UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ

CAMPUS LUIZ MENEGHEL

MARIELI MARCHIONE

CLASSIFICAÇÃO AUTOMÁTICA DE NOTÍCIAS TEXTUAIS UTILIZANDO MINERAÇÃO DE TEXTOS.

Bandeirantes

2010

MARIELI MARCHIONE

CLASSIFICAÇÃO AUTOMÁTICA DE NOTÍCIAS TEXTUAIS UTILIZANDO MINERAÇÃO DE TEXTOS.

Trabalho de Conclusão de Curso submetido à Universidade Estadual do Norte do Paraná – *campus* Luiz Meneghel - como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Me. André Luís A. Menolli

Bandeirantes

2010

MARIELI MARCHIONE

CLASSIFICAÇÃO AUTOMÁTICA DE NOTÍCIAS TEXTUAIS UTILIZANDO MINERAÇÃO DE TEXTOS.

Trabalho de Conclusão de Curso submetido à Universidade Estadual do Norte do Paraná – *campus* Luiz Meneghel - como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

COMISSÃO EXAMINADORA

_____

Prof. Me. André Luís A. Menolli
UENP – *campus* Luiz Meneghel

_____

Prof. Me. Glauco Carlos Silva
UENP – *campus* Luiz Meneghel

_____

Prof. Me Ricardo Gonçalves Coelho
UENP – *campus* Luiz Meneghel

Bandeirantes, 26 de novembro de 2010

A meu pai, Mário, por toda a confiança em mim depositada, por sempre saber me apoiar e passar valores de honestidade, perseverança e fé.

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus, sem o qual não chegaria a lugar nenhum. Aos meus pais, Vanilce e Mário, por todo apoio nesses quatro anos de caminhada, pela paciência e compreensão nas horas em que precisei me afastar, em função dos estudos.

Deixo também aqui, meu agradecimento e toda minha gratidão ao meu orientador, Professor André, que com muita sabedoria e paciência colaborou muito para a realização deste trabalho.

Quero agradecer também, ao meu irmão Mário Sérgio, pelo simples fato de estar perto de mim e me apoiar em minhas escolhas.

Agradeço a todos os companheiros de trabalho da Escola Municipal Ignez Panichi Hamzé, em especial à diretora Renata, por todas as vezes que me liberou para que eu pudesse ir à orientação, sempre muito compreensiva.

Ao companheirismo de todos da XII turma de Sistemas de Informação, em especial meus grandes amigos durante o curso: Ronaldo, Fabio, Bruna e Paulo. Obrigada pela amizade e carinho.

Agradeço também, a todos os amigos da “van do seu Júlio”, os quais faziam o caminho de Cambará para Bandeirantes ser bem mais curto.

As minhas amigas: Majorí, a irmã que escolhi, por todo o incentivo, Hellen pela ajuda com as traduções, e Barbara, minha advogada preferida, por sempre me acalmar.

Enfim, deixo aqui meu agradecimento a todos, que direta ou indiretamente colaboraram para que este trabalho pudesse ser realizado.

"Estamos afogados em informação,

mas morrendo de fome por conhecimento."

John Naisbett

RESUMO

Com a evolução dos sistemas de informação, ocorreu uma grande expansão na geração de dados, sendo estes, em sua grande maioria, dados textuais. Foi tratada da Mineração de Textos, a qual consiste em encontrar informação útil em documentos não-estruturados. O *Knowledge Discovery in Textbases*, como também é conhecido, utiliza técnicas para tratamento, como remoção das palavras menos importantes do texto, dos prefixos e sufixos, entre outras técnicas de limpeza e recuperação de informações, para transformar a informação textual em dados estruturados e depois aplicar técnicas já consagradas de Mineração de Dados com o intuito de encontrar e classificar as informações que estavam escondidas em meio às bases textuais, para que estas estejam rapidamente disponíveis, auxiliando assim o processo de tomada de decisão. Neste contexto, foram realizados testes com bases de Corpos de Texto – notícias, e efetuar comparações entre o desempenho de alguns algoritmos de classificação automática, entre eles K-Nearest Neighbor, Support Vector Machine, Árvores de Decisão e Naives Bayes, que foram executados dentro da ferramenta Rapidminer 5. Como conclusão deste trabalho foi observado e apontado o algoritmo mais indicado para este tipo de dado textual, que foi o algoritmo K-NN.

Palavras-chave: Mineração de Textos; Mineração de Dados; Classificação.

ABSTRACT

With the evolution of information systems, there was a boom in data generation, the latter being mostly, textual data. She was treated in Text Mining, which is to find useful information in unstructured documents. The Knowledge Discovery in Textbases, as is also known, uses techniques for treatment such as removal of less important words of text, prefixes and suffixes, among other cleaning techniques and information retrieval, to turn textual information into structured data and then apply techniques already established data mining in order to find and classify information that was hidden in the midst of text bases, so they are readily available, thereby aiding the process of decision making. In this context, tests were carried out with bases Bodies Text - news, and make comparisons between the performance of some algorithms of automatic classification, including K-Nearest Neighbor, Support Vector Machine, