Ronald Carvalho Ribeiro de Araújo

Um estudo sobre técnicas de Representação Estruturada de Textos

Feira de Santana – BA

Fevereiro - 2009

Ronald Carvalho Ribeiro de Araújo

Um estudo sobre técnicas de Representação Estruturada de Textos

Trabalho de Conclusão de Curso apresentado à Banca de Graduação em Engenharia de Computação da Universidade Estadual de Feira de Santana para a obtenção do título de Bacharel em Engenharia de Computação

Orientador:
Prof. Msc. Angelo Conrado Loula

Curso de Engenharia de Computação
Departamento de Ciências Exatas
Universidade Estadual de Feira de Santana

Feira de Santana – BA

Fevereiro - 2009

Monografia de Trabalho de Conclusão de Curso sob o título *"Um estudo sobre técnicas de Representação Estruturada de Textos"*, defendida por Ronald Carvalho Ribeiro de Araújo , em Feira de Santana, Estado da Bahia, pela banca examinadora constituída pelos professores:

Prof. Msc. Angelo Conrado Loula
Orientador
Universidade Estadual de Feira de Santaa

Profa. Dra. Fabiana Cristina Bertoni
Universidade Estadual de Feira de Santana

Prof. Msc. Matheus Giovanni Pires
Universidade Estadual de Feira de Santana

Dedico esta monografia a toda turma de 2004.1 do curso de Engenharia de Computação da Universidade Estadual de Feira de Santana.

Resumo

Este trabalho visa explorar a etapa de pré-processamento em um processo de Mineração de Texto. Serão discutidas as técnicas envolvidas visando apresentar um panorama da relevância destas técnicas para o resultado final de um processo de Mineração de Texto. É apresentado um levantamento bibliográfico no que tange a fase de pré-processamento. Num segundo momento é apresentada a metodologia utilizada para a elaboração e execução dos experimentos realizados. A fim de analisar os resultados do pré-processamento são utilizados dois algoritmos de classificação, sendo eles *K-Nearest Neighbors* e *Naive Bayes*. Por fim, são apresentados os resultados obtidos e a discussão a cerca dos mesmos.

Abstract

This work aims to explore the pre-processing step in a process of Text Mining. Will discuss the techniques involved to give an overview of the relevance of these techniques to the final outcome of a process of Text Mining. Bibliography is provided with respect to the pre-processing. Second is presented the methodology used for the preparation and execution of experiments made. In order to analyze the results of the pre-processing are two algorithms used for classification, they are Naive Bayes and K-Nearest Neighbors. Finally, the results obtained and discussion about them are present.

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

A informação é fator determinante no processo de tomada de decisões. Com o advento da globalização, o fluxo de dados vem crescendo de forma exponencial, promovendo um ambiente onde se faz necessário uma análise automática e refinada de todo este conjunto de dados, a fim de convertê-los em informação de qualidade, ou seja, informação capaz de produzir conhecimento.

Documentos de texto são a forma mais comum utilizada pelo homem para armazenamento de dados. Milhares de livros, manuscritos, emails, relatórios, contratos, prontuários e documentos textuais são gerados diariamente, porém, tais documentos comumente não apresentam uma forma de representação estruturada, são apenas textos livres, não apresentam um padrão, tornando complexo o processo de extração de informação a partir de tais dados. Nesta conjuntura, surge a Mineração de Texto (*Text Mining*), com o intuito de auxiliar no processo de extração de informação de qualidade a partir de textos não estruturados.

Segundo (SULLIVAN, 2000), Mineração de Texto pode ser definida como o estudo e prática de extração de informações de textos utilizando princípios da lingüística computacional. De acordo com (TAN, 1999), Mineração de Texto é uma área que possui grande potencial comercial, uma vez que a maior parte das informações de empresas está documentada em arquivos de texto. Assim, um processo de extração automática de conhecimento possui grande valor dentro das organizações coorporativas, uma vez que possibilita o acesso rápido a informação, oferecendo subsídio para a tomada de decisões.

Tipicamente um processo de Mineração de Texto é composto pelas fases de Pré-Processamento e Extração de Conhecimento. A Figura 1 ilustra as fases de um processo de Mineração de Texto.

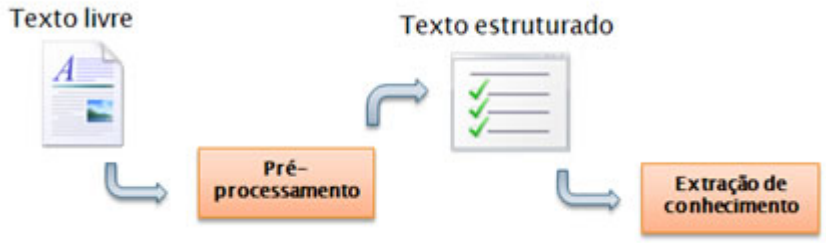

Figura 1: Fases de um processo de Mineração de Texto.

Pré-Processamento ou Refinamento do Texto é uma etapa fundamental no processo de Mineração de Texto (GOLDSCHIMIT, 2005). Na fase de Pré-Processamento ocorre a transformação do texto de forma livre, não estruturado, em uma representação estruturada. A partir desta representação são aplicadas técnicas de obtenção de conhecimento como agrupamento e classificação, técnicas que constituem a fase de Extração de Conhecimento.

As aplicações envolvendo Mineração de Texto são as mais diversas, alguns exemplos são: Inteligência de Negócios (*Business Intelligence*) (SPINAKIS, 2004); melhoria de processos de Engenharia de Software, com a análise de relatórios que descrevem o comportamento de determinado processo (HAYES, 2005); Biomedicina, com análise da literatura científica (ERHARDT, 2006); jurisprudências, com levantamento de decisões de juízes de direito (BEPPLER, 2005); categorização automática de mensagens (QUI, 2008); dentre outras.

A principal motivação deste trabalho é a busca por maiores esclarecimento a cerca da eficiencia das técnicas utilizadas na fase de pré-processamento em mineração de texto. Comumente a literatura aponta para o uso da técnica de TF-IDF, porém não se encontram justificativas plausiveis para esta escolha desta técnica. Assim, o presente trabalho objetiva o estudo, análise e comparação das técnicas existentes para a fase de Pré-Processamento em Mineração de Texto, utilizando como cenário de avaliação um problema de classificação. Serão abordados aspectos inerentes ao processo de Mineração de Texto, com ênfase na fase de Pré-Processamento. Será apresentado o modelo para representação estruturada de um documento de texto, técnicas para o balanceamento de relevância das palavras dentro de um documento, testes com variações na etapa de Pré-Processamento, além dos resultados obtidos e a discussão dos mesmos.

No Capítulo 2 é realizado um levantamento bibliográfico a cerca de temas que envolvem o processo de Mineração de Texto, com ênfase na fase de Pré-Processamento.

O Capítulo 3 aborda a metodologia adotada na condução dos experimentos realizados

neste trabalho.

No Capítulo 4 são apresentados os resultados encontrados nos experimentos realizados, bem como a discussão sobre os mesmos.

O Capítulo 5 relata os objetivos alcançados pelo trabalho, um resumo da discussão dos resultados e os possíveis trabalhos futuros.

2 Fundamentação Teórica

Nesta seção será exposto um levantamento sobre o estado da arte no que diz respeito à etapa de Pré-Processamento num processo de Mineração de Texto. Será discutida a importância do Pré-Processamento, o modelo de representação dos documentos de texto livre em um modelo estruturado, os modelos de distribuição da importância das palavras do texto numa representação estruturada, técnicas de extração de conhecimento a partir de uma representação estruturada, além de medidas de avaliação de qualidade para o processo de Mineração de Texto.

2.1 Pré-Processamento de Texto

Em um processo de mineração de texto, o pré-processamento é a etapa responsável por converter um documento de texto na forma livre, não estruturada, em um modelo de representação estruturada, formato adequado para aplicação de técnicas de extração de informação como classificação e clusterização. Uma abordagem usual para a etapa de pré-processamento é a *bag-of-words*, onde cada documento é representado pelo conjunto de palavras contidas na coleção de textos, formando uma matriz esparsa, onde cada coluna representa uma palavra da coleção de documentos (MATSUBARA, 2003).

Um dos grandes desafios da etapa de pré-processamento é minimizar a perda de informações inerentes ao processo de transformação de texto livre em modelo estruturado, uma vez que a representação estruturada é uma visão mais simplificada do documento original. A justificativa para esta simplificação é reduzir a quantidade de palavras associadas a cada documento, atenuando assim o esforço computacional para manipular o resultado da representação.

2.2 Modelo Espaço Vetorial

O Modelo Espaço Vetorial é uma forma de representação estruturada onde cada documento é representado por um vetor de termos. A ideal central é construir um espaço T-dimensional, onde cada termo da coleção de documentos representa uma dimensão deste espaço. Um exemplo com um conjunto de apenas três termos, logo com espaço 3-dimensional, é ilustrado na Figura 2, onde cada item pertencente ao espaço é identificado por três componentes, uma correspondendo a cada dimensão. Assim, a posição de um documento no espaço T-dimensional é dada pela contribuição que cada termo exerce no vetor de termos. A idéia do espaço com três dimensões pode ser estendida formando assim o espaço T-dimensional.

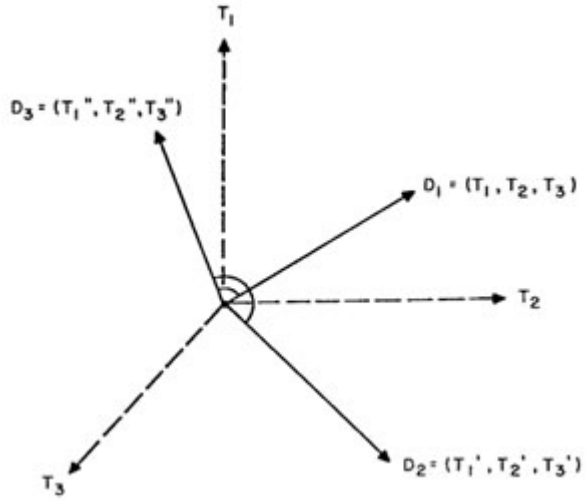

Figura 2: Representação de documentos num espaço com três dimensões (SALTON, 1997).

Sendo C uma coleção contendo N documentos, e T a quantidade de termos contidos nos N documentos, tem-se que cada documento é representado por um vetor Di com T posições, ou seja, cada documento é representado por um vetor, onde cada termo da coleção de documentos corresponde a uma posição neste vetor. A ocorrência de todos os termos da coleção em todos os vetores que representam os documentos é justificada pela necessidade de se ter vetores com igual quantidade de termos, ou seja, iguais dimensões no espaço T-dimensional, formando assim uma matriz em que as linhas correspondem aos vetores de cada documento e as colunas aos termos da coleção.

O vetor de termos indica em qual posição do espaço T-dimensional o documento se encontra, dessa forma, documentos similares tendem a estar próximos no espaço, enquanto documentos que apresentam pouca similaridade tendem a estar distantes. Esta

consideração é reafirmada por (SOUCY, 2005). A Figura 3 ilustra, em uma visão bidimensional, o posicionamento de documentos representados por espaço vetorial.

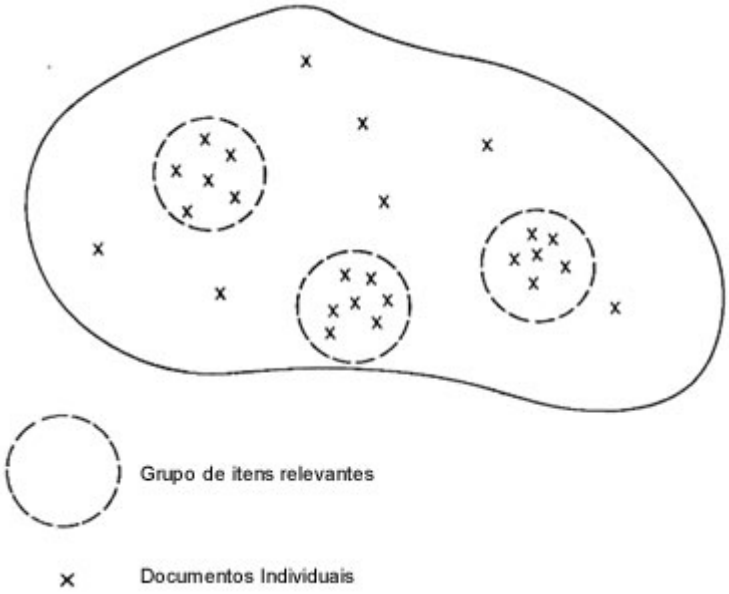

Figura 3: Visão bidimensional de documentos representados em espaço vetorial. Adaptado de (SALTON, 1997).

Desempenho é uma preocupação quando se trabalha com o Modelo de Espaço Vetorial. A alta dimensionalidade, igual à quantidade de termos, representa uma das desvantagens deste modelo. Em contraponto, a simplicidade e o bom comportamento para coleções de documentos genéricos são virtudes do modelo (NOGUEIRA, 2000).

Segundo (ARANHA, 2007), a técnica mais utilizada para representação de documentos é o modelo espaço vetorial, sendo aplicado geralmente em classificação automática de textos.

2.2.1 Distribuição de Pesos

No processo descrito por (SALTON, 1997), após a etapa inicial de construção dos vetores contendo todos os termos da coleção de documentos, inicia-se uma segunda etapa com objetivo de balancear o peso dos termos no vetor. O conceito é classificar os termos do vetor de acordo com sua relevância para o documento em questão, assim, o termo com maior relevância possui um peso maior que termos com baixa relevância (TAN, 1999). Para isto, aplicam-se algoritmos estatísticos que se baseiam na freqüência de ocorrência dos termos no documento. Um algoritmo bastante utilizado é o TFIDF, onde o peso do

termo é representado pela freqüência do mesmo no documento multiplicado por um valor discriminante, que mensura o quanto determinado termo é relevante para o documento (BEIL, 2002).

Ainda segundo (SALTON, 1997), após a classificação de relevância dos termos do vetor é conveniente um processo de normalização, a fim de equalizar os documentos que possuem uma grande quantidade de termos com os documentos que possuem uma menor quantidade. A normalização usualmente ocorre com objetivo de que o peso dos termos fique na faixa de valores entre zero e um.

A seguir são apresentadas as principais técnicas de balanceamento de pesos para a representação estruturada no modelo espaço vetorial.

2.2.1.1 Binário

A representação binária indica a presença ou não de determinado termo em um documento. Assim, se a palavra estiver presente no documento a representação deste termo no vetor será 1, em contrapartida, se a palavra não estiver presente a representação será 0. A representação binária é bastante simples, se preocupa apenas em indicar se a palavra existe ou não no documento. A grande desvantagem deste modelo é que todas as palavras presentes no documento possuem o mesmo valor de relevância, não sendo possível identificar os termos mais relevantes para o documento.

2.2.1.2 TF

TF (*Term Frequency*) é a medida de freqüência com que o termo aparece no documento, ou seja, quantas ocorrências de um mesmo termo acontecem em um documento. Para um documento Di, o peso associado (Pij) a um termo Tj do vetor de termos é dado por:

$$Pij = TF(Tj, Di)$$

TF é uma técnica bastante simples e ingênua, uma vez que numa coleção em que documentos tratam de um mesmo assunto é grande a probabilidade dos documentos conterem as mesmas palavras em grande freqüência, assim, com este cenário, a aplicação de TF é prejudicada, reduzindo o êxito em determinar palavras discriminantes para os documentos em meio à coleção.

2.2.1.3 IDF

IDF (*Inverse Document Frequency*) é uma medida que visa priorizar os termos que aparecem em poucos documentos da coleção. A idéia é que as palavras que não se repetem em muitos documentos, têm um valor de discriminação elevado, portanto, são boas para identificar determinado documento, enquanto que palavras que aparecem em muitos documentos possuem baixo grau de discriminação (KAO, 2005). Sendo N o número total de documentos da coleção e DTj o número de documentos em que determinado termo Tj ocorre, o valor de IDF de um termo Tj é dado por:

$$IDFj = log\frac{N}{DTj}$$

A utilização de logaritmo se dá com o intuito de evitar super ponderações de termos muito raros, e sub-ponderações de termos mais freqüentes. A variação na base do logaritmo constitui um importante ponto de análise na construção da técnica de IDF.

2.2.1.4 TFIDF

Esta é a técnica mais difundida para balanceamento de pesos em problemas que utilizam o modelo espaço vetorial. TFIDF é a junção das duas idéias anteriormente mencionadas, ou seja, observar a freqüência com que o termo aparece no documento e observar a freqüência com que este mesmo termo ocorre em diferentes documentos (RAMOS, 2003). Neste pensamento a medida IDF atua como um atenuador para a medida de TF, uma vez que IDF é calculado com utilização de uma função logarítmica. Alterações na base do lagaritmo obviamente provocarão alterações nos valores de IDF econtrados. A expressão para o cálculo de TFIDF é:

$$TFIDF(Tj, Di) = TF(Tj, Di) \times log\frac{N}{DTj}$$

2.2.1.5 Normalização

A aplicação de um padrão de normalização nas medidas de peso é uma prática bastante comum e objetiva evitar variações extremas e não necessariamente relevantes entre termos (dimensões) ou entre documentos. Pode-se colocar em iguais condições documentos que apresentam quantidades de termos distintos, ou seja, equiparam-se documentos que possuem uma grande quantidade de termos e documentos pequenos, com uma baixa quantidade de termos, evitando valores de TF muito grandes ou muito pequenos devido

ao tamanho do texto. Neste caso, divide-se o valor do TF encontrado para cada termo de um documento pela quantidade de termos do documento.

De forma mais usual, o processo de normalização do vetor é realizado com a divisão de todos os pesos dos termos de cada dimensão pelo maior peso nesta dimensão dentre todos vetores, gerando assim valores de peso na faixa entre zero e um, para todas as dimensões. Por motivos lógicos, a aplicação da normalização não se aplica à representação binária. Já em representações como TF, IDF e TFIDF a normalização é bastante utilizada.

2.2.2 Redução de Dimensionalidade

A alta dimensionalidade é o grande problema da representação espaço vetorial quando se utiliza uma abordagem bag-of-words (NOGUEIRA, 2000), ou seja, cada termo da coleção de documentos acrescenta uma dimensão ao problema. De acordo com o Modelo Espaço Vetorial, um documento que pertence a uma coleção, é representado por um vetor que contém todos os termos existentes na coleção de documentos. Assim, a dimensionalidade do problema é dada de acordo com a quantidade de termos existentes na coleção, sendo a dimensão diretamente proporcional à quantidade de termos.

A literatura conta com diversas técnicas que tem por objetivo reduzir a dimensionalidade de um problema baseado no modelo espaço vetorial. Dentre estas técnicas, merece destaque: *Stemming*, remoção de prefixos e sufixos de palavras; *StopWords*, remoção de palavras que não contribuem semanticamente com o documento; Validação de palavras, que busca eliminar termos que não pertencem a língua em questão; Cortes de Luhn e Poda.

Na seqüência são discutidos com maiores detalhes as técnicas citadas.

2.2.2.1 Lei de Zipf e Cortes de Luhn

De acordo com (MATSUBARA, 2003), os Cortes de Luhn tem como idéia central o fato de buscar uma observação sobre a distribuição de freqüência das palavras, conforme a Lei de Zipf, servindo como critério de eliminação de palavras que pouco colaboram para a identificação do conteúdo semântico.

A Lei de Zipf diz que a freqüência de ocorrência de um evento está de alguma forma relacionada a uma função de ordenação, assim a freqüência da palavra é inversamente proporcional ao *ranking* da mesma, sua posição numa ordenação por freqüência. Assim, o passo inicial para um conjunto de documentos é buscar a freqüência de cada termo presente

na coleção, comumente *Term Frequency*. Em seguida, ordenam-se os termos de forma decrescente, tendo como parâmetro sua freqüência (GELBUKH, 2001). Os Cortes de Luhn definem limites de freqüência máxima e mínima para os termos, os que estiverem acima do limite máximo ou abaixo do limite mínimo devem ser desconsiderados (MARTINS, 2003). A idéia é que palavras muito freqüentes e palavras raras não contribuem significativamente para a representação do documento. Luhn afirma que as palavras mais relevantes para a representação encontram-se no meio dos limites máximo e mínimo. A Figura 4, mostra um gráfico com a Lei de Zipf e os Cortes de Luhn. O eixo f representa a frequência do termo e o eixo r a posição do termo no vetor.

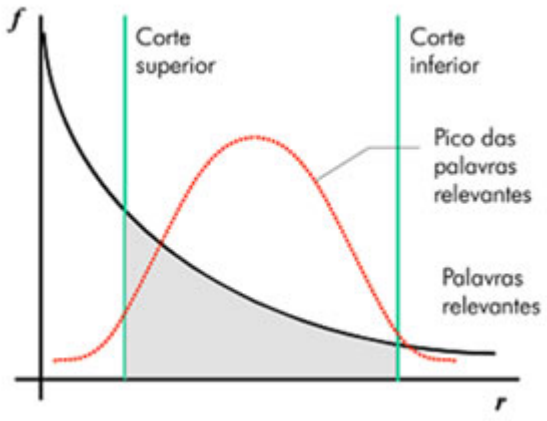

Figura 4: Lei de Zipf e Cortes de Luhn (MATSUBARA, 2003).

2.2.2.2 Validação de Palavras

Em documentos textuais não é incomum a ocorrência de erros de grafia. Palavras incorretas dificultam a análise do documento em uma abordagem estatística e automática, uma vez que, o termo incorreto contribui de forma equivocada na relação de freqüência do termo idealizado.

Uma alternativa que visa atenuar os erros de grafia em documentos de textos é o uso de um dicionário (*thesaurus*), a fim de validar se os termos pertencem à língua utilizada. O uso de um dicionário, além de contribuir para a redução da dimensionalidade do problema, através da eliminação de termos inválidos, é potencialmente útil para retificar palavras incorretas.

De acordo com (BERRY, 2004), o índice de um dicionário pode ser controlado, com possibilidade de adição e remoção de palavras pré-definidas, potencializando a identificação dos termos contidos nos documentos. Em (ARANHA, 2007), é apresentada a Figura 5,

onde se pode observar uma seqüência de caracteres. Os termos sublinhados não existem no dicionário, logo devem ser adicionados a ele ou rejeitados. Os termos grifados devem ser eliminados, pois não representam uma seqüência válida de caracteres para a língua trabalhada. Uma consideração importante é com relação às palavras especificas, como: nomes próprios, siglas e etc. Para estes casos os projetos devem definir algum procedimento a ser realizado, por exemplo: adicionar as palavras desejadas ao índice do dicionário.

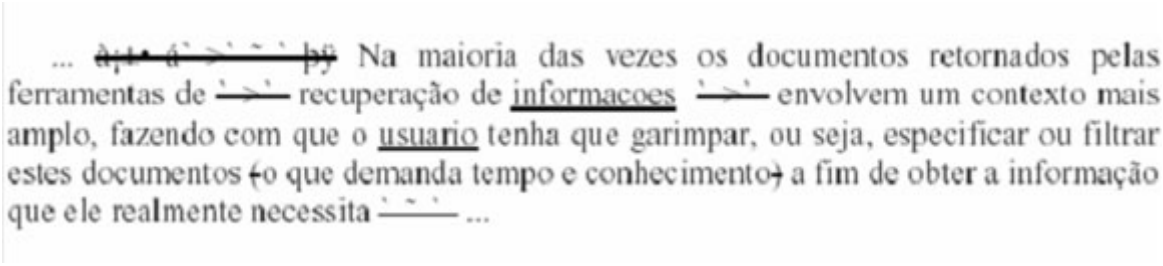
... Na maioria das vezes os documentos retornados pelas ferramentas de recuperação de informacoes envolvem um contexto mais amplo, fazendo com que o usuario tenha que garimpar, ou seja, especificar ou filtrar estes documentos o que demanda tempo e conhecimento a fim de obter a informação que ele realmente necessita ...

Figura 5: Identificação de palavras válidas (ARANHA, 2007).

Para a língua inglesa, uma gama de dicionários está presente na internet. Um exemplo é o projeto WordNet (WORDNET, 2004).

2.2.2.3 Sinônimos

A presença de sinônimos é bastante comum nos documentos textuais. Numa abordagem puramente estatística, sinônimos tendem a ser classificados como termos distintos, uma vez que a grafia é distinta, porém os sinônimos potencialmente podem ser representados por um único termo, elevando assim a análise semântica realizada no processo de representação do documento (BERRY, 2004).

O uso de sinônimos reduz a dimensionalidade do problema, com a redução do número de termos. A desvantagem desta abordagem é o sensível aumento do custo computacional, uma vez que cada termo gera uma nova consulta ao dicionário, além do fato de ser necessário manter uma estrutura de armazenamento para as palavras, a fim de identificar os sinônimos encontrados.

2.2.2.4 *Stop Words*

De modo geral, os idiomas possuem palavras que não agregam significado semântico nas orações, servindo estes apenas como auxiliares no processo lingüístico. Artigos, preposições, pronomes, advérbios são exemplos desta classe de palavras que não agregam

significado as orações. Como o objetivo de um processo de representação de texto é obter uma forma estruturada que represente semanticamente o documento, o grupo de palavras não-discriminantes, que não agregam significado, não deve estar presente na representação estruturada. Estas palavras não discriminantes recebem o nome de *StopWords*. A figura 6 mostra o exemplo de uma construção com as *StopWords* tachadas.

Segundo (BAEZA, 1999), a eliminação das *StopWords* reduz em média em 40% o tamanho da representação de um documento. Este dado reforça a afirmação que as *StopWords* devem ser eliminadas na etapa de pré-processamento de texto. Em diversos sites na internet é possível encontrar listas de *StopWords* para diversos idiomas. Para o inglês pode-se encontrar uma lista com mais de 600 *StopWords* em (ONIX, 2006). Em português uma lista pode ser encontrada em (WEB-MINING-FR, 2007).

De acordo com (ARANHA, 2007), o processo de construção de *StopWords* pode ocorrer de duas formas: montando-se manualmente a lista de palavras ou utilizando uma construção automática da lista de *StopWords*. Uma estratégia apresentada para construção automática é apresentada por (LO, 2005), onde se define um limite máximo de freqüência que uma palavra pode atingir, após ultrapassar este limite a palavra é adicionada a lista de *StopWords*.

Uma abordagem mista também pode ser utilizada no processo de construção da lista de *StopWords*, com a lista inicialmente sendo preenchida manualmente e posterior adição de palavras de acordo com a freqüência da mesma na coleção de termos.

~~The~~ proliferation ~~of~~ text databases ~~within~~ large organizations ~~and~~ ~~on~~ ~~the~~ Internet makes ~~it~~ difficult ~~for~~ ~~a~~ person ~~to~~ know ~~which~~ databases ~~to~~ search.

Figura 6: Trecho de um texto com destaque para as *StopWords*

2.2.2.5 *Stemming*

Plural, sufixo e flexões verbais são elementos que interferem no processo de análise estatística de um conjunto de palavras. Uma alternativa que visa atenuar estes problemas é a representação de palavras por seus *stems* (radicais). (STRZALKOWSKI, 1999) apresenta o conceito de *stem* como sendo o conjunto de palavras resultante de um processo de *stemming*, servindo como representação mínima não ambígua do termo.

Assim, o *stem* não necessariamente deve ser o termo que equivale à raiz da palavra. Em (KANTROWITZ, 2000) é apresentado o conceito de *Stemming* como sendo uma técnica de remoção de prefixos e sufixos de palavras, a fim de encontrar qual termo deve ser associado à ocorrência. Um exemplo é a representação de *played* e *playing*; pelo termo *play*.

O objetivo do uso de *stemming* é utilizar um único termo para representar as diversas variações de uma palavra, possibilitando assim uma redução do número de termos envolvidos num processo de representação estruturada de um documento. Segundo (GOMES, 2006), a técnica de *stemming* mais usual é a de Porter. A idéia é que sufixos são geralmente formados por sufixos ainda menores, de modo que, o algoritmo é constituído de cinco etapas, onde cada uma é composta de um conjunto de regras cujo objetivo é obter a menor representação não ambígua para o termo. A Figura 7 apresenta um diagrama que representa o algoritmo de Porter. Ainda de acordo com (GOMES, 2006) os cinco passos do algoritmo são compostos por regras como: Se um termo t possui mais do que s sílabas e termina com o sufixo SUFIX, o sufixo SUFIX é substituído por SUF.

Originalmente, o algoritmo de Porter foi desenvolvido para a língua inglesa e a implementação deste algoritmo em diversas linguagens de programação pode ser encontrado em (TARTARUS, 2004). Um aplicação online para *stemming* que implementa o algoritmo de Porter é disponibilizada em (MOBASHER, 2004).

A técnica de *stemming* de Porter é a mais difundida e utilizada em processos de Mineração de Textos (HOOPER, 2008). O algoritmo se baseia no prévio conhecimento morfológico da língua em questão e busca obter o menor radical que possa representar a palavra. O processo de *stemming* de Porter é dividido em fases, cinco ou seis, dependendo da implementação, neste projeto foi utilizada a implementação em seis fases. O algoritmo busca reconhecer *stems* que possuem o formato [Consoante](Vogal + Consoante)m[Vogal], onde m representa a quantidade de repetições da expressão entre parenteses e a letra y precedida de consoante, também é considerada vogal.

A fase inicial do algoritmo busca reduzir palavras no plural para o singular, além de remover as terminações "ed"e "ing". Na segunda fase, nas palavras que terminam com a letra y, a mesma é substituída por i, caso a palavras apresente mais de uma vogal. As fases seguintes tratam de identificar se o termo resultante das etapas anteriores constitui um *stem* válido, ou seja, obedece ao formato estabelecido pelo algoritmo. A figura 7 ilustra os passos do algoritmo de Porter.

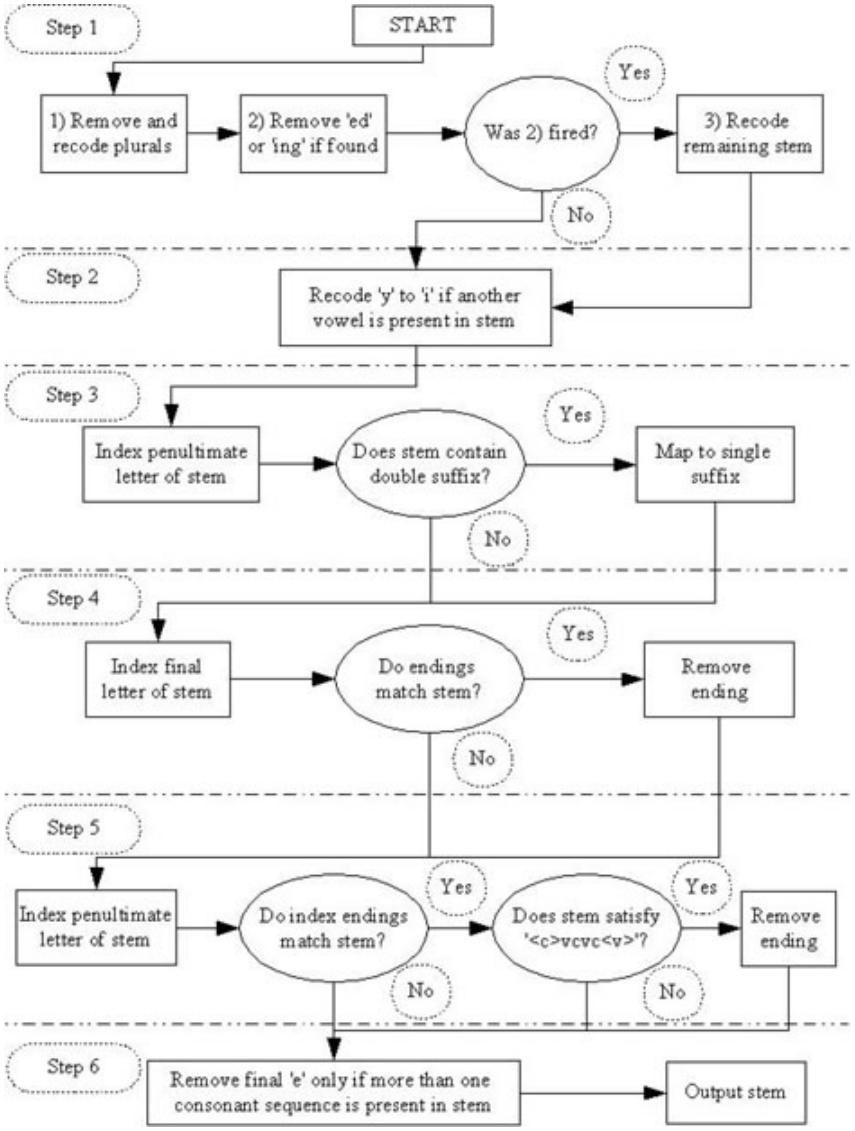

Figura 7: Fluxo de execução do algoritmo de Porter (HOOPER, 2008).

Segundo (KROVETZ, 2000), apesar do algoritmo de Porter ser muito utilizado, ele apresenta algumas limitações. Uma dessas limitações é o fato do algoritmo não restringir a produção de *stems*, dessa forma, palavras com significados totalmente diferentes podem ser reduzidos a um mesmo radical, como exemplo: *generate* e *general* possuem como *stem* o termo *gener*.

Um algoritmo alternativo ao desenvolvido por Porter foi apresentado por (KROVETZ, 2000). A idéia central deste algoritmo é que uma análise semântica é fundamental para o processo de *stemming*, dessa forma, o algoritmo propõe que seja utilizado um dicionário para decidir qual o *stem* pode representar determinada palavra, minimizando assim os problemas de produção não restrita de *stems* encontrados no algoritmo de Porter. Um problema com o algoritmo de Krovetz é que palavras que deveriam ser agrupadas em um

mesmo *stem* acabam não sendo devido à característica conservadora do algoritmo. Ex: *predictions* apresenta como forma mínima *prediction*, enquanto que *prediction* apresenta *predict* como sua forma mínima.

Stemming, quando aplicado corretamente, é uma prática útil na etapa de pré- processamento de textos, devido ao fato de potencializar uma redução significativa na quantidade de termos de uma coleção de documentos, além disto, pode reforçar a análise estatística de freqüência dos termos, uma vez que diferentes variações de uma mesma palavra estarão sendo representadas por um único termo.

2.3 Classificação de Textos

Classificação de textos é uma técnica de extração de conhecimento que é aplicada sobre a representação estruturada de um documento. Esta técnica visa identificar a qual, ou quais classes pertence determinado documento. A classificação de textos é realizada tendo como base a detecção de uma combinação de características nos textos. Classificação de textos é comumente utilizada para classificar notícias, publicações, criação de filtros e etc.

Na seqüência são apresentadas duas técnicas de classificação de textos amplamente divulgadas na literatura. São utilizadas duas técnicas distintas de classificação para que os resultados encontrados sejam confrontados a fim de evitar que a análise dos resultados fosse fortemente influenciada pelo algoritmo de classificação.

2.3.1 KNN (*K-Nearest Neighbor*)

KNN é um método de classificação não supervisionado e baseado em analogia. A idéia central do método é que dado um conjunto de textos de treinamento, estes com suas classes definidas, um novo documento é classificado de acordo com a sua distancia para os k elementos mais próximos, ou seja, os k vizinhos. Assim, a classe do novo documento é a classe mais freqüente entre os k documentos vizinhos (CAMARGO, 2007).

A distância entre dois documentos pode ser calculada de acordo com diferentes métricas, as métricas mais comuns são:

- Euclidiana

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2}$$

- Manhattan

$$d(x, y) = |\, x_1 - y_1 \,| + |\, x_2 - y_2 \,| + \ldots + |\, x_n - y_n \,|$$

- Minkowski

$$d(x, y) = (|\, x_1 - y_1 \,|^q + |\, x_2 - y_2 \,|^q + \ldots + |\, x_n - y_n \,|^q)^{\frac{1}{q}}$$

O KNN tende a exigir um grande esforço computacional, uma vez que é necessário o cálculo de distância entre muitos pontos pertencentes ao espaço. Visando reduzir o esforço computacional, algumas variações do método foram implementadas, uma delas exposta em (SILVA, 2005), se baseia na idéia de calcular a distância apenas entre pontos dentro de uma esfera de raio R. Uma desvantagem desta abordagem é a possibilidade de o número de pontos dentro da esfera ser menor que k, quantidade de vizinhos necessários.

2.3.2 *Naive Bayes*

Este método é baseado na probabilidade condicional de um conjunto de palavras, que representam uma classe, estarem presentes em determinado documento. Assim, mensura-se a probabilidade do documento pertencer a cada uma das categorias, sendo o documento finalmente classificado de acordo com a categoria que apresentar maior probabilidade (CAMARGO, 2007).

O *Naive Bayes* parte do pressuposto que cada termo da representação estruturada é independente, não apresentando assim nenhuma relação entre eles. Assim a definição da probabilidade do documento pertencer a uma classe é calculada através do produto das probabilidades de cada termo pertencer à determinada classe. A fim de evitar que a probabilidade seja zerada, caso um dos termos que apresente probabilidade igual a zero, uma constante maior do que zero é adicionada a probabilidade encontrada para cada termo (JACKSON, 2002).

2.4 Parâmetros de avaliação dos resultados

A seguir são apresentados os parâmetros que serão utilizados para mensurar a qualidade da classificação dos dados provenientes da etapa de pré-processamento. Os parametros de avaliação apontam indicadores que servem para mensurar a qualidade do processo de classificação.

2.4.1 Precisão

Esta medida apresenta a proporção entre a quantidade de acertos na classificação para determinada categoria e o total de documentos classificados como pertencentes à categoria. Seu cálculo é realizado utilizando a seguinte fórmula:

$$Pr = \frac{TP}{TP+FP}$$

onde, TP é o número de documentos corretamente classificados e FP é o número de documentos classificados de forma incorreta na classe em questão.

2.4.2 *Recall*

Apresenta a proporção entre o total de documentos corretamente classificados para uma dada classe e o total de documentos pertencentes à classe em questão. Sua fórmula é dada por:

$$Re = \frac{TP}{TP+FN}$$

onde, TP é o número de documentos corretamente classificados e FN é a classificação incorreta da classe desejada.

2.4.3 Medida F

A Medida F é uma combinação de Precisão e Recall, esta combinação é realizada através do cálculo da média harmônica entre as medidas. Segundo (SILVA, 2005) a medida F é uma excelente opção para casos em que o classificador apresenta desempenho não uniforme para classes distintas, ou seja, apresenta um bom desempenho para uma classe e um desempenho ruim para outra classe. A fórmula para a medida F é:

$$F = \frac{2 \times Precisão \times Recall}{Precisão+Recall}$$

2.4.4 Exatidão

Indica a proporção de acertos para a classificação e sua fórmula é definida como:

$$Ac = \frac{DC}{TD}$$

onde, DC é o número de documentos corretamente classificados e TD é o total de documentos. As medidas de Precisão e Exatidão são distintas, a medida de Precisão leva em consideração apenas os documentos que pertencem a classe e que são classificados como pertencentes a classe. Já a medida de Exatidão, leva em conta também os documentos que não pertencem a classe e que não foram classificados como pertencentes a classe.

2.5 Trabalhos Relacionados

Mineração de Textos é um campo que se encontra no estado da arte. Assim, é grande o número de publicações que tratam do assunto, porém a fase especifica de Pré-Processamento de dados ainda é muito pouco explorada (GOLDSCHIMIT, 2005).

Em (ARANHA, 2007) é proposto um novo modelo para a representação estruturada que se baseia no paradigma de *bag-of-lexens*, uma variação do tradicional modelo *bag-of-words*, onde o documento deixa de ser representado por palavras e passa a ser representado por lexemas, com o objetivo de aprimorar a análise semântica do mesmo. Para isto utilizam-se técnicas de Processamento de Linguagem Natural e Lingüística Computacional.

(KAO, 2005) discute sobre a importância do processamento de linguagem natural no processo de mineração de textos e indica que o uso de Processamento de Linguagem Natural potencializa o processo de Mineração de Texto com o enriquecimento da análise semântica, principalmente no que diz respeito a busca por palavras.

Em (SOUCY, 2005) é apresentada uma nova medida de peso para cada termo na representação Modelo Espaço Vetorial, esta nova medida é proveniente do tradicional TFIDF e é denominada de *ConfWeight*.

3 Configuração do Experimento

Esta sessão apresenta a configuração e os procedimentos adotados durante a montagem e realização dos experimentos envolvendo a fase de Pré-Processamento de um processo de Mineração de Texto. Inicialmente serão expostos os critérios de seleção das bases de teste utilizadas, logo após serão discutidos os procedimentos utilizados para a montagem do modelo de representação dos dados, também serão abordadas as técnicas de Pré-Processamento, Análise dos Principais Componentes de um hiper-espaço de dados e Classificação.

3.1 Seleção das Bases de Teste

A fim de avaliar o comportamento das diferentes técnicas de Pré-Processamento foram selecionadas duas bases de testes para a aplicação das técnicas e análise dos resultados. As bases de teste selecionadas neste projeto são subconjuntos da base Reuters-21578. Esta base é composta por 22 arquivos no formato XML e totaliza aproximadamente 21.000 notícias, classificadas manualmente em distintas categorias. A escolha da base Reuters-21578 se deu pela sua ampla utilização em processos de Mineração de Texto, sendo que esta base pode ser encontrada em (LEWIS, 2004).

Neste projeto optou-se por utilizar subconjuntos da base Reuters-21578. Esta decisão teve por objetivo obter bases de testes mais restritas, onde o controle dos resultados fosse simplificado e o processamento computacional requerido fosse reduzido. A definição dos subconjuntos foi baseada em critérios que simplificassem a análise posterior à etapa de Pré-Processamento. Neste sentido, o primeiro critério de seleção para a montagem das sub-bases foi selecionar apenas documentos que apresentam uma única categoria, com a aplicação deste critério o número total de documentos da base foi reduzido, ficando em aproximadamente 10.000 documentos.

Após selecionar apenas documentos que apresentam uma única categoria, foi veri-

ficada a quantidade de documentos por categoria. Esta análise mostrou uma grande discrepância, algumas categorias possuem muitos documentos, enquanto outras possuem poucos, como exemplo pode-se citar as categorias *Earn* e *Crude*, com 3.945 e 414 documentos, respectivamente, enquanto que categorias como *Sunseed* e *Naphtha* apresentam, respectivamente, 1 e 2 documentos. Em particular, duas categorias dominam a base e apresentam uma quantidade bastante elevada de documentos em comparação com as demais categorias, são elas: *Earn* e *Acq*, com 3.945 documentos para a primeira e 2.358 para a segunda.

Após a seleção de documentos que apresentam apenas uma categoria e a análise da quantidade de documentos por categoria, foram selecionadas duas bases de teste para este projeto. A primeira composta por 1.105 documentos, divididos em três categorias e a segunda base composta por 902 documentos, divididos em cinco categorias. A seguir são apresentadas as bases escolhidas para serem utilizadas neste experimento.

3.1.1 Base 3C

A base de teste denominada Base 3C (esta nomenclatura será utilizada no decorrer deste trabalho) possui 1.105 documentos distribuídos em três categorias: *Crude*, *Trade* e *Money*. A divisão da base em categorias pode ser observada na figura 8.

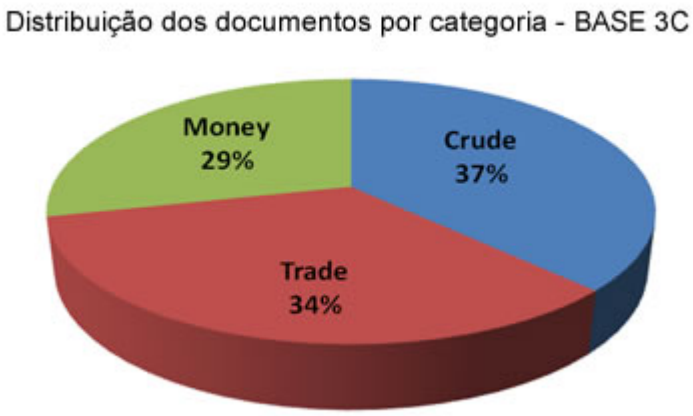

Figura 8: Distribuição dos documentos por categoria na Base 3C.

3.1.2 Base 5C

A segunda base utilizada é denominada Base 5C (nomenclatura esta que será utilizada no decorrer deste trabalho) sendo composta por 902 documentos, distribuídos em cinco

categorias distintas. A divisão dos documentos por categoria é apresentada na figura 9.

Figura 9: Distribuição dos documentos por categoria na Base 5C.

3.2 Realização do Experimento

Uma vez selecionadas as bases de teste, a seguir são apresentados os procedimentos que conduziram a realização do experimento. Inicialmente será apresentada a representação no Modelo Espaço Vetorial, com a utilização de TF, TFIDF e a representação Binária. A seguir serão apresentadas a configuração para as técnicas de *StopWords* e *Stemming*, além da escolha das configurações de pré-processamento.

3.2.1 Representação VSM

O passo inicial para a construção da representação do documento no Modelo Espaço Vetorial é conhecer as palavras que compõe o vocabulário da base de teste em questão. Para isto é necessário percorrer todos os documentos identificando novas palavras para a construção do vocabulário. Neste projeto foram definidos alguns critérios para determinar se a palavra seria ou não considerada válida para a representação.

Inicialmente o texto dos documentos foi separado por delimitadores, sendo que a lista dos delimitadores foi composta por todos os caracteres ASCII, exceto os alfanuméricos e o "_". Após a divisão do texto do documento em termos, os mesmos foram avaliados de acordo com a quantidade de caracteres que possuíam, de forma que os com menos de três caracteres foram descartados. Números também foram desconsiderados na representação,

constando como termo válido apenas os que são iniciados com caracteres alfabéticos e que possuem o restante dos caracteres como sendo alfanumérico, mais o caractere "_". Estas medidas tiveram por objetivo eliminar termos que não apresentam sentido semântico aos documentos.

Após conhecer todas as palavras que compõem os diferentes documentos da base, cada documento é representado por um vetor, onde cada termo do vocabulário corresponde a uma posição no vetor.

Com todos os documentos devidamente representados por seus vetores, é possível construir uma matriz onde as linhas da matriz são os documentos e as colunas correspondem aos termos do vocabulário. Por conseqüência de cada documento ser representado por um vetor composto de todos os termos do vocabulário, a matriz encontrada é uma matriz esparsa, uma vez que comumente os documentos não apresentam a mesma quantidade de termos que o vocabulário.

3.2.1.1 TF, TFIDF e Binário

No processo de construção dos vetores que representam cada documento, existem diferentes formas de identificar o grau de relevância que cada palavra exerce no mesmo. Neste projeto foram utilizadas quatro formas para representar a importância dos termos nos documentos.

A representação por TF (*Term Frequency*) é realizada atribuindo a cada termo do documento um valor inteiro que corresponde à quantidade de vezes que a palavra aparece no documento, para isto, basta percorrer os termos do documento e identificar a freqüência de ocorrência das palavras.

Para construir uma representação baseada em TF-IDF (*Term Frequency - Inverse Document Frequency*) é necessário além dos procedimentos para a representação por TF, identificar em quantos documentos da coleção cada termo ocorre. A medida de IDF possibilita inferir o quão discriminante o termo é para o documento, uma vez que um termo bastante comum, logo com alto TF, pode não contribuir para a discriminação do documento, já que é uma palavra bastante comum na coleção.

A construção da representação binária é bastante simples, basta identificar se a palavra ocorre ou não no documento, assim, palavras que ocorrem no documento possuem o valor 1 no vetor correspondente, as demais palavras recebem valor 0.

3.2.2 Aplicação das Técnicas de Pré-Processamento

A seguir será apresentada a forma de aplicação das técnicas de Pré-processamento. Primeiramente será demonstrada a aplicação da técnica de *StopWords*; em seguida é apresentada uma técnica de *Stemming* de Porter, a qual foi utilizada neste trabalho.

3.2.2.1 Remoção de *StopWords*

A remoção das *StopWords* neste projeto ocorreu no momento da formação do vocabulário das bases. Ao percorrer os termos de cada documento os mesmos foram comparados com uma lista das palavras *StopWords*. Esta lista foi previamente definida e pode ser encontrada em (ONIX, 2006). Desta forma, apenas os termos que não estivessem presentes na lista de *StopWords* foram mantidos na representação.

3.2.2.2 Aplicação de *Stemming*

Neste experimento foi utilizada a técnica de *stemming* de Porter. As técnicas de *stemming*, como já exposto no capitulo 2, tem por objetivo atenuar problemas de variações verbais, inerentes ao processo lingüístico. A aplicação de uma técnica de *Stemming* contribui para a redução do número de palavras do vocabulário e possivelmente altera a representação estatística baseada na freqüência de ocorrência dos termos, já que termos distintos podem ser reduzidos a um mesmo radical.

Neste projeto foi utilizada uma implementação, na linguagem Java, do algoritmo de Porter disponível em (TARTARUS, 2004).

3.2.3 Configurações escolhidas para os experimentos

Para a realização dos experimentos deste projeto foram escolhidas algumas configurações (variações) de técnicas de pré-processamento com o objetivo de analisar a contribuição das técnicas para a produção de um resultado. Inicialmente foram estabelecidas as técnicas a serem utilizadas para o balanceamento de peso dos termos, sendo definidas três abordagens: a primeira com a utilização de TF, a segunda utilizando TF-IDF e uma terceira com o uso de uma abordagem Binária.

Após a representação dos documentos no Modelo Espaço Vetorial e com seus pesos devidamente registrados, foi introduzida a técnica de *StopWords* em todos os vetores, com diferentes balanceamentos de pesos. A aplicação de técnicas de *stemming* também

constou como parte da análise, sendo que esta foi realizada com a utilização do algoritmo de Porter. A seguir é apresentada a tabela de configurações para o experimento, utilizada neste projeto.

TF, TFIDF e BINÁRIO	STOPWORDS
	STOPWORDS + PORTER STEMMING

Tabela 1: Configurações de pré-processamento utilizadas.

3.2.4 PCA

Análise de Componentes Principais é uma técnica que tem por objetivo projetar dados de altas dimensões para dimensões menores. A dificuldade dos humanos em analisar dados representados com mais de três dimensões, faz com que a aplicação da técnica de PCA seja aconselhável, uma vez que possibilita que os dados sejam visualizados numa dimensão menor, geralmente em três dimensões.

Neste projeto foi utilizado o software MatLab© para realizar o processo de PCA e visualização dos dados representados em três dimensões.

3.2.5 Classificação

O processo de classificação teve como objetivo verificar o desempenho das técnicas de Pré-Processamento. Foram utilizadas duas técnicas de classificação neste projeto: KNN e *Naive Bayes*.

Após a realização do Pré-Processamento, o vetor resultante foi submetido aos dois algoritmos de classificação. Para isto foi utilizado o software Weka© que oferece suporte necessário para a realização da tarefa em questão. Para utilização do software foi necessário gerar um arquivo no formato .arff, apropriado para o programa, sendo que este arquivo armazena informações sobre o vocabulário de palavras da base de trabalho, além da matriz que representa os documentos da coleção.

4 Resultados e Discussão

A seguir são apresentados os resultados para os experimentos realizados, assim como a discussão a cerca dos mesmos. Inicialmente é apresentada a distribuição das palavras por TF e DF para cada uma das bases de teste. Logo após é ilustrada uma representação em três dimensões da distribuição dos documentos que compõem as bases. A análise da classificação, quando aplicadas técnicas de pré-processamento também é apresentada, além da discussão a cerca dos experimentos.

