2004) foi realizada a comparação entre o pré-processamento usual e o pré-processamento baseado em informações lingüísticas, foram realizadas classificações através de redes neurais e agrupamento. Em GALHO (2004) foi realizada a classificação de textos em português através de lógica difusa, neste experimento foi utilizado o método de escore de relevância para seleção dos termos de cada categoria.

MOENS (2000) apresenta uma versão do algoritmo de Widrow-Hoff com *backpropagation* que é utilizada para treinamento de redes neurais. Ele foi utilizado para categorização de textos médicos e obteve resultados de até 72% de aproveitamento em exemplos novos (LEWIS apud MOENS, 2000).

O método SVM foi utilizado por JOACHIMS (1998) e YANG & LIU (1999) na coleção Reuters-21578, e obteve um desempenho superior aos encontrados por outros métodos para esta coleção.

Em MOENS (2000), encontra-se a informação de que, nos experimentos de Lewis com o algoritmo de Rocchio, os resultados obtidos com a categorização de textos médicos foram inferiores a 50%. Mas afirma que este algoritmo é uma boa escolha para treinamento em coleções em que se tenham bons exemplos de treinamento.

DRUCKER, WU & VAPNIK (1999) compararam o método SVM com o algoritmo de Rocchio, Ripper e árvores de decisão na classificação de *spam* e concluíram que o SVM obteve melhor desempenho, inclusive com menor tempo de treinamento.

Em BERNARDINI (2002) foram combinados classificadores simbólicos para geração de regras de decisão para um único classificador, no entanto, não foram obtidos bons resultados.

Em KALVA (2005) foram combinados classificadores neurais e bayesianos para classificação de imagens com informação contextual e os resultados obtidos foram melhores que os obtidos sem a combinação.

Em LARKEY & CROFT (1996) foram combinados classificadores bayesianos, K vizinhos mais próximos e feedback de relevância com resultados melhores que os obtidos individualmente.

Em CIMIANO et. al. (2005) foram utilizadas as duplas formadas por sujeito+verbo e verbo+complementos para o agrupamento de textos e geração de hierarquias entre conceitos.

Em PEREZ & VIEIRA (2005) foram gerados mapas conceituais a partir das estruturas sujeito+verbo+complemento com resultados pouco satisfatórios.

4.5 – Considerações

Para a realização da classificação de textos diversos algoritmos já foram usados e alguns deles se destacam pelos resultados obtidos. Para a realização deste trabalho foram escolhidos dois algoritmos que apresentaram excelentes resultados em textos em inglês. São eles os classificadores *Naive Bayes Multinomial model* e a Máquina de Vetor Suporte (*Support Vector Machines*). Tais algoritmos ainda não foram testados com a seleção de palavras por suas funções sintáticas e espera-se a repetição do bom desempenho já obtido para a seleção por pré-processamento usual.

No próximo capítulo serão enunciados os resultados e a comprovação da performance esperada.

Capítulo 5 – Experimentos

Neste capítulo serão descritos os materiais utilizados e os experimentos realizados. Neste trabalho buscou-se a utilização de coleções já citadas em outros trabalhos de modo a compararem-se os resultados e acrescentou-se uma nova coleção de textos científicos. Os programas de classificação também são os mesmos utilizados em alguns trabalhos anteriores de modo a garantir-se a comparação.

5.1 - Materiais utilizados

5.1.1 – Corpus

Corpus é o nome que se dá às coleções de textos a serem usadas para classificação. São dois conjuntos, sendo o primeiro denominado Corpus Jornal formado por um conjunto de textos extraídos do Jornal Folha de São Paulo, do ano de 1994, elaborado pelo NILC – Núcleo Interinstitucional de Lingüística Computacional. São 855 textos classificados em cinco categorias: esporte, imóveis, informática, política e turismo, sendo 171 arquivos por classe. E o outro denominado Corpus Teses formado por um conjunto de textos gerados pela junção do título e resumo das teses de pós-graduação (mestrado e doutorado) da área de Engenharia Elétrica da Universidade Federal do Rio de Janeiro – COPPE-UFRJ. São 475 textos classificados nas categorias controle, microeletrônica, processamento de sinais, redes de computadores e sistemas de potência, sendo 95 arquivos por classe.

5.1.2 – *Software*

Para processamento dos textos utilizou-se da ferramenta WVTOOL – *Word Vector Tool*, desenvolvida por Michael Wurst na linguagem Java, da Universidade de Dortmund, que realiza a criação de listas de palavras e, através destas listas, a criação dos vetores numéricos com base nas palavras determinadas. Alterações foram feitas para cálculo dos escores de relevância e coeficiente de correlação, e geração dos arquivos weka. Esta ferramenta foi desenvolvida em módulos independentes (ou classes) possibilitando a inclusão de ou em qualquer outro programa em Java e inclui classes para realização de *stemming* em diversos idiomas, entre eles o português, e classes para tratamento dos arquivos de entrada.

Para a extração dos termos e geração das duplas sintáticas foram desenvolvidos programas na linguagem Java para aplicarem-se as folhas de estilo XSL e obterem-se as duplas no formato sujeito+verbo e verbo e seus complementos,

tais como: verbo+objeto direto, verbo+objeto indireto, verbo+agente da passiva e, verbo+predicativo do sujeito. Para geração da combinação entre termos e duplas sintáticas, após terem sido extraídos tanto os termos como as duplas, foi desenvolvido um programa em Java.

A ferramenta utilizada na realização dos experimentos é denominada WEKA (*Waikato Environment for knowledge Analysis*) (WITTEN, 2000). É formada por uma coleção de algoritmos de Aprendizado de Máquina para resolução de problemas reais de mineração de dados (MD). Estes algoritmos são implementados na linguagem Java, apresentando como características a portabilidade, podendo ser executado em qualquer plataforma e aproveitar os benefícios de uma linguagem orientada a objetos e ser de domínio público. Os algoritmos implementados na ferramenta suportam métodos de aprendizado supervisionado e não-supervisionado, e ferramentas de visualização de dados que permitem plotar os resultados obtidos nos experimentos.

As execuções dos algoritmos podem ser realizadas por linha de comando ou módulo gráfico. A figura 5.1 mostra o módulo gráfico (entrada dos dados).

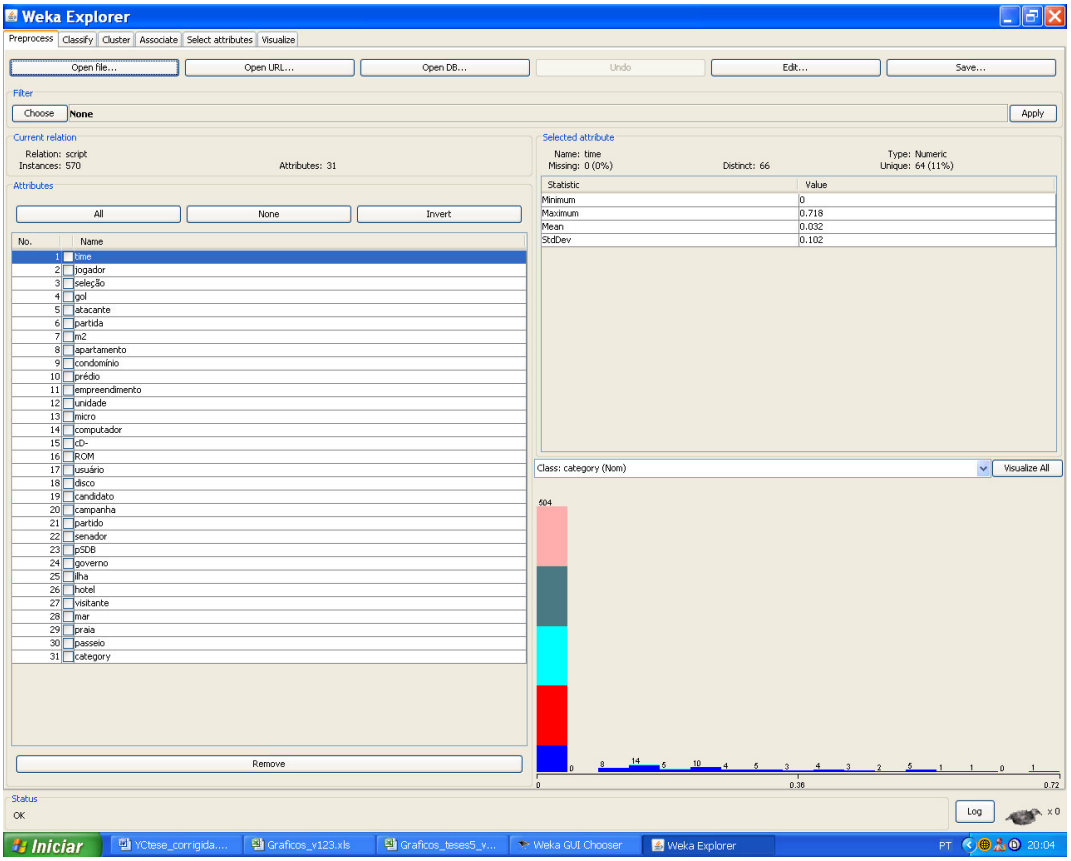

Figura 5.1 – Tela do Programa WEKA.

Para que os dados possam ser aplicados à ferramenta é necessário que sejam convertidos ao formato de entrada do *WEKA*. O *WEKA* apresenta um formato de entrada de dados próprio, denominado ARFF. Esse formato consiste basicamente de duas partes: uma lista de atributos e uma lista de exemplos.

O resultado é dado em arquivo texto que pode ser salvo e onde encontramos informações como medidas de desempenho, matriz de confusão e outras informações complementares.

5.2 – Resultados e Análise dos experimentos

Para realização dos experimentos foram adotados os classificadores Naive Bayes e *Support Vector Machines* – SVM devido aos excelentes resultados obtidos na classificação de textos listados na bibliografia, ambos implementados pela ferramenta *weka*. Abaixo veremos as performances dos dois classificadores para cada uma das técnicas de seleção de termos adotadas neste trabalho.

Neste trabalho adotou-se a redução de características do tipo local, i.e., foi escolhido o conjunto mais relevante de termos representativos para cada categoria. Esta redução determina quantos termos comporão os vetores locais de cada categoria, e consequentemente, quantos termos comporão o vetor global, que consiste na soma dos vetores locais e será o vetor de índice para a criação dos vetores representativos de cada texto.

Para avaliação dos métodos de seleção de atributos foram criados, através da truncagem (seleção dos n termos mais relevantes de acordo com o cálculo de relevância considerado), vetores locais com 6, 12, 18, 24 e 30 termos totalizando vetores globais com 30, 60, 90, 120 e 150 termos, assim como em SILVA (2004).

Para realização dos experimentos com termos foram adotadas as mesmas combinações de termos de SILVA (2004) com o intuito de comparar os resultados. Foram elas: substantivo, substantivo+adjetivo, substantivo+nome próprio, substantivo+verbo, substantivo+verbo+adjetivo, substantivo+nome próprio+adjetivo, nome próprio+adjetivo e verbo.

Nos experimentos envolvendo as duplas sintáticas foram considerados os pares sujeito+verbo e verbo+complementos. A idéia é trazer mais conteúdo semântico aos termos que representam cada documento. Desta forma, se tomarmos como exemplo a palavra bola e considerarmos cada ação que pode ser vinculada a ela, obteremos informações complementares, tais como comprar+bola, jogar+bola, arremessar+bola e bola+furar que representam cada qual uma idéia mais elaborada que a palavra isolada.

Para eliminar possíveis discrepâncias entre os resultados resultantes da formação dos conjuntos de treinamento e teste foi adotada a validação cruzada que consiste no particionamento do conjunto de exemplos em quantas partes se desejar, sendo que uma das partes é considerada de teste, e as demais como treinamento. Neste trabalho foram adotados três conjuntos, V1, V2 e V3, sendo que V1 corresponde a considerarmos o terceiro conjunto para teste e os demais para treino, V2 corresponde a considerarmos o segundo conjunto para teste e os demais para treino e V3, o primeiro conjunto para teste e os demais para treino.

Os valores apresentados correspondem aos valores médios dos erros obtidos em cada particionamento do conjunto de dados.

5.2.1 – Resultados para categorização baseada em termos

Nesta seção são apresentados os resultados obtidos no processo de categorização dos documentos dos dois corpus para as técnicas de seleção de atributos frequência do documento (DF – *document frequency*), coeficiente de correlação (CC) e escore de relevância (ER).

5.2.1.1 – Resultados para Naive Bayes

Os valores de erro referem-se à média obtida nos três conjuntos de teste. Os menores valores estão em negrito.

Tabela 5.1 – Resultados do classificador Naive Bayes para o Corpus Jornal.

Termos	Métodos de seleção/N.termos	**30**	**60**	**90**	**120**	**150**
Substantivo	FD	20,70	16,96	15,44	13,69	**12,63**
	CC	22,11	16,49	13,68	11,82	**11,11**
	ER	23,39	14,27	10,88	10,17	**9,12**
Substantivo + Adjetivo	FD	21,17	15,79	14,50	14,15	**12,86**
	CC	19,88	15,09	12,40	10,29	**9,36**
	ER	21,29	13,57	10,64	9,83	**8,42**
Substantivo + Nome próprio	FD	25,49	18,48	16,96	14,97	**14,03**
	CC	24,21	16,61	14,15	11,23	**10,64**
	ER	25,14	14,62	13,57	10,99	**9,36**
Substantivo + Verbo	FD	38,60	24,91	21,75	19,65	**17,55**
	CC	22,93	16,61	13,57	11,23	**11,11**
	ER	23,28	13,22	10,76	9,47	**9,36**
Substantivo + Verbo + adjetivo	FD	38,60	23,04	19,53	18,13	**17,43**
	CC	19,65	14,27	11,46	10,18	**8,65**
	ER	20,00	13,45	10,29	**8,77**	**8,77**
Substantivo + Nome próprio + Adjetivo	FD	24,33	17,31	14,50	**12,40**	12,63
	CC	17,66	12,28	9,71	8,54	**7,49**
	ER	18,83	12,63	9,36	**8,19**	8,30
Nome próprio + Adjetivo	FD	44,21	35,67	33,33	31,58	**29,24**
	CC	32,98	24,91	23,04	21,29	**18,95**
	ER	35,44	27,25	24,80	21,29	**19,65**
Verbo	FD	67,13	58,60	53,92	49,82	**47,49**
	CC	44,44	38,36	36,61	36,02	**32,87**
	ER	44,68	39,53	37,43	35,56	**34,39**

Para melhor visualização abaixo de cada tabela foi plotado o gráfico de melhores valores ordenados decrescentemente.

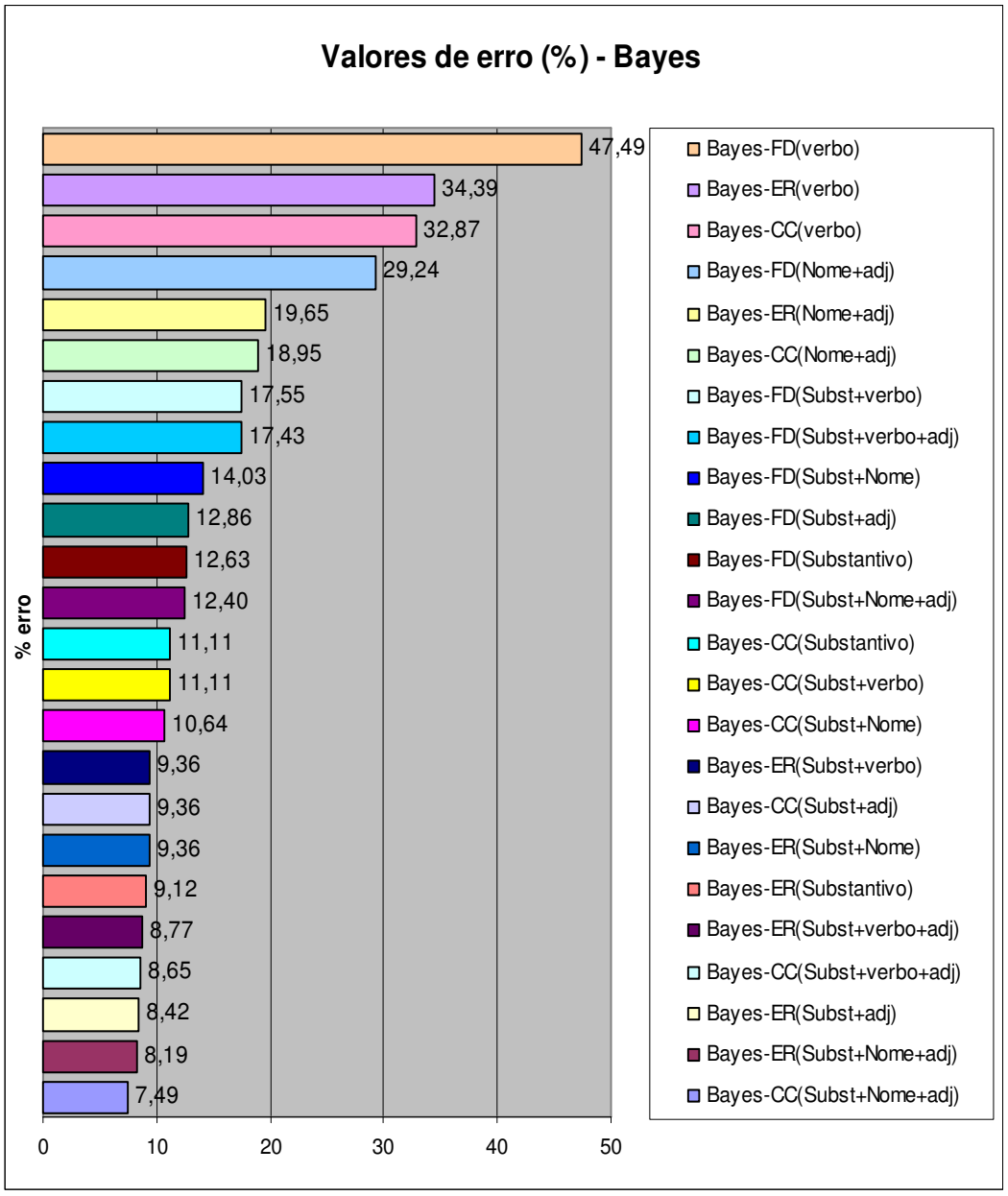

Figura 5.2 – Gráfico de melhores resultados Bayes – Corpus Jornal.

Tabela 5.2 – Resultados do classificador Naive Bayes para o Corpus Teses.

Termos	Métodos de seleção/N.termos	**30**	**60**	**90**	**120**	**150**
Substantivo	FD	41,07	37,49	34,55	33,92	**30,53**
	CC	29,68	25,16	24,30	23,44	**23,23**
	ER	31,40	25,81	**25,38**	26,02	25,80
Substantivo + Adjetivo	FD	41,23	35,32	35,70	**34,20**	35,22
	CC	27,31	24,30	21,29	21,50	**19,36**
	ER	29,89	22,36	18,71	18,28	**17,85**
Substantivo + Nome próprio	FD	39,30	35,86	35,01	34,44	**33,99**
	CC	29,68	27,74	24,73	23,66	**22,15**
	ER	31,40	26,88	**24,30**	25,16	26,23
Substantivo + Verbo	FD	45,42	39,95	36,86	34,13	**33,08**
	CC	29,89	26,24	26,67	23,66	**23,23**
	ER	30,97	26,88	**23,66**	26,67	26,02
Substantivo + Verbo + adjetivo	FD	46,93	41,26	36,61	34,26	**33,62**
	CC	26,25	23,07	19,10	17,84	**17,64**
	ER	27,47	22,45	18,45	18,07	**15,98**
Substantivo + Nome próprio + Adjetivo	FD	38,61	35,69	34,38	**31,84**	33,10
	CC	28,27	26,42	22,61	17,93	**16,87**
	ER	26,40	22,84	20,48	**20,09**	20,51
Nome próprio + Adjetivo	FD	65,05	57,26	53,47	50,53	**49,89**
	CC	49,46	41,72	**38,06**	39,57	45,16
	ER	**50,32**	55,48	54,41	55,92	54,63
Verbo	FD	75,79	71,16	66,53	65,05	**64,21**
	CC	49,05	39,16	38,53	36,42	**36,42**
	ER	64,63	49,68	45,05	39,37	**35,58**

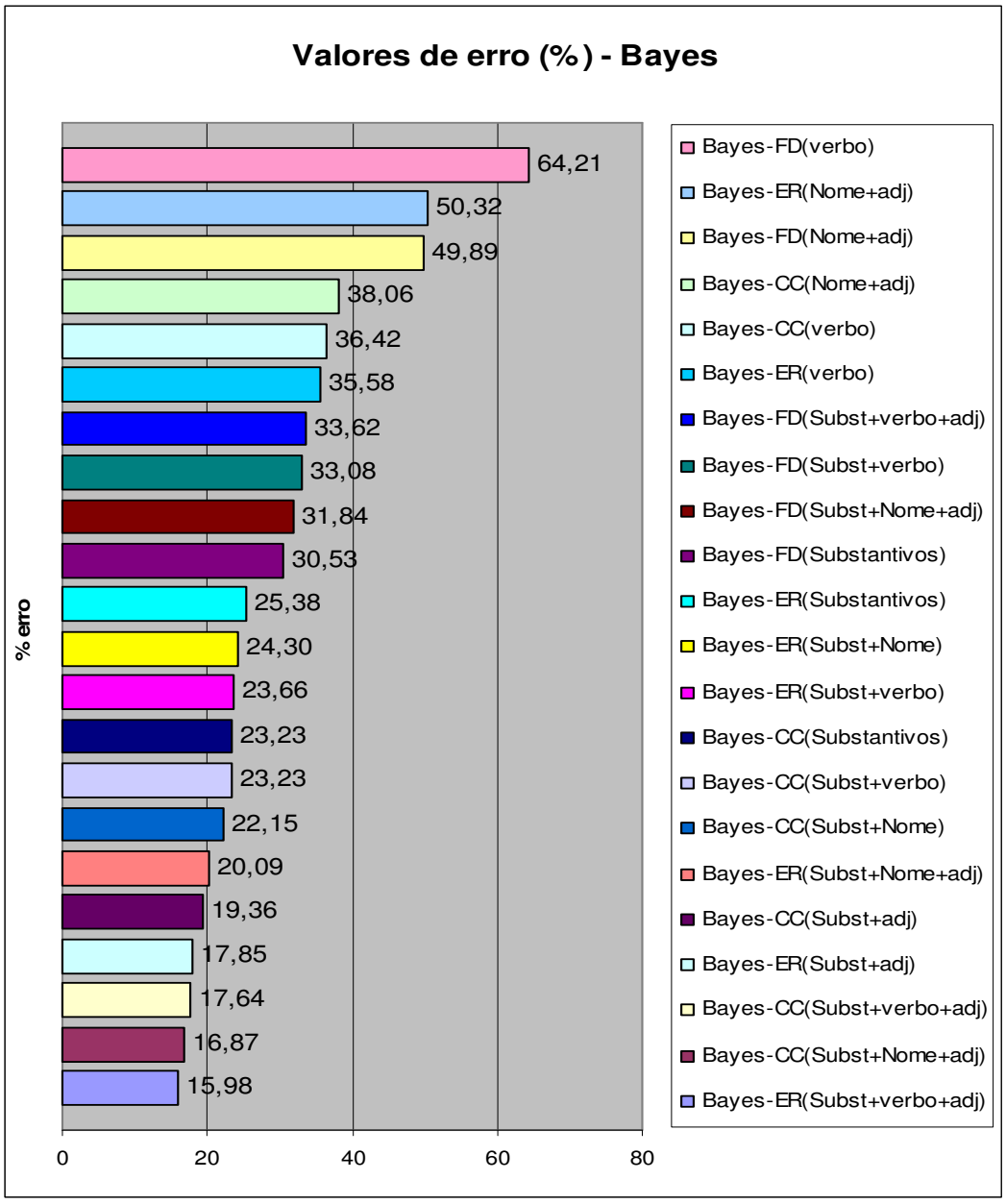

Figura 5.3 – Gráfico de melhores resultados Bayes – Corpus Teses.