4.1 Distribuição do TF e DF

A distribuição por TF de uma base de dados apresenta a idéia de como se encontra a distribuição de freqüência dos termos. Através da análise da distribuição por TF é possível verificar as palavras que aparecem com maior freqüência dentro da base, em geral, pouco representativas, não sendo relevante para a análise discriminatória dos documentos. As Figuras 10 e 11 apresentam os gráficos de TF para as bases de dados utilizadas neste experimento, são apresentadas também as curvas de TF, TF com aplicação da técnica de StopWords e TF com aplicação de *StopWords* e *Stemming*, todos representados em escala logarítmica. Percebe-se que a aplicação de *StopWords* e *Stemming* combinados reduz sensivelmente a quantidade de termos da representação e que o uso de *Stemming* eleva o valor de TF para muitos termos da representação.

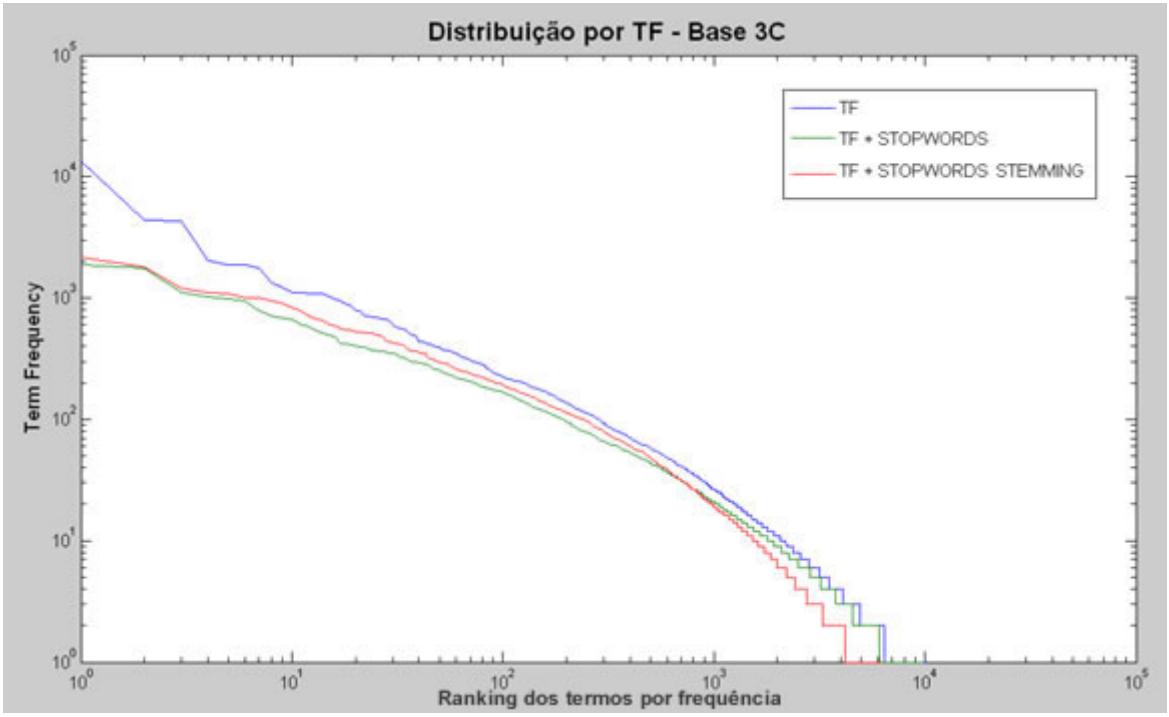

Figura 10: Distribuição das palavras por TF - Base 3C.

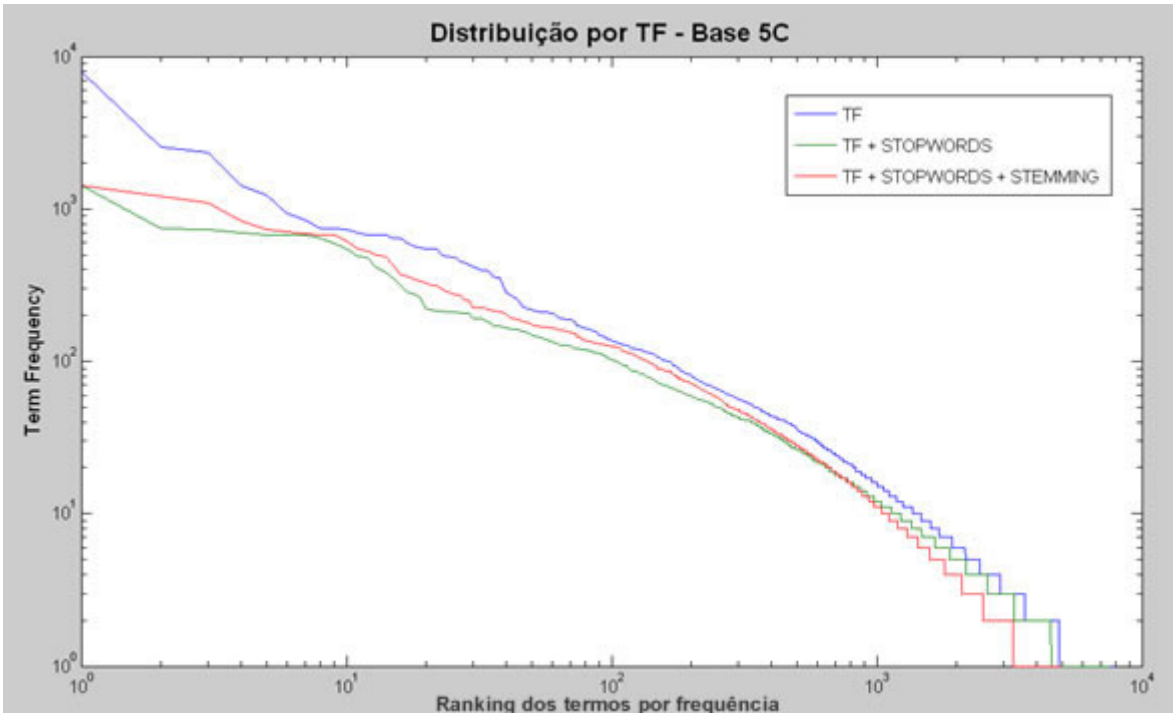

Figura 11: Distribuição das palavras por TF - Base 5C.

A análise de DF, *Document Frequency*, têm por objetivo identificar em quantos documentos uma mesma palavra está distribuida. Uma observação que pode ser constatado com a análise de DF é que palavras que aparecem em muitos documentos são palavras

comuns, pouco discriminatórias, e não contribuem de forma significante para a identificação do documento. A figura 12 ilustra a distribuição por DF para as bases de dados trabalhadas neste experimento.

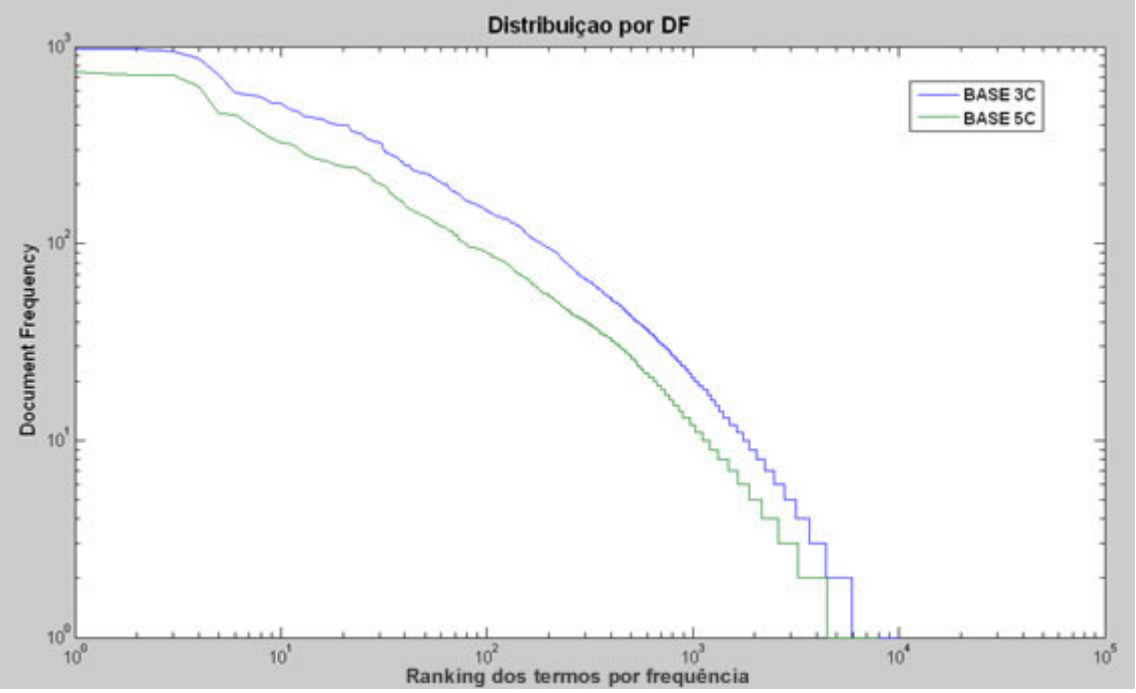

Figura 12: Distribuição das palavras por DF.

4.2 Visualização por PCA das Bases

A técnica de Análise de Componentes Principais possibilita que dados de um espaço com alta dimensionalidade sejam projetados numa dimensionalidade menor, possibilitando assim uma análise da distribuição dos mesmos numa dimensão que o ser humano é capaz de avaliar. A análise da distribuição dos dados é importante, pois demonstra, aproximadamente, a maneira como os dados estão distribuídos, servindo assim como suporte para a interpretação dos resultados encontrados em um processo de extração de informação, como exemplo um processo de classificação ou clusterização.

As Figuras 13 e 14 representam a distribuição, em três dimensões, dos dados representados na forma binária para as Bases 3C e 5C, respectivamente. Destaca-se na Figura 13 a dispersão dos dados da classe *Money_Fx*, enquanto que na Figura 14 apenas os dados da classe *Ship* apresentam um nível de compactação, estando os dados das demais classes bastante dispersos em relação aos da classe *Ship*.

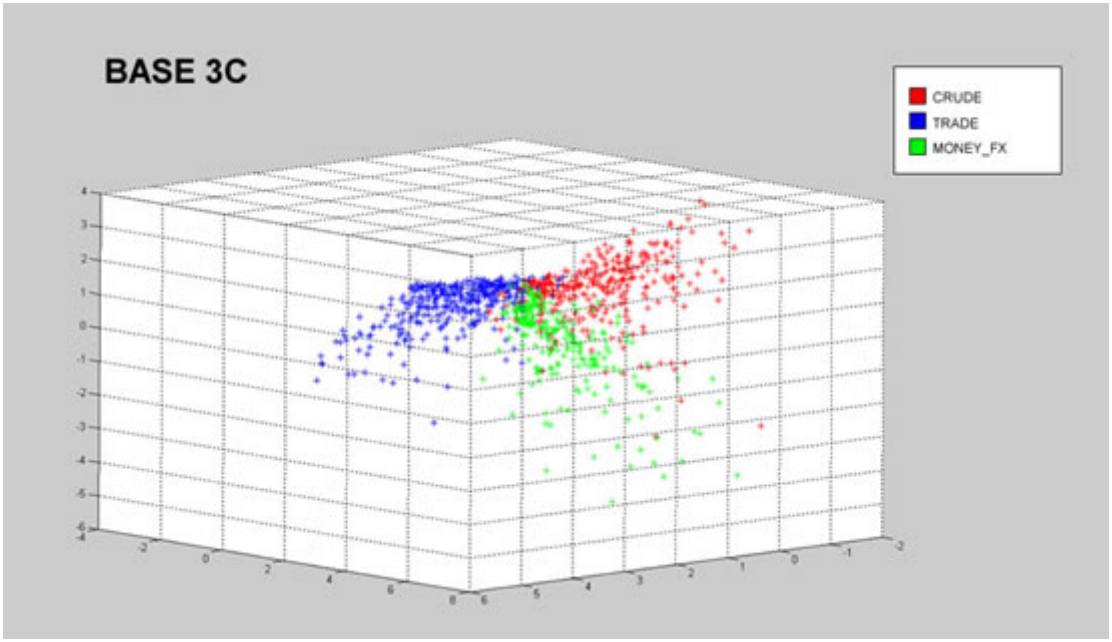

Figura 13: Visualização da distribuição aproximada dos dados - Base 3C.