5.2.1.2 - Resultados para Support Vector Machines (SVM)

Tabela 5.3 – Resultados do classificador SVM para o Corpus Jornal.

Termos	Métodos de seleção/N.termos	**30**	**60**	**90**	**120**	**150**
Substantivos	FD	23,97	19,06	16,14	14,85	**12,51**
	CC	30,29	22,46	17,66	15,09	**14,15**
	ER	30,41	19,76	15,21	13,34	**12,63**
Substantivo + Adjetivo	FD	27,25	18,01	17,19	15,56	**13,92**
	CC	29,36	21,76	17,78	15,79	**13,92**
	ER	27,60	18,72	16,02	13,69	**13,33**
Substantivo + Nome próprio	FD	28,77	21,29	19,18	17,78	**15,55**
	CC	32,75	24,80	21,05	15,32	**14,15**
	ER	31,58	20,70	16,96	14,97	**13,92**
Substantivo + Verbo	FD	41,29	27,02	21,40	19,77	**18,83**
	CC	33,10	23,51	17,90	16,14	**15,09**
	ER	31,81	19,30	15,67	13,45	**12,52**
Substantivo + Verbo + Adjetivo	FD	43,16	26,43	21,87	18,71	**18,13**
	CC	31,81	24,21	21,63	17,08	**16,73**
	ER	32,98	21,05	17,78	16,02	**14,74**
Substantivo + Nome próprio + Adjetivo	FD	29,59	21,99	19,53	16,49	**16,26**
	CC	29,12	20,00	17,66	16,02	**14,04**
	ER	26,55	19,77	17,19	14,86	**13,80**
Nome próprio + Adjetivo	FD	44,91	36,14	33,1	31,11	**29,47**
	CC	37,31	28,77	25,85	25,26	**21,75**
	ER	38,83	31,46	29,24	27,13	**24,56**
Verbo	FD	62,92	56,73	53,45	49,47	**47,02**
	CC	57,19	44,56	41,4	39,88	**39,42**
	ER	52,05	44,56	41,99	38,25	**37,78**

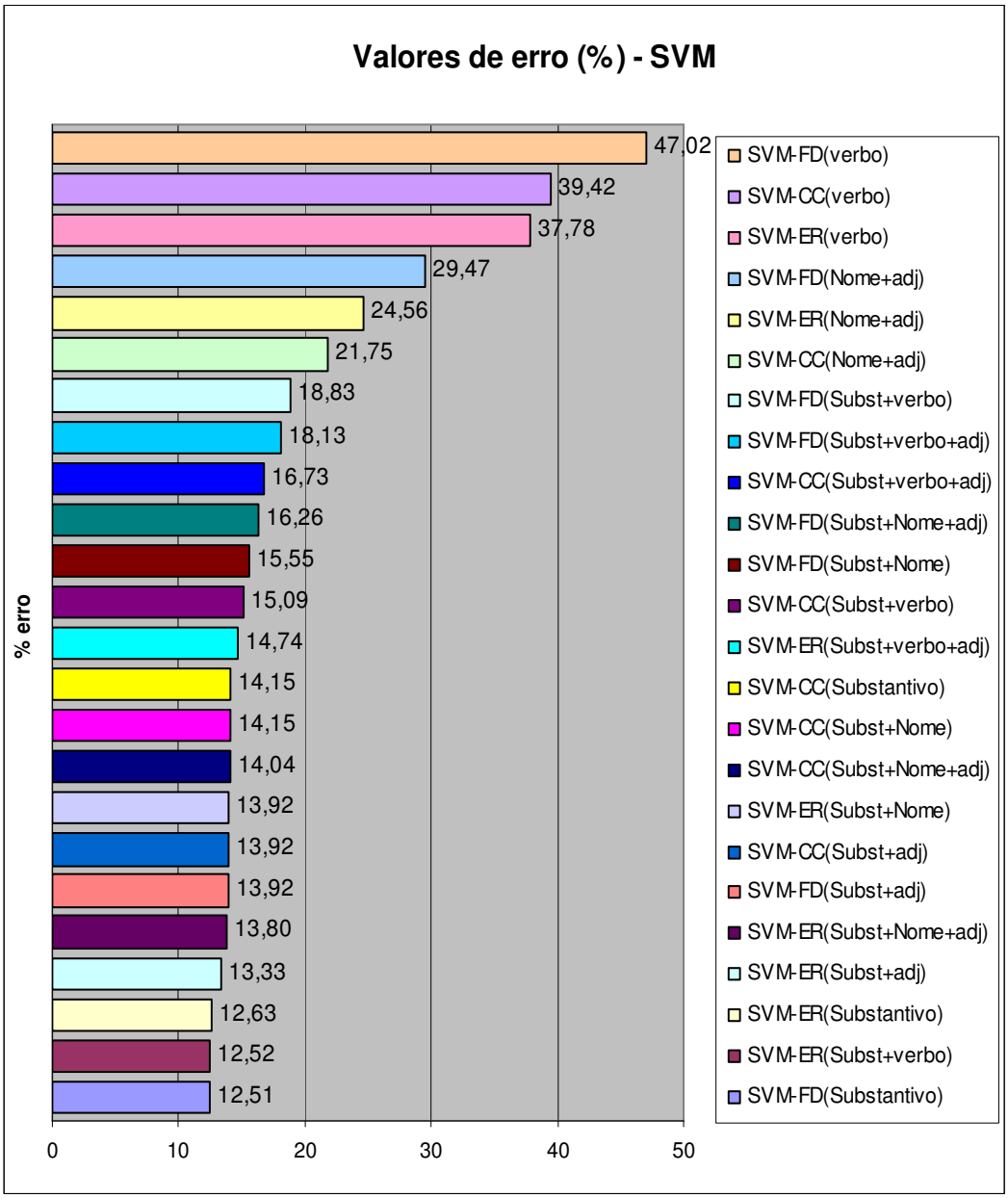

Figura 5.4 – Gráfico de melhores resultados SVM – Corpus Jornal.

Tabela 5.4 – Resultados do classificador SVM para o Corpus Teses:

Termos	Métodos de seleção/N.termos	**30**	**60**	**90**	**120**	**150**
Substantivos	FD	44,87	37,05	36,84	36,22	**35,41**
	CC	37,42	33,12	26,67	26,45	**24,95**
	ER	35,70	30,76	27,31	25,37	**24,30**
Substantivo + Adjetivo	FD	41,26	37,66	37,02	33,64	**32,77**
	CC	32,69	29,46	27,53	26,02	**25,59**
	ER	33,12	24,51	22,15	21,72	**21,29**
Substantivo + Nome próprio	FD	42,23	38,24	36,75	36,12	**35,29**
	CC	36,77	33,55	29,25	**26,88**	27,53
	ER	36,34	29,03	26,67	24,30	**23,87**
Substantivo + Verbo	FD	46,96	44,87	40,07	35,41	**34,57**
	CC	40,22	33,76	33,34	**30,11**	**30,11**
	ER	38,06	33,12	27,10	26,45	**24,73**
Substantivo + Verbo + Adjetivo	FD	48,81	45,02	42,27	43,34	**39,49**
	CC	31,98	31,74	28,83	26,11	**24,38**
	ER	34,50	31,12	24,56	23,09	**20,78**
Substantivo + Nome próprio + Adjetivo	FD	42,35	31,63	**30,56**	32,44	32,03
	CC	35,15	33,93	32,86	27,41	**27,20**
	ER	35,42	29,99	25,51	25,77	**21,19**
Nome próprio + Adjetivo	FD	61,33	57,54	54,04	51,37	**48,70**
	CC	55,48	49,03	45,59	**45,38**	48,82
	ER	**53,98**	59,35	59,36	58,49	56,13
Verbo	FD	73,33	69,97	64,28	62,64	**60,56**
	CC	66,18	56,56	56,63	46,74	**45,75**
	ER	72,14	58,25	56,77	48,91	**38,39**

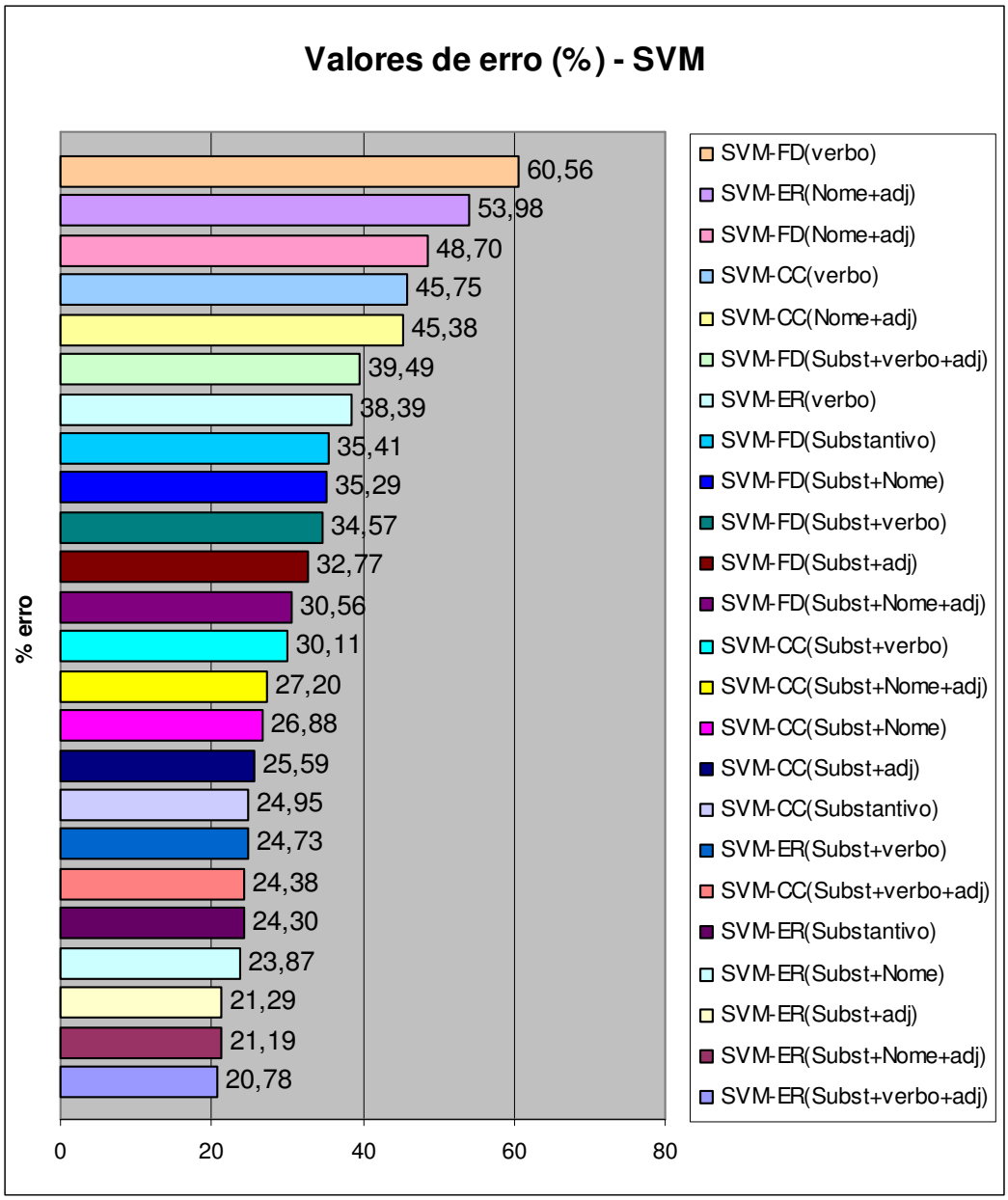

Figura 5.5 – Gráfico de melhores resultados SVM – Corpus Teses.

5.2.1.3 – Análise dos resultados para categorização baseada em termos

Dos resultados obtidos pode-se concluir que o classificador bayesiano é o que apresenta melhores resultados tanto para textos de jornal (7,49%) quanto para textos científicos (15,98%), confirmando os resultados da literatura. O classificador SVM, embora apresente bons resultados em trabalhos anteriores, não repetiu a preformance para os experimentos realizados.

Em relação aos métodos de seleção de atributos (termos) avaliados pode-se afirmar que o escore de relevância e o coeficiente de correlação apresentam melhores resultados que a freqüência do documento por considerar no cálculo de relevância os arquivos das não-classes onde o termo ocorre, diminuindo seu escore (do termo) quanto maior sua ocorrência nos arquivos das não-classes. O método escore de relevância apresentou melhor resultado em maior número de experimentos.

Pode-se afirmar que em função do algoritmo implementado para mapeamento dos pontos referentes a cada vetor, a Máquina de Vetor Suporte apresenta bons resultados para o método freqüência do documento, o que dependendo da aplicação a ser desenvolvida pode representar menos tempo de pré-processamento dos textos.

A combinação que melhor resultado obteve, no Corpus Jornal, foi a de substantivos, nomes próprios e adjetivos confirmando os resultados de SILVA (2004). No entanto, para os textos das teses a combinação que melhores resultados obteve foi a de substantivos, verbos e adjetivos, para os dois classificadores, evidenciando que os substantivos e adjetivos são de fundamental importância na categorização de textos, seja por sua freqüência, seja pela informação semântica que carregam. Nos textos das teses verifica-se que a natureza dos verbos influencia na sua relevância. Por exemplo, nos textos de jornal, os verbos são, em sua maioria, verbos de ligação que aparecem em praticamente todos os textos, tais como *ser, estar, poder*, etc., e que no cálculo de relevância, por aparecerem em muitos textos das não-classes, recebem valores baixos de escore. Já nos textos científicos os verbos são mais específicos e correlacionados com a área de conhecimento. Por exemplo, os verbos *monitorar* e *controlar*, *sintetizar* e *automatizar*, *codificar* e *chavear*, *distribuir* e *difundir*, *interligar* e *operar*, para as classes Controle, Microeletrônica, Processamento de sinais, Redes de Computadores e Sistemas de Potência, respectivamente.

Os resultados inferiores obtidos pelos textos da teses (científicos) são devidos aos estilos próprios de cada tipo de coleção. Este fato aliado a especificidade dos textos das teses, i.e., cada tese trata de um assunto extremamente específico, justifica a dificuldade em encontrar padrões neste tipo de texto científico.

Para avaliação do efeito da quantidade de palavras diferentes por coleção foram igualados artificialmente os números de arquivos entre as duas coleções

(selecionados apenas 95 arquivos por classe da coleção Jornal) e os resultados mantiveram-se nos mesmos patamares evidenciando que a menor quantidade de arquivos da coleção Teses não influenciou no resultado já que mesmo com o mesmo número de arquivos (e de palavras) a Coleção Jornal apresentou resultados excelentes. Para efeito de comparação a tabela abaixo mostra as quantidades de palavras diferentes por coleção sem a alteração das quantidades originais.

Tabela 5.5 – Quantidade de palavras por coleção.

Coleção	**Jornal**				**Teses**			
Combinação sintática/conjuntos	V1	V2	V3	Total	V1	V2	V3	total
Substantivo	1344	1310	1332	1739	641	606	615	802
Substantivo+adjetivo	1650	1604	1655	2159	884	831	835	1100
Substantivo+nome próprio	1896	1919	1941	2622	878	820	797	1134
Substantivo+verbo	1935	1879	1891	2459	901	848	848	1105
Substantivo+verbo+adjetivo	2244	2173	2213	2879	1139	1076	1076	1406
Substantivo+nome+adjetivo	2200	2213	2264	3042	1119	1051	1029	1438
Nome próprio+adjetivo	1035	1007	1038	1355	541	508	511	673
Verbo	573	556	560	728	251	236	236	308

5.2.2 – Resultados para categorização baseada em duplas sintáticas

Nesta seção são apresentados os resultados obtidos no processo de categorização dos documentos dos dois corpus, para as mesmas técnicas de seleção de atributos (FD, CC e ER), utilizando-se de estruturas no formato sujeito_(verbo), verbo_(objeto direto ou indireto), verbo_(predicativo), e verbo_(agente da passiva). A formação dos vetores segue a mesma codificação em número de termos que a codificação por palavras, ou seja, 30, 60, 90, 120 e 150 termos nos vetores globais, só que neste caso as duplas representam os termos que representam as classes.

5.2.2.1 – Resultados para Naive Bayes

Tabela 5.6 - Resultados do classificador Naive Bayes para o Corpus Jornal:

Seleção/N.termos	**30**	**60**	**90**	**120**	**150**
FD	65,89	61,33	56,54	56,19	**55,49**
CC	66,24	60,4	55,26	**53,15**	53,62
ER	64,95	58,29	54,44	53,04	**52,34**

Tabela 5.7 - Resultados do classificador Naive Bayes para o Corpus Teses:

Seleção/N.termos	**30**	**60**	**90**	**120**	**150**
FD	70,32	61,47	59,16	**58,32**	58,53
CC	70,95	67,58	61,68	56,21	**54,74**
ER	73,68	68,42	60,63	58,11	**55,16**

5.2.2.2 – Resultados para Support Vector Machines (SVM)

Tabela 5.8 - Resultados do classificador SVM para o Corpus Jornal:

Seleção/N.termos	**30**	**60**	**90**	**120**	**150**
FD	67,41	62,85	56,31	55,14	**54,21**
CC	65,65	59,70	55,61	53,39	**52,22**
ER	66,36	59,46	54,67	53,62	**50,93**

Tabela 5.9 - Resultados do classificador SVM para o Corpus Teses:

Seleção/N.termos	**30**	**60**	**90**	**120**	**150**
FD	74,53	58,53	54,74	54,11	**52,00**
CC	72,84	65,89	61,26	52,42	**50,32**
ER	73,89	68,63	57,89	52,63	**51,37**

5.2.2.3 – Análise dos resultados para categorização baseada em duplas sintáticas

Os resultados obtidos não recomendam a utilização das duplas mesmo com o acréscimo semântico esperado. Os resultados podem ser explicados por dois fatores. O primeiro é a quantidade de vetores nulos gerados para representar os arquivos exemplos. Isto porque devido a seleção de termos com base no cálculo da relevância e a técnica de truncagem utilizada, o conjunto de termos mais relevantes selecionado nem sempre estará presente em alguns arquivos dos textos utilizados para treinamento, assim, cada dupla sintática (posição no vetor) terá peso zero e, portanto, será gerado um vetor nulo (todas as posições do vetor zeradas). Se estes vetores nulos forem utilizados como entrada na ferramenta de classificação WEKA, o valor zero será considerado diminuindo a performance do classificador.

O segundo fator é o número de duplas geradas por arquivo face ao estilo do escritor e a estrutura mais elaborada das frases. Esta estrutura mais elaborada torna a formação das duplas uma tarefa complexa. Pois a maioria dos textos é composta por

períodos compostos, ou seja, é um conjunto de orações. A oração é a idéia que se organiza em torno de um verbo, sendo que o verbo pode estar elíptico (não aparece mas existe). E o período composto pode ser:

- Período composto por coordenação: é aquele em que as orações se ligam pelo sentido mas não existe dependência entre elas. As orações coordenadas se dividem em assindéticas (não são introduzidas por conjunção) e sindéticas (são introduzidas por conjunção). As orações coordenadas sindéticas podem ser aditivas, adversativas, alternativas, conclusivas ou explicativas, dependendo da idéia que expressem.
- Período composto por subordinação: é aquele no qual existe pelo menos uma oração principal e uma subordinada. A oração principal é sempre incompleta, ou seja, alguma função sintática está faltando. As orações subordinadas desempenham a função sintática que falta na principal: objeto direto ou indireto, sujeito, predicativo, etc. As orações subordinadas se dividem em substantivas, adjetivas ou adverbiais.

Desta forma, além do analisador sintático não cobrir todas as possibilidades de construção dos períodos por utilizar uma versão mais simples da gramática da língua analisada, e ao fato do programa elaborado em Java não extrair todas as combinações possíveis e corretas, o número de duplas gerado automaticamente é muito menor do que o número de termos obtidos dos textos, ou ainda, do que o número de duplas que seria gerado manualmente.

5.2.3 - Resultados para categorização baseada em combinações de termos com duplas sintáticas

Nesta seção são apresentados os resultados obtidos no processo de categorização dos documentos dos dois corpus, para as técnicas de seleção de atributos de Escore de Relevância e Coeficiente de Correlação, utilizando-se tanto os termos como as duplas. O classificador é o Naive Bayes, devido aos melhores resultados apresentados.

Face à alta incidência dos termos comparados as duplas, os vetores locais de cada categoria não incluíram as duplas quando da truncagem dos mesmos, desta forma, foram inseridas as duplas de maiores escores artificialmente. Assim, para cada grupo de termos foram inseridas 1 ou 2 duplas de modo a obterem-se vetores locais com 20 termos e 10 duplas, e 25 termos e 5 duplas.

O conjunto de termos selecionados foi aquele que melhor resultado apresentou na categorização por termos, no caso do Corpus Jornal substantivo, nome próprio e adjetivo (7,49%) e do Corpus Teses substantivo, verbo e adjetivo (15,98%). Os melhores valores estão em negrito. A motivação para este experimento é avaliar o

aumento do erro propiciado pelo aumento do valor semântico dos termos representativos de cada classe.

5.2.3.1 – Resultados para Naive Bayes

- Corpus Jornal:

Tabela 5.10 – Resultados para Naive Bayes (combinação 25 termos x 5 duplas)

Duplas+Substantivos+Nomes próprios+adjetivos		CC (25 termos x 5 duplas)			
Conjunto/N.termos	**30**	**60**	**90**	**120**	**150**
V1	22,46	17,89	15,44	**12,28**	12,98
V2	23,86	11,93	8,42	**8,07**	8,77
V3	20,00	15,79	13,33	**10,18**	**10,18**
Média	22,11	15,20	12,40	**10,18**	10,64

Tabela 5.11 – Resultados para Naive Bayes (combinação 20 termos x 10 duplas)

Duplas+Substantivos+Nomes próprios+adjetivos		CC (20 termos x 10 duplas)			
Conjunto/N.termos	**30**	**60**	**90**	**120**	**150**
V1	29,47	19,30	17,19	15,44	**12,28**
V2	27,02	17,89	10,88	8,42	**8,07**
V3	22,81	16,14	15,79	11,93	**10,18**
Média	26,43	17,78	14,62	11,93	**10,18**

- Corpus Teses:

Tabela 5.12 – Resultados para Naive Bayes (combinação 25 termos x 5 duplas)

Duplas+Substantivos+verbos +adjetivos		ER (25 termos x 5 duplas)			
Conjunto/N.termos	**30**	**60**	**90**	**120**	**150**
V1	30,32	24,52	**16,13**	19,35	18,71
V2	24,38	20,00	**14,38**	16,25	16,25
V3	34,38	25,00	16,88	**16,25**	21,88
Média	29,69	23,17	**15,80**	17,28	18,95

Tabela 5.13 – Resultados para Naive Bayes (combinação 20 termos x 10 duplas)

Duplas+Substantivos+verbos +adjetivos		ER (20 termos x 10 duplas)			
Conjunto/N.termos	**30**	**60**	**90**	**120**	**150**
V1	26,45	26,45	19,35	**16,13**	19,35
V2	25,63	20,00	**15,00**	15,63	15,63
V3	31,88	30,63	20,00	19,38	**17,50**
Média	27,99	25,69	18,12	**17,05**	17,49

5.2.3.2 – Análise dos resultados para categorização baseada em combinações de termos com duplas sintáticas

Como se pode observar, a inclusão das duplas sintáticas diminuiu a performance do classificador para o Corpus Jornal. Isto porque ao inserirem-se as duplas artificialmente, foram retiradas, dos vetores que representam cada arquivo, outras dimensões de maior peso já que as duplas não atingiram escore suficiente para sua utilização como dimensões dos referidos vetores.