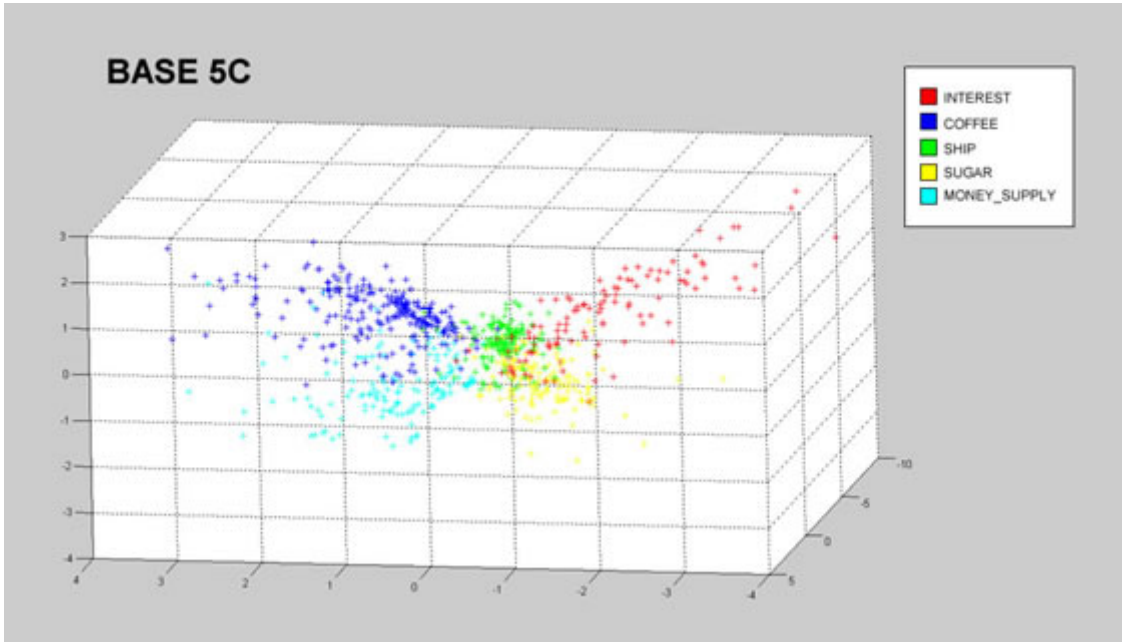

Figura 14: Visualização da distribuição aproximada dos dados - Base 5C.

4.3 Comparativo dos pré-processamentos

A seguir são apresentadas as tabelas com os resultados da aplicação da técnica de classificação nos dados provenientes da etapa de Pré-Processamento. Nas tabelas constam as diferentes variações de técnicas utilizadas nos experimentos. As métricas para avaliação

dos resultados apresentadas neste projeto são: Precisão, *Recall* (Abrangência), Medida F e Exatidão.

Para realizar o processo de classificação foram aplicados os algoritmos *Naive Bayes* e KNN, sendo que 66% das amostras constituíram a base de treinamento e o restante foi separado para teste. No caso do KNN foi utilizado um K=5.

A seguir é apresentado o resultado de Precisão, Recall, Medida F e Exatidão obtidos com a aplicação dos algoritmos *Naive Bayes* e KNN nas diversas variações no pré-processamento. As notações BE e B10 indicam respectivamente, logaritmo na base *e* e 10.

4.3.1 Resultados para a base 3C

As Tabelas 2 e 3 apresentam os resultados da análise de Precisão para a base 3C com aplicação dos algoritmos *Naive Bayes* e KNN, respectivamente.

BASE 3C	PRECISÃO		
	CRUDE	MONEY_FX	TRADE
TF	0,969	0,787	0,906
TFIDF_BE	0,969	0,787	0,906
TFIDF_B10	0,969	0,787	0,906
BINÁRIO	0,993	0,860	0,874
TF + STOPWORDS	0,970	0,835	0,914
TFIDF_BE + STOPWORDS	0,970	0,835	0,914
TFIDF_B10 + STOPWORDS	0,970	0,835	0,914
BINÁRIO + STOPWORDS	1	0,890	0,903
TF + STOPWORDS + STEMMING	0,977	0,793	0,894
TFIDF_BE + STOPWORDS + STEMMING	0,977	0,809	0,901
TFIDF_B10 + STOPWORDS + STEMMING	0,977	0,894	0,908
BINÁRIO + STOPWORDS + STEMMING	1	0,848	0,879

Tabela 2: Precisão resultante da aplicação do algoritmo de classificação *Naive Bayes* - Base 3C.

BASE 3C	PRECISÃO		
	CRUDE	MONEY_FX	TRADE
TF	0,851	0,401	0,717
TFIDF_BE	0,851	0,474	0,606
TFIDF_B10	0,851	0,474	0,606
BINÁRIO	0,987	0,513	0,877
TF + STOPWORDS	0,866	0,408	0,709
TFIDF_BE + STOPWORDS	0,866	0,481	0,602
TFIDF_B10 + STOPWORDS	0,866	0,481	0,602
BINÁRIO + STOPWORDS	1	0,540	0,941
TF + STOPWORDS + STEMMING	0,875	0,516	0,800
TFIDF_BE + STOPWORDS + STEMMING	0,875	0,583	0,658
TFIDF_B10 + STOPWORDS + STEMMING	0,875	0,583	0,658
BINÁRIO + STOPWORDS + STEMMING	0,980	0,600	0,933

Tabela 3: Precisão resultante da aplicação do algoritmo de classificação KNN - Base 3C.

A análise dos resultados de exatidão da classe 3C evidencia que a aplicação da técnica de representação binária apresenta os melhores resultados, especialmente quando combinada com apenas a técnica de *StopWords*.

As Tabelas 4 e 5 apresentam os resultados da análise de *Recall* para a base 3C com aplicação dos algoritmos *Naive Bayes* e KNN, respectivamente.

BASE 3C	RECALL		
	CRUDE	MONEY_FX	TRADE
TF	0,872	0,914	0,885
TFIDF_BE	0,872	0,914	0,885
TFIDF_B10	0,872	0,914	0,885
BINÁRIO	0,943	0,876	0,908
TF + STOPWORDS	0,915	0,914	0,900
TFIDF_BE + STOPWORDS	0,915	0,914	0,900
TFIDF_B10 + STOPWORDS	0,915	0,914	0,900
BINÁRIO + STOPWORDS	0,943	0,924	0,931
TF + STOPWORDS + STEMMING	0,887	0,876	0,908
TFIDF_BE + STOPWORDS + STEMMING	0,901	0,886	0,908
TFIDF_B10 + STOPWORDS + STEMMING	0,887	0,876	0,908
BINÁRIO + STOPWORDS + STEMMING	0,936	0,905	0,892

Tabela 4: Recall resultante da aplicação do algoritmo de classificação *Naive Bayes* - Base 3C.

BASE 3C	RECALL		
	CRUDE	MONEY_FX	TRADE
TF	0,404	0,829	0,508
TFIDF_BE	0,404	0,785	0,543
TFIDF_B10	0,404	0,785	0,543
BINÁRIO	0,539	0,943	0,715
TF + STOPWORDS	0,411	0,800	0,562
TFIDF_BE + STOPWORDS	0,411	0,800	0,533
TFIDF_B10 + STOPWORDS	0,411	0,800	0,533
BINÁRIO + STOPWORDS	0,610	0,971	0,731
TF + STOPWORDS + STEMMING	0,546	0,924	0,615
TFIDF_BE + STOPWORDS + STEMMING	0,546	0,754	0,752
TFIDF_B10 + STOPWORDS + STEMMING	0,546	0,754	0,752
BINÁRIO + STOPWORDS + STEMMING	0,702	0,971	0,754

Tabela 5: Recall resultante da aplicação do algoritmo de classificação KNN - Base 3C.

Para a medida de abrangência (*Recall*) o resultado é similiar ao apresentado na análise de exatidão, com a representação binária apresentando os melhores resultados.

As Tabelas 6 e 7 apresentam os resultados da análise da Medida F para a base 3C com aplicação dos algoritmos *Naive Bayes* e KNN, respectivamente.

BASE 3C	MEDIDA F		
	CRUDE	MONEY_FX	TRADE
TF	0,918	0,846	0,895
TFIDF_BE	0,918	0,846	0,895
TFIDF_B10	0,918	0,846	0,895
BINÁRIO	0,967	0,868	0,891
TF + STOPWORDS	0,942	0,873	0,907
TFIDF_BE + STOPWORDS	0,942	0,873	0,907
TFIDF_B10 + STOPWORDS	0,942	0,873	0,907
BINÁRIO + STOPWORDS	0,971	0,907	0,917
TF + STOPWORDS + STEMMING	0,929	0,833	0,901
TFIDF_BE + STOPWORDS + STEMMING	0,937	0,845	0,904
TFIDF_B10 + STOPWORDS + STEMMING	0,929	0,833	0,901
BINÁRIO + STOPWORDS + STEMMING	0,967	0,876	0,885

Tabela 6: Medida F resultante da aplicação do algoritmo de classificação *Naive Bayes* - Base 3C.

BASE 3C	MEDIDA F		
	CRUDE	MONEY_FX	TRADE
TF	0,548	0,540	0,595
TFIDF_BE	0,548	0,591	0,573
TFIDF_B10	0,548	0,591	0,573
BINÁRIO	0,697	0,664	0,788
TF + STOPWORDS	0,558	0,540	0,627
TFIDF_BE + STOPWORDS	0,558	0,601	0,566
TFIDF_B10 + STOPWORDS	0,558	0,601	0,566
BINÁRIO + STOPWORDS	0,758	0,694	0,823
TF + STOPWORDS + STEMMING	0,672	0,658	0,702
TFIDF_BE + STOPWORDS + STEMMING	0,500	0,669	0,678
TFIDF_B10 + STOPWORDS + STEMMING	0,500	0,669	0,678
BINÁRIO + STOPWORDS + STEMMING	0,818	0,742	0,834

Tabela 7: Medida F resultante da aplicação do algoritmo de classificação KNN - Base 3C.

A medida F confirma os resultados obtidos na análise de Precisão e *Recall*, apresentando assim a representação binária como a mais eficiente.

As Tabelas 8 e 9 apresentam os resultados da análise de Exatidão para a base 3C com aplicação dos algoritmos *Naive Bayes* e KNN, respectivamente.

BASE 3C	EXATIDÃO
TF	88,8298
TFIDF_BE	88,8298
TFIDF_B10	88,8298
BINÁRIO	91,2234
TF + STOPWORDS	90,9574
TFIDF_BE + STOPWORDS	90,9574
TFIDF_B10 + STOPWORDS	90,9574
BINÁRIO + STOPWORDS	93,3511
TF + STOPWORDS + STEMMING	89,0957
TFIDF_BE + STOPWORDS + STEMMING	89,8936
TFIDF_B10 + STOPWORDS + STEMMING	89,0957
BINÁRIO + STOPWORDS + STEMMING	91,2234

Tabela 8: Exatidão resultante da aplicação do algoritmo de classificação *Naive Bayes* - Base 3C.

BASE 3C	EXATIDÃO
TF	55,8511
TFIDF_BE	57,4468
TFIDF_B10	57,4468
BINÁRIO	71,2766
TF + STOPWORDS	57,1809
TFIDF_BE + STOPWORDS	57,9787
TFIDF_B10 + STOPWORDS	57,9787
BINÁRIO + STOPWORDS	75,266
TF + STOPWORDS + STEMMING	67,5532
TFIDF_BE + STOPWORDS + STEMMING	62,766
TFIDF_B10 + STOPWORDS + STEMMING	62,766
BINÁRIO + STOPWORDS + STEMMING	79,5213

Tabela 9: Exatidão resultante da aplicação do algoritmo de classificação KNN - Base 3C.

As Figuras 15 e 16 apresentam os resultados da análise de Exatidão para a base 3C com aplicação dos algoritmos *Naive Bayes* e KNN, respectivamente.

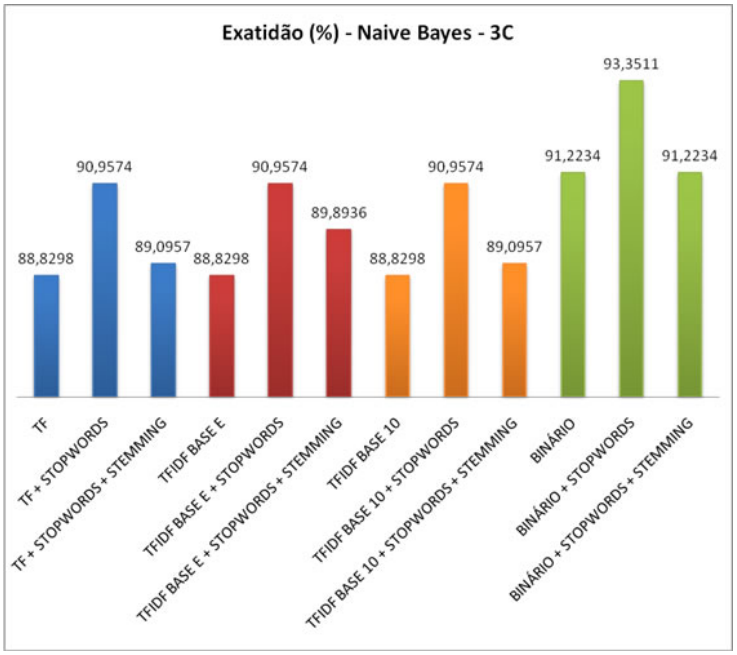

Figura 15: Exatidão percentual resultante da aplicação do algoritmo de classificação *Naive Bayes* - Base 3C.