Embora o resultado seja inferior pode-se considerar a inclusão das duplas no intuito de aumentar-se o conteúdo semântico dos termos que representam cada categoria. O acréscimo de erro foi de 2,69% em valores absolutos para o Corpus Jornal.

Já no Corpus Teses, isto não ocorreu com a inserção de cinco duplas pois algumas delas se equiparavam em freqüência aos termos escolhidos para representação. Isto ocasionou uma melhora no resultado de 0,18%, valor quase irrisório mas que aliado ao acréscimo semântico obtido, pode ser interessante de acordo com a aplicação a ser desenvolvida. No caso da inserção de dez duplas o acréscimo de erro foi de 1,07%.

5.2.4 – Aplicação da Decomposição por valor unitário

Através da Decomposição de Valor Singular (SVD), obteremos a matriz original (*M*) como um produto de três matrizes:

M = T x S x D, onde: *T* = matriz de vetores singulares à esquerda; *S* = matriz diagonal de valores singulares em ordem decrescente; *D* = matriz de vetores singulares à direita.

Reduzimos, então, a dimensão destas matrizes, eliminando as linhas e colunas correspondentes aos menores valores singulares da matriz *S* assim como as colunas da matriz *T* e linhas da matriz *D* correspondentes.

A Decomposição de Valor Singular é normalmente utilizada para localizar a informação semântica essencial em uma matriz de co-ocorrência de palavras. Com isto, a partir desta decomposição, é possível, com a redução de dimensão das matrizes *T*, *S* e *D (*mantendo somente os maiores valores singulares), descartar as informações acidentais que geralmente estão presentes. Sendo assim, o objetivo com o produto das três novas matrizes reduzidas é obter um espaço semântico condensado que revele as melhores relações entre as palavras e documentos. Porém, o número de dimensões a ser reduzida de forma a otimizar o resultado é bastante questionado, e, segundo EFRON (2003), parece estar bastante relacionado ao *corpus* (coleção de documentos) utilizado para a construção do espaço.

Neste trabalho adotou-se a eliminação do mesmo número de termos da matriz diagonal para cada um dos exemplos, mesmo que eles diferissem em tamanho. A motivação é avaliar se os termos obtidos para representação de cada categoria pelos métodos de seleção utilizados (Coeficiente de correlação, escore de relevância e freqüência do documento) podem ser refinados, ou seja, escolhidos somente aqueles que efetivamente agregam informação para o classificador. Ou ainda, avaliar se o espaço semântico condensado traz vantagens para a classificação de textos em português.

Os exemplos escolhidos foram aqueles que melhores resultados obtiveram na classificação por termos, ou seja, para Naive Bayes a combinação de substantivos, nomes próprios e adjetivos para o Corpus Jornal e a combinação substantivo, verbo e adjetivo para o Corpus Teses e, para SVM substantivos para o Corpus Jornal e a combinação substantivo, verbo e adjetivo para o Corpus Teses.

Os valores apresentados referem-se à média de erro de teste dos três conjuntos de dados (V1, V2 e V3) tanto para os valores de erro como para o número de acertos. Convêm lembrar que para o Corpus Jornal o número de arquivos de teste é 285 e para o Corpus Teses é 155 para o conjunto V1, e 160 para os conjuntos V2 e V3.

5.2.4.1 – Resultados para Naive Bayes

- Corpus Jornal:

Tabela 5.14 – Resultados para SVD (Naive Bayes) – Corpus Jornal.

Categoria	Método de seleção	% erro	acertos	**N.termos eliminados**	**SVD - % erro**	acertos
Substantivo + Nome próprio + adjetivo – (150 termos)	CC	7,49	263,67	**1**	7,60	263,33
		7,49	263,67	**2**	7,60	263,33
		7,49	263,67	**5**	7,84	262,67
		7,49	263,67	**10**	7,60	263,33
Substantivo + Nome próprio + adjetivo – (120 termos)	ER	8,30	260	**1**	8,42	261
		8,30	260	**2**	8,30	261,33
		8,30	260	**5**	8,19	261,67
		8,30	260	**10**	8,19	261,67
Substantivo + Nome próprio + adjetivo - (120 termos)	FD	12,63	249	**1**	12,40	249,67
		12,63	249	**2**	13,10	247,67
		12,63	249	**5**	13,10	247,67
		12,63	249	**10**	13,10	247,67

- Corpus Teses:

Tabela 5.15 – Resultados para SVD (Naive Bayes) – Corpus Teses.

Categoria	Método de seleção	% erro	acertos	**N.termos eliminados**	**SVD - % erro**	acertos
Substantivo + verbo + adjetivo – (150 termos)	CC	17,64	130,33	**1**	17,85	130
		17,64	130,33	**2**	17,85	130
		17,64	130,33	**5**	17,85	130
		17,64	130,33	**10**	17,85	130
Substantivo + verbo + adjetivo – (150 termos)	ER	15,98	133	**1**	15,98	133
		15,98	133	**2**	15,98	133
		15,98	133	**5**	15,98	133
		15,98	133	**10**	15,98	133
Substantivo + verbo + adjetivo - (150 termos)	FD	33,62	105	**1**	33,62	105
		33,62	105	**2**	33,62	105
		33,62	105	**5**	33,62	105
		33,62	105	**10**	33,62	105

5.2.4.2 - Resultados para Support Vector Machines (SVM)

- Corpus Jornal:

Tabela 5.16 – Resultados para SVD (SVM) – Corpus Jornal.

Categoria	Método de seleção	% erro	acertos	**N.termos eliminados**	**SVD - % erro**	acertos
Substantivo – (150 termos)	CC	14,15	244,67	**1**	14,04	245
		14,15	244,67	**2**	13,80	245,67
		14,15	244,67	**5**	14,27	244,33
		14,15	244,67	**10**	14,15	244,67
Substantivo – (150 termos)	ER	12,63	249	**1**	12,39	249,67
		12,63	249	**2**	12,75	248,67
		12,63	249	**5**	12,52	249,33
		12,63	249	**10**	12,40	249,67
Substantivo - (150 termos)	FD	12,51	249,33	**1**	12,51	249,33
		12,51	249,33	**2**	12,40	249,67
		12,51	249,33	**5**	12,40	249,67
		12,51	249,33	**10**	12,40	249,67

- Corpus Teses:

Tabela 5.17 – Resultados para SVD (SVM) – Corpus Teses.

Categoria	Método de seleção	% erro	acertos	**N.termos eliminados**	**SVD - % erro**	acertos
Substantivo + verbo + adjetivo – (150 termos)	CC	24,38	120	**1**	24,18	120
		24,38	120	**2**	23,97	120,33
		24,38	120	**5**	23,97	120,33
		24,38	120	**10**	23,97	120,33
Substantivo + verbo + adjetivo – (150 termos)	ER	20,78	125,33	**1**	21,61	124
		20,78	125,33	**2**	20,78	125,33
		20,78	125,33	**5**	20,99	125
		20,78	125,33	**10**	20,99	125
Substantivo + verbo + adjetivo - (150 termos)	FD	39,49	95,67	**1**	39,49	95,67
		39,49	95,67	**2**	39,49	95,67
		39,49	95,67	**5**	39,49	95,67
		39,49	95,67	**10**	39,49	95,67

5.2.4.3 – Análise da aplicação da Decomposição por valor unitário

Para melhor visualização os resultados foram listados na tabela 5.18 abaixo.

Tabela 5.18 – Resultados da aplicação da SVD.

Classificador	Naive Bayes			SVM		
Corpus Resultado\	Jornal	Teses	Total	Jornal	Teses	Total
Melhor	3	0	3	8	4	12
Pior	8	4	12	2	3	5
Igual	1	8	9	2	5	7

Podemos observar que há uma grande diferença de resultados entre classificadores, o que pode ser explicado pelo algoritmo empregado por cada um deles na separação dos pontos de cada classe. O classificador SVM apresenta melhores resultados pois sua superfície de separação é melhorada com a variação das dimensões dos vetores que representam cada classe, ou seja, a alteração dos valores de peso de cada termo, mesmo que pequena, representa a alteração de coordenadas dos pontos escolhidos como superfície de separação entre classes. Já o classificador bayesiano não tem melhora porque as alterações nos pesos correspondem a pequenas alterações nas probabilidades de ocorrência de cada termo. E já que a classificação se dará pela soma de probabilidades de cada dimensão (termo), as pequenas alterações promovidas pela SVD não influenciam no resultado final salvo em poucos casos. Pode-se observar que, ao contrário, os resultados deste classificador sofrem prejuízo com a aplicação desta técnica.

Os resultados evidenciam que as diferenças entre coleções afetam os resultados obtidos na SVD, como já comprovado na literatura. O número de termos eliminados parece não ter influência neste caso estudado, já que não é possível traçar um paralelo entre o número de termos eliminados e os resultados melhor, pior ou igual.

5.2.5 – Considerações parciais

Os experimentos realizados neste trabalho comprovam os resultados da literatura se considerarmos a categorização baseada em termos. No entanto os trabalhos realizados com palavras em português são em número reduzido e esta dissertação agrega conhecimento quando compara o classificador Bayes e o SVM,

aliados a seleção dos termos por sua função sintática (o trabalho com tais informações realizado por Silva (2004) utilizou a classificação através de redes neurais e seleção por freqüência do termo redundando em 16,96% de erro como melhor resultado). Ao considerarmos estas informações dos termos aumentamos o conteúdo semântico que há em cada palavra isoladamente e ainda obtém-se um erro de apenas 7,49% com a escolha correta de classificador e método de seleção de termos.

Apesar dos excelentes resultados obtidos pelos termos as duplas não repetiram o êxito devido ao problema de construção das mesmas. O número de duplas geradas automaticamente é ainda muito inferior ao gerado manualmente. Isto acarreta a insuficiência de termos para um cálculo eficiente de relevância além da impossibilidade de serem geradas duplas que efetivamente representem cada categoria. Tal estratégia já havia sido tentada por Cimiano (2005) para agrupamento e os resultados foram pouco satisfatórios.

A fusão dos termos com as duplas apresentou resultado proporcional a razão entre o número de termos e o número de duplas selecionados para representação dos textos, exceto na representação dos textos das teses que obtém melhora no acerto com conseqüente aumento do conteúdo semântico dos termos que representam os textos. Isto evidencia que dependendo da natureza dos textos a utilização das duplas pode ser um processo viável para melhora de performance do classificador.

O objetivo de classificadores automáticos de textos é atingir níveis de acerto próximos a 100%, no entanto, isto na prática é inviável e volta o foco das pesquisas para avaliação das melhores combinações de técnicas que redundem nesse nível de acerto. É nesse contexto que foi avaliada a aplicação da decomposição por valor unitário que visa a otimizar o grupo de termos escolhidos para representação dos textos. Os resultados obtidos não recomendam o uso desta técnica pelo acréscimo de processamento que acarreta e pelas reduzidas melhoras nos valores de erro dos exemplos avaliados. Na busca de menores valores de erro, será avaliada no próximo item a combinação de classificadores.

5.2.6 – *Ensemble* de classificadores

Visando otimizar o resultado obtido nos classificadores Bayes e SVM foram combinados os dois classificadores (Classificadores 1 e 2 da figura 5.6) de modo a combinar os melhores resultados e com isso melhorar a margem de erro final. Para tanto as regras criadas para o Classificador 3 são bastante simples de modo a avaliar a possibilidade de melhoria com o menor esforço computacional possível. O resultado desta classificação será a combinação dos resultados do classificador bayesiano (Classificador 1) e do classificador SVM (Classificador 2) através das regras enunciadas abaixo. Nestas regras entende-se como claramente vencedora a classe cujo ranking seja superior a 20% do valor da classe imediatamente inferior e indecisão como sendo a classificação idêntica em duas classes, ou seja classes diferentes tendo classificação de claramente vencedora ou vencedora. Os valores de ranking de todos os classificadores variam entre zero e um. Abaixo as regras aplicadas.

1) Se Classificador 1 = classe X claramente vencedora e Classificador 2 = classe X claramente vencedora então Classificador 3 = classe X;
2) Se Classificador 1 = classe X claramente vencedora e Classificador 2 = classe X vencedora ou indecisão ou classe Y vencedora então Classificador 3 = classe X;
3) Se Classificador 1 = classe X vencedora e Classificador 2 = classe X vencedora então Classificador 3 = classe X;
4) Se Classificador 1 = classe X vencedora e Classificador 2 = indecisão então Classificador 3 = classe X;
5) Se Classificador 1 = classe X vencedora e Classificador 2 = classe Y vencedora então Classificador 3 = indecisão;
6) Se Classificador 1 = classe X claramente vencedora e Classificador 2 = classe Y claramente vencedora então Classificador 3 = indecisão;
7) Se Classificador 1 = classe X indecisão e Classificador 2 = indecisão então Classificador 3 = indecisão.

A topologia deste sistema seria a seguinte:

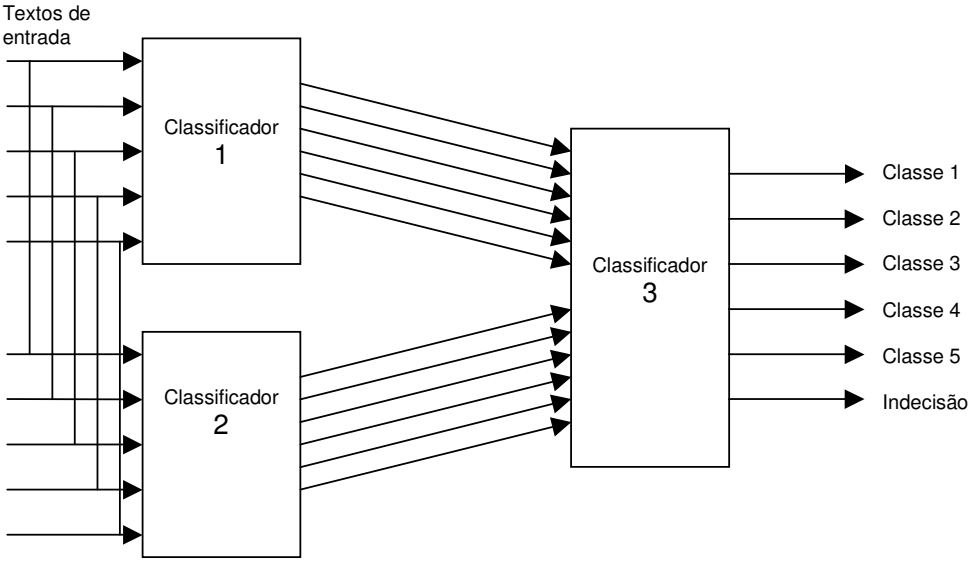

Figura 5.6 – Topologia do ensemble de classificadores.

Os classificadores apresentaram seus melhores resultados para o corpus Jornal com métodos diferentes de seleção de características, coeficiente de correlação para o classificador bayesiano e escore de relevância para o classificador SVM. Já no corpus Teses os melhores resultados foram obtidos com o escore de relevância.

Pode-se perceber que há uma pequena diferença a maior no erro dos classificadores se comparados às tabelas 5.1 a 5.4. Isto é devido ao fato de na ferramenta *weka* a indecisão ser considerada como classificação correta na primeira categoria inserida como exemplo. Nas tabelas seguintes isto não ocorre daí a diferença. Abaixo os resultados obtidos para os dois Corpus (valores médios dos conjuntos V1, V2, V3):

- Corpus Jornal:

Tabela 5.19 – Resultados do ensemble de classificadores.

Classificador / método de seleção	Classificador 1 CC	Classificador 2 ER	Classificador 3
Categoria	% erro	% erro	% erro
Esporte	4,68%	16,37%	15,20%
Imóveis	13,45%	14,62%	17,54%
Informática	5,85%	12,28%	14,62%
Política	5,85%	16,96%	19,30%
Turismo	8,19%	3,51%	3,51%
Média	**7,60%**	**12,75%**	**14,04%**

- Corpus Teses:

Tabela 5.20 – Resultados do ensemble de classificadores.

Classificador / método de seleção	Classificador 1 ER	Classificador 2 ER	Classificador 3
Categoria	% erro	% erro	% erro
Esporte	9,48%	29,44%	25,20%
Imóveis	29,30%	27,08%	28,16%
Informática	14,72%	9,44%	10,52%
Política	11,63%	25,30%	23,25%
Turismo	14,78%	16,87%	16,87%
Média	**15,98%**	**21,63%**	**20,80%**

Os resultados evidenciam que não houve melhora nos resultados após a combinação dos classificadores. Isto porque o ensemble de classificadores deve ser realizado com classificadores que apresentem resultados diferentes para o mesmo conjunto de dados. No entanto, mesmo com algoritmos distintos, os dois classificadores combinados apresentaram o mesmo comportamento com os exemplos, acarretando acertos e erros coincidentes e consequentemente não deixando margem para a obtenção de melhoras nos resultados. Desta forma, visando obter um classificador que apresente características diferentes, foi desenvolvido um classificador baseado em regras lógicas (classificador 2). Para tanto, neste classificador, adotou-se como palavras representativas de cada classe o vetor local gerado com os exemplos de treinamento de cada categoria. Assim, cada documento será classificado como pertencente ou não a determinada classe se o número de palavras encontradas no texto for o maior número possível de palavras que compõem o vetor local de determinada categoria.

Para seleção do número de termos para a composição das regras lógicas do classificador 2 foram elaborados vetores locais com 20, 40, 60, 80 e 100 termos, perfazendo um total de 100, 200, 300, 400 e 500 termos nos vetores globais gerados pelos métodos de seleção de atributos que melhores resultados obtiveram nas classificações por Naive Bayes baseadas em termos, ou seja, escore de relevância e coeficiente de correlação.

Abaixo os resultados obtidos para os dois Corpus (valores médios dos conjuntos V1, V2, V3):

- Corpus Jornal:

Tabela 5.21 – Resultados do classificador de regras lógicas (Coeficiente de correlação).

Método de seleção	Coeficiente de correlação				
Classe / número de termos	**100**	**80**	**60**	**40**	**20**
Esporte	5,26%	3,51%	3,51%	4,09%	5,85%
Imóveis	15,79%	15,20%	15,79%	16,37%	18,71%
Informática	5,26%	5,26%	5,85%	8,77%	17,54%
Política	5,85%	7,02%	6,43%	6,43%	8,19%
Turismo	14,04%	18,13%	19,30%	20,47%	22,22%
Média	**9,24%**	9,82%	10,18%	11,23%	14,50%

Tabela 5.22 – Resultados do classificador de regras lógicas (Escore de relevância).

Método de seleção	Escore de relevância				
Classe / número de termos	**100**	**80**	**60**	**40**	**20**
Esporte	7,02%	5,26%	5,26%	5,85%	5,85%
Imóveis	14,04%	18,71%	13,45%	15,20%	18,71%
Informática	7,02%	6,43%	5,26%	7,60%	14,62%
Política	7,60%	9,36%	9,94%	8,77%	8,77%
Turismo	11,11%	12,87%	18,71%	22,81%	31,58%
Média	**9,36%**	10,53%	10,53%	12,05%	15,91%

- Corpus Teses:

Tabela 5.23 – Resultados do classificador de regras lógicas (Coeficiente de correlação).

Método de seleção	Coeficiente de correlação				
Classe / número de termos	**100**	**80**	**60**	**40**	**20**
Controle	9,51%	10,58%	11,63%	12,74%	20,13%
Microeletrônica	17,78%	23,05%	25,13%	24,03%	25,13%
Processamento de sinais	12,63%	15,79%	15,79%	23,12%	25,30%
Redes	5,24%	5,24%	9,51%	7,36%	9,51%
Sistemas de potência	19,99%	19,96%	17,81%	24,16%	30,34%
Média	**13,03%**	14,93%	15,97%	18,28%	22,08%

Tabela 5.24 – Resultados do classificador de regras lógicas (Escore de relevância).

Método de seleção	Escore de relevância				
Classe / número de termos	**100**	**80**	**60**	**40**	**20**
Controle	21,17%	24,36%	19,05%	16,97%	23,35%
Microeletrônica	29,50%	27,42%	27,42%	17,81%	27,28%
Processamento de sinais	40,02%	37,94%	40,09%	30,61%	21,03%
Redes	8,37%	8,37%	8,37%	14,75%	16,94%
Sistemas de potência	36,90%	37,94%	37,94%	29,47%	26,28%
Média	27,19%	27,20%	26,57%	**21,92%**	22,98%

Face aos resultados obtidos foram escolhidos para os Corpus Jornal e Teses o classificador (Classificador 2) baseado em regras lógicas composto de 100 termos no vetor local de cada categoria selecionados pelo método Coeficiente de correlação.

Os melhores resultados do classificador 1 (Bayes) foram obtidos pelos métodos de seleção coeficiente de correlação para o Corpus Jornal e escore de relevância para o Corpus Teses.

Abaixo serão listados os resultados para o Classificador 3 baseado nas regras lógicas supracitadas. Os valores correspondem aos valores de erros médios obtidos nos três conjuntos de dados (V1, V2 e V3).

- Corpus Jornal:

Tabela 5.25 – Resultados do ensemble de classificadores.

Coeficiente de correlação	Classificador 1	Classificador 2	Classificador 3
Categoria	% erro	% erro	% erro
Esporte	4,68%	5,26%	4,68%
Imóveis	13,45%	15,79%	10,53%
Informática	5,85%	5,26%	6,43%
Política	5,85%	5,85%	5,26%
Turismo	8,19%	14,04%	8,77%
Média	**7,60%**	**9,24%**	**7,13%**

Para diminuir o erro do classificador 2 foram selecionados os menores valores de erro para os dois métodos de seleção de atributos (tabelas 5.21 e 5.22) obtendo-se o resultado abaixo.

Tabela 5.26 – Resultados do ensemble de classificadores.

Coeficiente de correlação + Escore de relevância	Classificador 1	Classificador 2 (CC + ER)	Classificador 3
Categoria	% erro	% erro	% erro
Esporte	4,68%	5,26%	4,68%
Imóveis	13,45%	14,04%	11,11%
Informática	5,85%	5,26%	6,43%
Política	5,85%	5,85%	5,26%
Turismo	8,19%	11,11%	7,02%
Média	**7,60%**	**8,30%**	**6,90%**

- Corpus Teses:

Tabela 5.27 – Resultados do ensemble de classificadores.