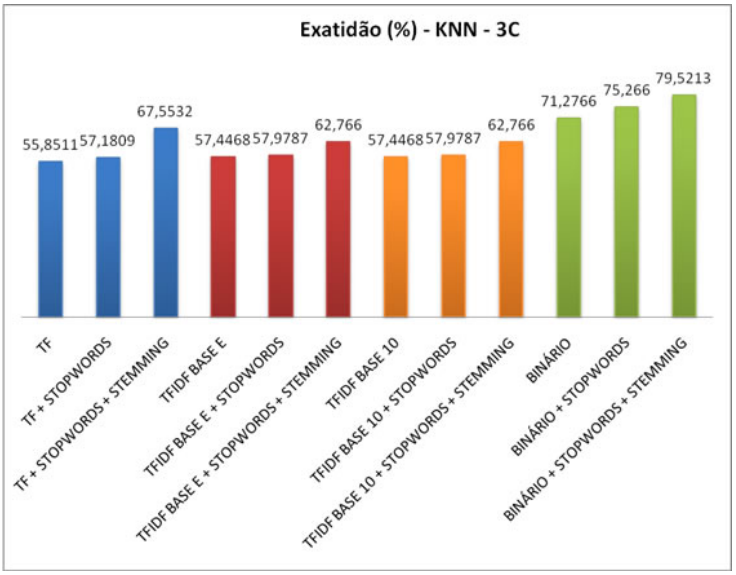

Figura 16: Exatidão percentual resultante da aplicação do algoritmo de classificação KNN - Base 3C.

Analisando a exatidão é possivel verificar que para o algoritmo *Naive Bayes*, o melhor resultado é obtido com a aplicação da representação binária juntamente com a técnica de *StopWords*. Já para o algoritmo KNN, a técnica que apresenta o melhor desempenho ainda é a representação binária, porém com a aplicação de *StopWords* e *Stemming*.

4.3.2 Resultados para a base 5C

As Tabelas 10 e 11 apresentam os resultados da análise de Precisão para a base 5C com aplicação dos algoritmos *Naive Bayes* e KNN, respectivamente.

BASE 5C	PRECISÃO				
	COFFEE	INTEREST	MONEY-SUPPLY	SHIP	SUGAR
TF	0,864	0,889	0,529	0,934	0,889
TFIDF_BE	0,864	0,889	0,529	0,934	0,889
TFIDF_B10	0,864	0,889	0,529	0,934	0,889
BINÁRIO	0,914	0,890	0,667	0,889	0,943
TF + STOPWORDS	0,886	0,892	0,688	0,966	0,895
TFIDF_BE + STOPWORDS	0,886	0,894	0,710	0,966	0,895
TFIDF_B10 + STOPWORDS	0,886	0,892	0,688	0,966	0,895
BINÁRIO + STOPWORDS	0,947	0,922	0,918	0,905	0,945
TF + STOPWORDS + STEMMING	0,816	0,903	0,613	0,947	0,907
TFIDF_BE + STOPWORDS + STEMMING	0,816	0,903	0,613	0,947	0,907
TFIDF_B10 + STOPWORDS + STEMMING	0,816	0,903	0,613	0,947	0,907
BINÁRIO + STOPWORDS + STEMMING	0,900	0,922	0,915	0,903	0,946

Tabela 10: Precisão resultante da aplicação do algoritmo de classificação *Naive Bayes* - Base 5C.

BASE 5C	PRECISÃO				
	COFFEE	INTEREST	MONEY-SUPPLY	SHIP	SUGAR
TF	0,471	0,732	0,301	1	0,786
TFIDF_BE	0,471	0,732	0,301	1	0,786
TFIDF_B10	0,471	0,732	0,301	1	0,786
BINÁRIO	0,933	0,972	0,283	0,923	1
TF + STOPWORDS	0,633	0,707	0,301	1	0,929
TFIDF_BE + STOPWORDS	0,633	0,707	0,301	1	0,929
TFIDF_B10 + STOPWORDS	0,633	0,707	0,301	1	0,929
BINÁRIO + STOPWORDS	1	0,985	0,266	0,889	1
TF + STOPWORDS + STEMMING	0,607	0,598	0,313	1	0,879
TFIDF_BE + STOPWORDS + STEMMING	0,607	0,598	0,313	1	0,879
TFIDF_B10 + STOPWORDS + STEMMING	0,607	0,598	0,313	1	0,879
BINÁRIO + STOPWORDS + STEMMING	0,933	0,975	0,301	0,941	1

Tabela 11: Precisão resultante da aplicação do algoritmo de classificação KNN - Base 5C.

Para a classe 5C os resultados são similares aos apresentados ara a classe 3C, tendo assim, a representação binária apresentando os resultados mais eficientes.

As Tabelas 12 e 13 apresentam os resultados da análise de *Recall* para a base 5C com aplicação dos algoritmos *Naive Bayes* e KNN, respectivamente.

BASE 3C	RECALL				
	COFFEE	INTEREST	MONEY-SUPPLY	SHIP	SUGAR
TF	0,884	0,577	0,882	0,950	0,857
TFIDF_BE	0,884	0,577	0,882	0,950	0,857
TFIDF_B10	0,884	0,577	0,882	0,950	0,857
BINÁRIO	0,774	0,835	0,863	0,933	0,893
TF + STOPWORDS	0,907	0,763	0,863	0,950	0,911
TFIDF_BE + STOPWORDS	0,907	0,784	0,863	0,950	0,911
TFIDF_B10 + STOPWORDS	0,907	0,763	0,863	0,950	0,911
BINÁRIO + STOPWORDS	0,837	0,969	0,882	0,950	0,929
TF + STOPWORDS + STEMMING	0,930	0,670	0,902	0,900	0,875
TFIDF_BE + STOPWORDS + STEMMING	0,930	0,670	0,902	0,900	0,875
TFIDF_B10 + STOPWORDS + STEMMING	0,930	0,670	0,902	0,900	0,875
BINÁRIO + STOPWORDS + STEMMING	0,837	0,969	0,843	0,933	0,946

Tabela 12: Recall resultante da aplicação do algoritmo de classificação *Naive Bayes* - Base 5C.

BASE 5C	RECALL				
	COFFEE	INTEREST	MONEY-SUPPLY	SHIP	SUGAR
TF	0,372	0,536	0,980	0,133	0,393
TFIDF_BE	0,372	0,536	0,980	0,133	0,393
TFIDF_B10	0,372	0,536	0,980	0,133	0,393
BINÁRIO	0,326	0,722	0,961	0,200	0,607
TF + STOPWORDS	0,442	0,546	0,980	0,133	0,464
TFIDF_BE + STOPWORDS	0,442	0,546	0,980	0,133	0,464
TFIDF_B10 + STOPWORDS	0,442	0,546	0,980	0,133	0,464
BINÁRIO + STOPWORDS	0,256	0,680	0,980	0,133	0,571
TF + STOPWORDS + STEMMING	0,395	0,567	0,882	0,167	0,518
TFIDF_BE + STOPWORDS + STEMMING	0,395	0,567	0,882	0,167	0,518
TFIDF_B10 + STOPWORDS + STEMMING	0,395	0,567	0,882	0,167	0,518
BINÁRIO + STOPWORDS + STEMMING	0,326	0,794	0,961	0,267	0,589

Tabela 13: Recall resultante da aplicação do algoritmo de classificação KNN - Base 5C.

A análise de *Recall* também apresentou resultados similares aos da classe 3C, com a técnica de representação binária apresentando os melhores resultados. Aqui a aplicação da representação binária em conjunto com a técnica de *StopWords* apresentou os melhores resultados quando da aplicação do algoritmo *Naive Bayes*, já para o algoritmo KNN os resultados mais eficientes foram encontrados com a aplicação da técnica binário + *StopWords* + *Stemming*.

As Tabelas 14 e 15 apresentam os resultados da análise da Medida F para a base 5C com aplicação dos algoritmos *Naive Bayes* e KNN, respectivamente.

BASE 5C	MEDIDA F				
	COFFEE	INTEREST	MONEY-SUPPLY	SHIP	SUGAR
TF	0,874	0,700	0,662	0,942	0,873
TFIDF_BE	0,874	0,700	0,662	0,942	0,873
TFIDF_B10	0,874	0,700	0,662	0,942	0,873
BINÁRIO	0,821	0,862	0,759	0,911	0,917
TF + STOPWORDS	0,897	0,822	0,765	0,958	0,903
TFIDF_BE + STOPWORDS	0,897	0,835	0,779	0,958	0,903
TFIDF_B10 + STOPWORDS	0,897	0,822	0,765	0,958	0,903
BINÁRIO + STOPWORDS	0,889	0,945	0,900	0,927	0,937
TF + STOPWORDS + STEMMING	0,870	0,769	0,730	0,923	0,891
TFIDF_BE + STOPWORDS + STEMMING	0,870	0,769	0,730	0,923	0,891
TFIDF_B10 + STOPWORDS + STEMMING	0,870	0,769	0,730	0,923	0,891
BINÁRIO + STOPWORDS + STEMMING	0,867	0,945	0,878	0,918	0,946

Tabela 14: Medida F resultante da aplicação do algoritmo de classificação *Naive Bayes* - Base 5C.

BASE 5C	MEDIDA F				
	COFFEE	INTEREST	MONEY-SUPPLY	SHIP	SUGAR
TF	0,416	0,619	0,461	0,235	0,524
TFIDF_BE	0,416	0,619	0,461	0,235	0,524
TFIDF_B10	0,416	0,619	0,461	0,235	0,524
BINÁRIO	0,483	0,828	0,438	0,329	0,756
TF + STOPWORDS	0,521	0,616	0,461	0,235	0,619
TFIDF_BE + STOPWORDS	0,521	0,616	0,461	0,235	0,619
TFIDF_B10 + STOPWORDS	0,521	0,616	0,461	0,235	0,619
BINÁRIO + STOPWORDS	0,407	0,805	0,418	0,232	0,727
TF + STOPWORDS + STEMMING	0,479	0,528	0,462	0,286	0,652
TFIDF_BE + STOPWORDS + STEMMING	0,479	0,528	0,462	0,286	0,652
TFIDF_B10 + STOPWORDS + STEMMING	0,479	0,528	0,462	0,286	0,652
BINÁRIO + STOPWORDS + STEMMING	0,483	0,875	0,458	0,416	0,742

Tabela 15: Medida F resultante da aplicação do algoritmo de classificação KNN - Base 5C.

As Tabelas 16 e 17 apresentam os resultados da análise de Exatidão para a base 5C com aplicação dos algoritmos *Naive Bayes* e KNN, respectivamente.

BASE 5C	EXATIDÃO
TF	79,4788
TFIDF_BE	79,4788
TFIDF_B10	79,4788
BINÁRIO	85,6678
TF + STOPWORDS	86,3192
TFIDF_BE + STOPWORDS	86,9707
TFIDF_B10 + STOPWORDS	86,3192
BINÁRIO + STOPWORDS	92,50811
TF + STOPWORDS + STEMMING	82,7362
TFIDF_BE + STOPWORDS + STEMMING	82,7362
TFIDF_B10 + STOPWORDS + STEMMING	82,7362
BINÁRIO + STOPWORDS + STEMMING	91,8567

Tabela 16: Exatidão resultante da aplicação do algoritmo de classificação *Naive Bayes* - Base 5C.

BASE 5C	EXATIDÃO
TF	48,2085
TFIDF_BE	48,2085
TFIDF_B10	48,2085
BINÁRIO	58,3062
TF + STOPWORDS	50,8143
TFIDF_BE + STOPWORDS	50,8143
TFIDF_B10 + STOPWORDS	50,8143
BINÁRIO + STOPWORDS	54,3974
TF + STOPWORDS + STEMMING	50,8143
TFIDF_BE + STOPWORDS + STEMMING	50,8143
TFIDF_B10 + STOPWORDS + STEMMING	50,8143
BINÁRIO + STOPWORDS + STEMMING	61,5635

Tabela 17: Exatidão resultante da aplicação do algoritmo de classificação KNN - Base 5C.

As Figuras 17 e 18 apresentam os resultados da análise de Exatidão para a base 5C com aplicação dos algoritmos *Naive Bayes* e KNN, respectivamente.