Coeficiente de correlação	Classificador 1 (ER)	Classificador 2 (CC)	Classificador 3
Categoria	% erro	% erro	% erro
Controle	9,48%	9,51%	9,54%
Microeletrônica	29,30%	17,78%	16,77%
Processamento de sinais	14,72%	12,63%	11,59%
Redes	11,63%	5,24%	12,67%
Sistemas de potência	14,78%	19,99%	16,90%
Média	**15,98%**	**13,03%**	**13,49%**

5.2.6.1 – Análise dos resultados para ensemble de classificadores

Através dos resultados obtidos podemos concluir que a combinação de classificadores obteve melhora no resultado apenas para o Corpus Jornal, evidenciando que a natureza dos textos e a composição da coleção (número de textos) influenciam no resultado. A redução obtida no Corpus Jornal foi de 6,2% (de 7,6 para 7,13%) na utilização do classificador 2 apenas com os termos selecionados pelo método coeficiente de correlação e de 9,21% (de 7,6% para 6,9%) para a

combinação dos termos pelos dois métodos de seleção escolhidos. Para o Corpus Teses a combinação obteve melhora, mas o classificador baseado em regras lógicas (classificador 2) apresentou uma redução de 18,46% (de 15,98% para 13,03%) comparado ao resultado obtido pelo classificador Naive Bayes, que já era o melhor resultado obtido na comparação entre os dois classificadores (Bayes e SVM) realizada nesta dissertação, e 3,41% (de 13,49% para 13,03%) se comparado ao resultado do classificador de regras lógicas (classificador 3).

5.3 – Considerações

A combinação de classificadores é uma das opções para a obtenção de melhores resultados na classificação. Nesta dissertação o resultado obtido comprova esta afirmação quando obtém, na melhor hipótese, uma diminuição do erro em 9,21%, valor bastante próximo ao obtido no trabalho de KALVA(2005). Ainda há melhorias a serem realizadas nas regras de combinação dos classificadores pois a idéia era avaliar a melhora obtida com o menor acréscimo de processamento possível. Estas novas regras devem ser avaliadas principalmente no conjunto de textos científicos pois foi nesse corpus que os resultados não comprovaram a literatura.

Trabalho semelhante deve ser desenvolvido com o classificador baseado em regras lógicas que apresenta melhores resultados nos textos científicos. Parte desse bom desempenho pode ser explicado pelo fato deste classificador utilizar vetores locais maiores que aqueles adotados na classificação por termos. Mas avaliações mais acuradas devem ser realizadas no sentido de otimizar este classificador de modo a obterem-se bons resultados com vetores locais de tamanho semelhantes aos adotados na classificação por termos. Vetores locais maiores representam uma abrangência maior de características específicas de cada categoria.

Capítulo 6 – Conclusões e trabalhos futuros

6.1 - Conclusões

Este trabalho apresenta um estudo sobre a aplicação de diferentes técnicas para a seleção de termos, pré-processamento e de classificação para categorização de textos em português. Este estudo visa a elencar as possíveis técnicas para minimização do erro obtido na classificação de textos, isto porque, face a já consagrada posição do Google como mecanismo de busca na Internet, o foco das pesquisas tende a voltar-se para a limitação do espaço de busca através da colocação das informações em classes pré-estabelecidas.

Para a realização do estudo foram utilizadas duas coleções de textos, uma do corpus NILC, contendo 855 textos jornalísticos das seções Esporte, Imóveis, Informática, Política e Turismo do Jornal Folha de São Paulo do ano de 1994. E outra composta pelos títulos e resumos das teses de Mestrado e Doutorado em Engenharia Elétrica da COPPE/UFRJ, divididas nas categorias Controle, Microeletrônica, Processamento de sinais, Redes de computadores e Sistemas de potência. Os corpus foram divididos em três conjuntos distintos de treinamento e teste.

Para pré-processamento foram utilizadas as informações lingüísticas dos textos. Para isso todos os documentos foram submetidos a um analisador sintático denominado PALAVRAS, a fim de obter-se a análise sintática dos mesmos. Para em seguida, com base nas marcações do analisador, serem submetidos a ferramenta XTRACTOR que gera três arquivos de saída para cada um de entrada, o primeiro com as palavras do texto denominado *words*, o segundo com as informações morfossintáticas das palavras do texto denominado POS (*part-of-speech*) e o terceiro com as estruturas de combinação entre as palavras denominado *chunks*. De posse destes arquivos utilizam-se folhas de estilo XSL para obtenção das estruturas morfossintáticas substantivo, verbo, adjetivo e nome próprio, e das estruturas sintáticas sujeito, verbo e complementos (objeto, predicativo e agente da passiva). As folhas de estilo implementadas extraíram dos textos as seguintes combinações: substantivo, substantivo - adjetivo, substantivo - nome próprio, substantivo - verbo, substantivo - adjetivo - nome próprio, substantivo - verbo - adjetivo, nome próprio – adjetivo e verbo. E as estruturas sintáticas sujeito, verbo e objeto que foram combinadas gerando os pares sujeito – verbo, verbo – complementos.

Após a extração destas estruturas os termos selecionados foram submetidos à preparação dos dados e então submetidos aos algoritmos de classificação. A

preparação dos dados foi baseada nos cálculos de relevância freqüência do documento, coeficiente de correlação e escore de relevância para estabelecerem-se quais termos representariam cada classe, sua seleção foi realizada através da truncagem (escolha dos n termos mais relevantes) das listas de termos e sua codificação vetorial através do cálculo da freqüência relativa. A codificação escolhida para representação de cada texto foi a codificação vetorial que consiste no estabelecimento de pesos referentes a cada termo presente no vetor local de cada categoria caso os termos ocorram no texto que se deseja representar. Os classificadores utilizados foram o Naive Bayes, o Support Vector Machines e Classificadores baseados em regras lógicas. Para todos os experimentos foi realizada a categorização binária dos exemplos, ou seja, cada exemplo poderia pertencer a apenas uma categoria e a escolha da vencedora se deu pelo maior escore obtido no classificador. Para cada exemplo, um vetor global composto por 30, 60, 90, 120 e 150 termos foi montado através da codificação por freqüência relativa dos termos nos documentos, correspondendo aos 6, 12, 18, 24 e 30 termos mais relevantes para representação de cada classe.

O primeiro experimento consistiu da classificação de textos representados pelos termos e suas combinações, através dos três métodos de seleção de características (FD, CC e ER), e dos dois classificadores mencionados. Os resultados foram para o Corpus Jornal de 7,49% de erro para a combinação substantivo – nome próprio – adjetivo e para o Corpus Teses de 15,98% para a combinação substantivo – verbo – adjetivo para Naive Bayes, e 12,51% para o Corpus Jornal e 20,78% para o Corpus Teses para a combinação substantivo – verbo – adjetivo para o classificador SVM. Esta diferença é devida a discrepância entre os estilos dos dois Corpus. Estes valores confirmam a performance obtida pelo classificador Naive Bayes em experimentos anteriores da literatura mas destacam o bom resultado do SVM.

O segundo experimento foi realizado com os pares formados pelos sujeito e verbo, e verbo e complementos das sentenças dos textos. Os resultados obtidos não recomendam estas estruturas para classificação. Os resultados ficaram em 52% de erro médio. A idéia era aumentar o conteúdo semântico dos termos representativos dos textos.

O terceiro experimento consistiu da fusão entre o primeiro e o segundo, ou seja, a utilização tanto dos termos quanto dos pares para a classificação dos textos. Os resultados demonstram que quando a freqüência dos termos é muito maior que a dos pares o resultado é pior, mas quando a freqüência é semelhante, ocorre ganho na performance do classificador aliada ao ganho semântico no processo de categorização.

O quarto experimento consistiu da aplicação da técnica de semântica latente, oriunda da decomposição por valor unitário, na matriz representada pelos vetores dos documentos. Este procedimento visou a eliminação de termos pouco relevantes que estivessem sendo considerados como dimensão dos vetores no processo de classificação. Dos resultados depreende-se que esta técnica não apresenta bons resultados quando utilizando dimensões reduzidas, no caso os 150 termos representativos de cada vetor. Esta técnica originalmente é utilizada para extração de características dos textos de uma coleção. No entanto, pode-se perceber que o classificador SVM apresenta melhor comportamento quando da utilização desta técnica.

O último experimento consistiu da construção de classificadores baseados em regras lógicas combinados com os classificadores que melhores resultados apresentaram na categorização por termos. Dos resultados obtidos, para o Corpus Jornal 6,9% e para o Corpus Teses 13,03%, pode-se concluir que a combinação de classificadores não apresenta a mesma performance para diferentes coleções de textos, devendo ser aprofundada a pesquisa para descoberta das causas desta diferença. Para o Corpus Jornal o resultado obtido confirma os resultados da literatura onde a combinação apresenta melhores resultados que os classificadores separados. Mas o Corpus Teses não confirma este resultado pois um dos classificadores, baseado em regras lógicas, apresentou melhor resultado que a combinação.

Conclui-se que o trabalho atingiu o fim a que se propôs e ainda apresentou resultados excelentes quando consideradas as obras existentes na literatura. Pode-se afirmar que o desenvolvimento de ferramentas de classificação automática de textos deve ser baseado no Classificador Naive Bayes aliado a métodos de seleção dos termos que levem em consideração a ocorrência dos termos do vetor global tanto nas classes como nas não-classes. No entanto, não se pode prescindir da análise sintática dos textos para utilização destas informações na seleção dos termos como demonstrado no trabalho de SILVA (2004) e comprovado nesta dissertação.

6.2 – Trabalhos Futuros

Como trabalho futuro pretende-se:

- melhorar a construção dos pares sintáticos de modo a avaliar possíveis melhorias nos resultados apresentados;
- desenvolver novas regras de decisão mais elaboradas de modo a corrigir o comportamento apresentado, na combinação de classificadores, pelo Corpus Teses;
- verificar a conseqüência destas novas regras na classificação do Corpus Jornal;

- melhorar os textos do Corpus Teses com a inserção de novos textos tornando a coleção mais semelhante ao Corpus Jornal;
- buscar a relação entre as duas coleções de textos evidenciando o comportamento dos classificadores para cada uma delas; e
- desenvolver algoritmos próprios de classificação de modo a obter o melhor resultado possível com a otimização do classificador Naive Bayes.

Em paralelo, pretende-se desenvolver aplicações que utilizem a compartimentação da informação na segurança de empresas que dependem deste procedimento para realização de suas funções, principalmente aquelas envolvidas com desenvolvimento de tecnologia militar.

Outra aplicação possível seria a combinação da classificação automática de textos com programas que transformam a voz em texto de modo a classificar diálogos como suspeitos ou não, e utilizar esta informação no reconhecimento de ilícitos na região de cobertura do SIVAM, Sistema Integrado de Vigilância da Amazônia.

7 – Referências Bibliográficas

APTÈ, C., DAMERAU, F., WEISS, S., 1994, "Automated Learning of Decision Rules for Text Categorization". In: *ACM Transactions on Information Systems*, v. 12(3), pp. 233-251, July.

APTÈ, C., DAMERAU, F., WEISS, S., 1994, "Towards Language Independent Learning of text Categorization Models". In: *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, Dublin, pp. 23-30, July.

BAEZA-YATES, R., RIBEIRO-NETO, B., 1999, *Modern Information Retrieval*. 1 ed., United Kingdom, Addison-Wesley.

BEALE, R., JACKSON, T., 1990, *Neural Computing: An Introduction*. 3 ed., London, Institute of Physics Publishing.

BERNARDINI, F., 2002, *Combinação de classificadores simbólicos para melhorar o poder preditivo e descritivo de Ensembles*. Dissertação de M.Sc. Instituto de Ciências Matemáticas e de Computação – ICMC/USP, São Carlos.

BICK, E., 2000, *The parsing System PALAVRAS: Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. 1 ed., Arhus University, Denmark, Arhus University Press.

BRAGA, A. P., LUDERMIR, T. B., CARVALHO, A. C. P. L. F., 2000, *Redes Neurais Artificiais: Teoria e aplicações*. Rio de Janeiro: LTC - Livros Técnicos e Científicos Editora S.A.

CHUTE, C., YANG, Y., 1994, "An example-based mapping method for text categorization and retrieval". In: *ACM Transaction on Information Systems*, v.12(3), pp.252-277, New York, July.

COOPER, W.S., CHEN, A., GEY, F.C. , 1994, "Full text retrieval based on probabilistic equations with coefficients fitted by logistic regression". In: *The Second Text Retrieval Conference (TREC-2)*, pp. 57-66, Washington DC, March.

CORREA, R. F., 2002, *Categorização de textos utilizando redes neurais – análise comparativa com técnicas não-conexionistas.* Dissertação de M.Sc., Universidade Federal de Pernambuco, Pernambuco.

DRUCKER, H., WU, D., VAPNIK, V., 1999, "Support Vector Machines for spam categorization". In: *IEEE Transactions on Neural Networks*, v 10(5), pp.1048-1054, september.

DUDA, R.O.; HART, P.E.; STORK, D.G., 2000, *Pattern Classification*, 2 Ed., California, Wiley Interscience.

EFRON, M., 2003, *Eigenvalue-based Estimator for Optimal Dimensionality Reduction in Information Retrieval*, Tese de D.Sc., School of Information and Library Science, University of North Carolina at Chapel Hill, North Carolina.

FRAKES, W. B., 1992, “Stemming Algorithms”. In: FRAKES, W. B., BAEZA-YATES, R., *Information Retrieval: Data structures & algorithms*. 1 ed., Capítulo 3, New Jersey, Prentice Hall PTR.

GASPERIN, C., VIEIRA, R., GOULART, R., QUARESMA, P., 2003, “Extracting XML Syntactic Chunks from Portuguese Corpora”. In: *Proceedings of the Workshop TALN 2003 Natural Language Processing of Minority Languages and Small Languages,* vol. 2, Batz-sur-Mer, France, June.

GEMAN, S., BIENENSTOCK, E. DOURSAT, R., 1992, “Neural Networks and the bias/variance dilemma”. *Neural Computation*, v. 4(1), pp.1-58.

GROBELNIK, M., MLADENIC, D., 1998, “Efficient text categorization”. In: *Proceedings of 10th European Conference on Machine Learning ECML98*, Springer, pp. 95-100, Berlin.

HANSEN, L., SALAMON, P., 1990, “Neural networks ensembles”. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* v.12(10), pp. 993-1001.

HARMAN, D., 1991, “How effective is suffixing”. *Journal of the American Society for Information Science* v. 42(1), pp.7-15.

HAYKIN, S., 2000, *Redes Neurais – Princípios e Práticas,* Brasil, 2 ed., Ed. Bookman.

JACKSON, P., MOULINIER, I., 2002, *Natural Language processing for Online Applications – Text retrieval, Extraction and Categorization*. Philadelphia: John Benjamins B.V.

JANG, J. R., SUN, C., 1995, “Neuro-Fuzzy Modeling and Control”, In: *Proceedings of the IEEE*, v. 83, pp. 378-406, March.

JOACHIMS, T., 1998, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features”. In: *Proceedings of European Conference on Machine Learning (ECML)*, pp. 137-142, Berlin.

JOHN, G. H., LANGLEY, P., 1995, “Estimating Continuous Distributions in Bayesian Classifiers”. In: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence,* vol 1, pp. 338-345, San Mateo.

KALVA, P. R., 2005, *Classificação de imagens usando combinação de classificadores e informações contextuais*. Dissertação de M.Sc., Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná, Curitiba.

KOHONEN, T., 1997, *Self-Organizing Maps*. 2 ed., Berlim, Germany, Springer.

KRAAJI, W., POHLMANN, R., 1996, “Viewing stemming as recall enhancement”. In: *Proceedings of SIGIR-96, 19th International Conference on Research and Development in Information Retrieval*, pp. 40-48, Switzerland, August.

LARKEY, L. S., CROFT, W. B., 1996, “Combining classifiers in text categorization”. In: *Proceedings of SIGIR-96, 19th International Conference on Research and Development in Information Retrieval*, pp. 289-297, Switzerland, August.

LENNON, M., PIERCE, D., TARRY, B., WILLETT, P., 1981, “An evaluation of some conflation algorithms for information retrieval”. *Journal of Information Science,* n.3, pp.177-183.

LEWIS, D. D., 1992, *Representation and Learning in Information Retrieval*. PhD Thesis. Department of Computer and Information Science, University of Massachusetts, Massachusetts.

LEWIS, D.D. & RINGUETTE, M., 1994, “A Comparison of Two Learning Algorithms for Text Categorization”. In: *Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval*, vol.1, pp. 81-93, Las Vegas, USA, April.

LOH, S., 2001, *Abordagem baseada em Conceitos para Descoberta de Conhecimento em Textos*. Requisito Parcial ao Grau de Doutor em Ciência da Computação, Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.

LOSEE, R. M., 1998, *Text retrieval and filtering: Analytic Models of Performance*. Massachusetts. Kluwer Academic Publishers.

MARON, M.E., 1961, “Automatic Indexing: An Experimental Inquiry”. *Journal of ACM*, v. 8, pp. 404-417.

McCALLUM, A. K., NIGAM, K., RENNIE, J., SEYMORE, K., 1999, “Building Domain-Specific Search Engines with Machine Learning Techniques”. In: *AAAI-99 Spring Symposium on Intelligent Agents in Cyberspace*.

McCALLUM, A. and NIGAM, K., 1998, “A comparison of event models for naive bayes text classification”. In: *AAAI/ICML-98 Workshop on Learning for Text Categorization*, Technical Report WS-98-05, AAAI Press, pp. 41-48, Madison, USA.

MITCHELL, T. M., 1997, *Machine Learning*. USA, Ed. McGraw-Hill.

MLADENIC, D. & GROBELNIK, M., 1998, “Feature Selection for Classification Based on Text Hierarchy”. In: *Working Notes of Learning from Text and the Web, Conf. Automated Learning and Discovery (CONALD-98),* Pittsburgh.

MOENS, M. F., 2000, *Automatic indexing and abstract of document texts*. Massachusetts, Kluwer Academic Publishers.

NEVES, M. L., 2001. *PubsFinder - um Agente Inteligente para Busca e Classificação de Páginas de Publicações*. Dissertação de M.Sc. Centro de Informática da UFPE, Recife.

NG, H. T., GOH, W. B. & LOW, K. L., 1997, “Feature Selection, Perceptron learning and a Usability Case Study for Text Categorization”. In: *Proceedings of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval*, pp. 67-73, Philadelphia, USA.

OLIVEIRA, H. M., 1996. *Seleção de Entes Complexos utilizando lógica difusa*. Dissertação de M.Sc., Instituto de Informática, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre.

PEREZ, C. & VIEIRA, R., 2005, “Mapas Conceituais: geração e avaliação”. In*: TIL - Workshop de Tecnologias da Informação e Linguagem Humana. Anais do XXV Congresso da SBC*, v.1, pp. 2158-2167, São Leopoldo.

PRECHELT, L., 1994, “Proben1 - A Set of Neural Network Benchmark Problems and Benchmarking Rules”. In: *Technical Report 21/94*.

QUINLAN, J. R., 1993, *C4.5: Programs for Machine Learning***.** San Mateo, Morgan Kufmann Publishers.

QUINLAN, J. R., 1986, “Induction of Decision Trees”. In: *Readings in Knowledge Acquisition and Learning*, Bruce G. Buchanan & David C. Wilkins, Morgan Kaufmann, pp. 349-361.

RIJSBERGEN, C. J., 1979, *Information Retrieval*. 2 ed., Ed. Department of Computer Science, University of Glasgow, United Kingdom.

RILOFF, E., 1995, “Little words can make big difference for text Classification”. In: *Proceedings of SIGIR-95, 102 18th ACM International Conference on Research and Development in Information Retrieval*, pp. 130-136, Seattle, US.

RIZZI, C. B., WIVES, L. K., OLIVEIRA, J. P. M., ENGEL, P.M. , 2000, “Fazendo uso da Categorização de Textos em Atividades Empresariais”, *International Symposium on Knowledge Management/Document Management - ISKDM/DM 2000*, pp. 251-268, Curitiba, Brasil.

RIZZI, C. B., VALIATI, J. F., ENGEL, P. M., 2001, “Uma Proposta para Categorização de Textos por uma Rede Neural”. In: *Proceedings of V Congresso Brasileiro de Redes Neurais - VI Escola de Redes Neurais*, pp. 517-522, Rio de Janeiro, RJ, Brasil, Abril.

RIZZI, C. B., 1999. *Um estudo sobre recuperação de informações não estruturadas*. Trabalho individual, Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.

ROCCHIO, J. J., 1971, *Relevance Feedback in Information Retrieval*. In: G. Salton, editor. *The Smart Retrieval Systems: Experiments in Automatic Document Retrieval*, Engelwood Cliffs, pp. 313-323, New Jersey. Ed. Prentice-Hall.

RUMELHART, D. E.; McCLELLAND, J. L., 1986, *Parallel Distributed Processing, volume 1*: *Foundations*. USA, The MIT Press.

SALTON, G. & BUCKLEY, C., 1987, “Term Weighting Approaches in Automatic Text Retrieval*”*. In: *Technical Report TR87/881*, New York.

SALTON, G. & BUCKLEY, C., 1988, “Term Weighting Approaches in Automatic Text Retrieval”. In: *Information Processing and Management*, *v. 24, n. 5*, pp. 513-523.

SALTON, G., 1983, *Introduction to Modern Information Retrieval*. New York, Ed. McGraw-Hill.

SCHNEIDER, K., 2003, “A comparison of event models for Naive Bayes Anti-Spam E-mail filtering*”*. In: *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics,* Passau.

SCHUTZE, H. & SILVERSTEIN, C., 1997, "Projections for efficient document clustering", In: *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pp.74–81, Philadelphia, USA.

SEBASTIANI, F., 1999, "A Tutorial on Automated Text Categorization". In: *Proceedings of ASAI-99, 1st Argentinian Symposium on Artificial Intelligence*, pp.7-35, Buenos Aires, AR.

SILVA, C. F., 2004. *Uso de Informações Lingüísticas na etapa de pré-processamento em Mineração de Textos*. Dissertação de M.Sc. Universidade do Vale do Rio dos Sinos, Programa Interdisciplinar de pós-graduação em Computação aplicada, São Leopoldo – Rio Grande do Sul.

STEINBERG, D. & COLLA, P., 1995, *CART: Tree-Structured Non-Parametric Data Analysis*. San Diego, CA, Ed. Salford Systems.

VAPNIK, V. N., 1995, *The Nature of Statistical Learning Theory*. New York, Ed. Springer.

WIENER, E., PEDERSEN, L.O., WEIGEND, A.S., 1995, "A Neural Network Approach to Topic Spotting". In: *Proceedings of the Symposium on Document Analysis and Information Retrieval*, pp.317-332, Las Vegas, US.

WITTEN, I.H., 2000, *Data mining: Pratical machine Learning Tools and techniques with Java implementations*. New Zealand, Ed. Academic Press.

WIVES, L. K., 1999, *Um estudo sobre Agrupamento de documentos textuais em Processamento de Informações não estruturadas usando técnicas de Clustering*. Dissertação de M.Sc., Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.

YANG, Y. & LIU, X., 1999, "A Re-examination of Text Categorization Methods". In: *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pp. 44-49, Berkley, US.

YANG, K., 1997, *Combining Multiple Document Representations and Multiple Relevance Feedback Methods to Improve Retrieval Performance*. Master's Thesis, School of Information and Library Science. University of North Carolina at Chapel Hill, USA.

YANG, Y., 1999, "An Evaluation of Statistical Approaches to Text Categorization". *Journal of Information Retrieval*, v. 1, n. 1/2, pp. 67-88.

YANG, Y., 1995, "Noise reduction in a Statistical Approach to Text Categorization". In: *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Seattle, Washington, USA, July.

YANG, Y. & PEDERSEN, J.O., 1997, "A Comparative Study on Feature Selection in Text Categorization". In: *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97)*, pp. 412-420, San Francisco, USA.

YIN, L. L., 1996, *Learned Text Categorization by Backppropagation Neural Network*. Master's Thesis. The Hong Kong University of Science and Technology, Hong Kong.