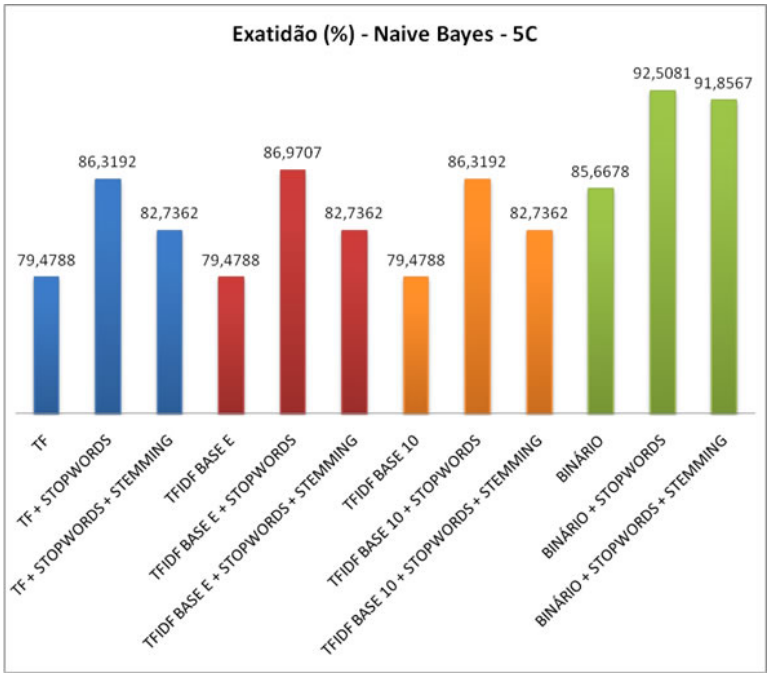

Figura 17: Exatidão percentual resultante da aplicação do algoritmo de classificação *Naive Bayes* - Base 5C.

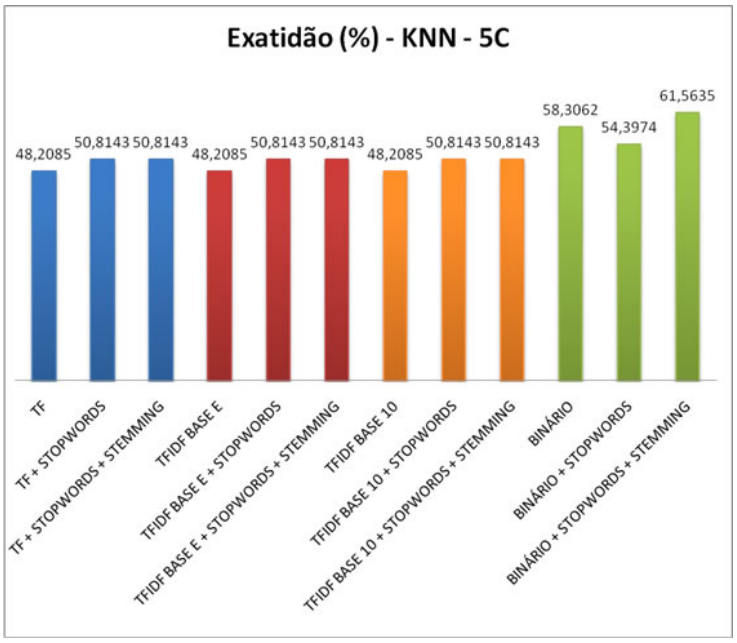

Figura 18: Exatidão percentual resultante da aplicação do algoritmo de classificação KNN - Base 5C.

Os resultados de exatidão para a classe 5C foram semelhantes aos apresentados para a classe 3C.

4.4 Discussão

Analisando os resultados da distribuição dos dados por TF, é possível verificar que poucas palavras possuem uma freqüência bastante elevada e muitas palavras são pouco freqüentes, estas palavras comumente não apresentam um conteúdo discriminatório para o documento.

Percebe-se na Figura 10 que após a aplicação da técnica de *StopWords*, os termos mais freqüentes foram retirados da representação, sendo os mesmos identificados como *StopWords*. O termo de maior freqüência na base, antes da aplicação de *StopWords* possui um valor de TF em torno de 14.000, após a aplicação de *StopWords* o termo mais freqüente teve um valor de TF em torno de 1.900. Este resultado evidencia que as palavras com freqüência mais elevadas, eram de fato *StopWords* e confirma o exposto por (BAEZA, 1999), onde é indicado que as *StopWords* possuem alta freqüência em documentos de texto.

Outra análise importante, em se tratando da distribuição de TF, é que após a aplicação da técnica de *Stemming*, em conjunto com a técnica de *StopWords*, os valores da distribuição por TF são alterados. Isto é justificado principalmente por dois motivos: o primeiro é o fato da aplicação de *Stemming* condensar diversas palavras em um único termo, alterando assim o valor de TF. Como exemplo pode-se citar as palavras *banks*, *banking* e *bank*, que após a aplicação de *Stemming* passaram a ser representadas pelo termo *bank*; o segundo motivo é o erro provocado pela aplicação da técnica de *Stemming*, palavras com significados distintos são reduzidas a um mesmo termo, como exemplo: *general*, *generate*, *generated* são reduzidos ao termo *gener*, provocando um erro no valor de TF atribuído.

O DF serve como indicação para a quantidade de documentos em que aparece uma mesma palavra, com esta indicação e de acordo com a idéia do IDF, que indica que palavras que estão presentes em muitos documentos tendem a ser pouco discriminatórias, é possível definir parâmetros para corte de termos que potencialmente são pouco representativos, estes cortes são indicados na literatura como Cortes de Luhn. Em (MATSUBARA, 2003) é exposto certa arbitrariedade na escolha dos limites de corte superior e inferior da representação e ainda discute que para Mineração de Texto as palavras que possuem maior poder de discriminação estão num nível intermediário do valor de DF.

A análise de Componentes Principais das bases de dados trabalhadas neste experimento mostra que a distribuição dos documentos, identificados por classes, não apresenta fronteiras bem definidas para a distribuição das classes, como pode ser observado nas

Figuras 13 e 14, que representam a base 3C e 5C, respectivamente.

O comparativo dos resultados da classificação para as diferentes técnicas de pré-processamento permite indicar que a técnica binária foi a mais eficiente. As Figuras 15, 17, 16 e 18 apresentam os resultados de Exatidão para todas as configurações utilizadas neste experimento. Um ponto interessante dos resultados de exatidão é que a variação da base do logaritmo, para o cálculo do IDF, provocou poucas mudanças no resultado final da classificação. Para a base 3C, Figura 15, a representação utilizando o logaritmo na base "e"apresentou um melhor resultado quando aplicadas as técnicas de *Stemming* e *StopWords*.

Percebe-se também que quando combinada as técnicas de *StopWords* e *Stemming* o resultado foi um pouco prejudicado quando comparado a aplicação apenas da técnica de StopWords, a explicação para este fato é proveniente dos erros provocados pelo processo de *Stemming*, onde palavras com significado distintos são representadas pelo menos termo.

5 Conclusão

O objetivo principal do presente trabalho foi o estudo, análise e comparação de diferentes técnicas de pré-processamento num processo de Mineração de Textos. O experimento realizado e os resultados obtidos indicam que foi atingido o principal objetivo proposto. Algumas considerações sobre a variações de pré-processamento foram omitidas e figuram dentre os possíveis trabalhos futuros, dentro elas pode-se citar a análise da técnica de normalização dos dados.

No que diz respeito ao processo de Mineração de Textos foi possível entender que com o crescimento exponencial do volume de informação disponível, técnicas de descoberta de conhecimento automáticas representam uma grande vantagem competitiva no cenário globalizado. Em meio a este grande volume de informações, encontrar informação de qualidade é uma tarefa árdua e que quando realizada manualmente acaba por requerer uma grande quantidade de tempo. Assim, a mineração de textos tem a função de auxiliar no processo de descoberta de conhecimento, automatizando o processo e, por conseguinte, o tornando menos obstrutivo.

Diversas áreas de conhecimento, como: medicina, biologia, química, física, administração e economia, se utilizam dos benefícios proporcionados pela mineração de textos, o que a torna uma área chave na conjuntura onde se encontra um grande volume de dados.

É notória a importância que o pré-processamento possui dentro de um processo de Mineração de Textos, um pré-processamento bem ajustado possibilita uma melhora sensível na exatidão dos resultados encontrados, além de reduzir a dimensionalidade do problema e conseqüentemente reduzir o esforço computacional.

Outro ponto de destaque em Mineração de Texto é que a base de dados e o algoritmo de extração de conhecimento utilizado possui uma grande importância na eficiência da aplicação de determinada técnica, sendo assim, não foi possível determinar uma configuração de pré-processamento que possua sempre um desempenho superior.

Para trabalhos futuros, um ponto importante a ser observado é a busca por um incre-

mento semântico na análise de Mineração de Texto, este incremento pode ser obtido com o uso de um dicionário de sinônimos para a construção de uma ontologia, possibilitando assim que a semântica seja otimizada. A aplicação de outros algoritmos de classificação também representa um trabalho futuro, assim como autilização de uma outra base de documentos.

Referências

ARANHA, C. *Uma Abordagem de Pré-Processamento Automático para Mineração de Textos em Português: Sob o Enfoque da Inteligência Computacional*. Tese (Doutorado), 2007.

BAEZA, Y. *Modern Information Retrieval*. [S.l.]: Addison Wesley, 1999.

BEIL, F. *Frequent Term-Based Text Clustering*. [S.l.], 2002.

BEPPLER, D. *Aplicação de Text Mining para a Extração de Conhecimento Jurisprudencial*. [S.l.], 2005.

BERRY, M. W. *Survey of Text Mining: Clustering, Classification, and Retrieval*. [S.l.: s.n.], 2004.

CAMARGO, Y. *Abordagem Linguística na Classificação Automática de Textos em Português*. Tese (Doutorado), 2007.

ERHARDT, A. *Status of Text-mining Techniques Applied to Biomedical Text*. [S.l.], April 2006.

GELBUKH, A. *Zipf and Heaps Laws' Coefficients Depend on Language*. [S.l.], Jan 2001.

GOLDSCHIMIT, R. *Data Mining: um Guia Prático*. [S.l.]: Editora Campus, 2005.

GOMES, G. *Integração de Repositórios de Sistemas de Bibliotecas Digitais e de Sistemas de Aprendizagem*. Tese (Doutorado), 2006.

HAYES, J. *Text mining for software engineering: how analyst feedback impacts final results*. [S.l.]: ACM SIGSOFT Software Engineering Notes, 2005.

HOOPER, R. *The Lancaster Stemming Algorithm*. [S.l.], 2008. Disponível em: <http://www.comp.lancs.ac.uk/computing/research/ stemming/index.htm>.

JACKSON, P. *Natural Language processing for Online Applications - Text retrieval, Extraction and Categorization*. [S.l.]: Philadelphia: John Benjamins B.V., 2002.

KANTROWITZ, R. *Stemming and its effects on TFIDF ranking*. [S.l.], 2000.

KAO, A. *Text mining and natural language processing: introduction for the special issue*. [S.l.], 2005.

KROVETZ, R. *Viewing morphology as an inference process*. [S.l.: s.n.], 2000.

LEWIS, D. *Reuters-21578 Test Collecion*. [S.l.], 2004. Disponível em: <http://www.daviddlewis.com/resources/testcollections/reuters21578>.

LO, R. *Automatically Building a Stopword List for an Information Retrieval System*. [S.l.], 2005.

MARTINS, C. *Reducing the Dimensionality of Bag-of-Words Text Representation Used by Learning Algorithms*. [S.l.], 2003.

MATSUBARA, E. *PreTexT - uma ferramenta para pré-processamento de textos utilizando a abordagem bag-of-words*. [S.l.], 2003.

MOBASHER, B. *Intelligent Information Retrieval*. [S.l.], 2004. Disponível em: <http://maya.cs.depaul.edu/ classes/ds575/porter.html>.

NOGUEIRA, O. *Recuperação de Informação*. [S.l.], 2000.

ONIX, A. *Onix Text Retrieval Toolkit*. [S.l.], 2006.

QUI, L. *An Extensive Empirical Study of Feature Selection for Text Categorization*. [S.l.], 2008.

RAMOS, J. *Using TF-IDF to Determine Word Relevance in Document Queries*. [S.l.], 2003.

SALTON, G. *A Vector Space Model for Automatic Indexing*. [S.l.: s.n.], 1997.

SILVA, L. *Uma Aplicação de Árvores de Decisão, Redes Neurais e KNN para Identificação de Modelos ARMA Não-Sazonais e Sazonais*. Tese (Doutorado), 2005.

SOUCY, P. *Beyond TFIDF Weighting for Text Categorization in the Vector Space Model*. [S.l.], August 2005.

SPINAKIS, A. *A Text Mining Tool Supporting Business Intelligence*. [S.l.], January 2004.

STRZALKOWSKI, T. *Natural language information retrieval*. [S.l.: s.n.], 1999.

SULLIVAN, D. *The Need for Text Mining in Business Intelligence*. [S.l.], December 2000. Disponível em: <http://www.dmreview.com/issues/20001201/2791-1.html>.

TAN, A. *Text Mining: The state of the art and the challenges*. [S.l.], April 1999.

TARTARUS, A. *The Porter Stemming Algorithm*. [S.l.], 2004.

WEB-MINING-FR, A. *Portail francophone du web mining et des network sciences*. [S.l.], 2007.

WORDNET, A. *A lexical database for the English language*. [S.l.], 2004.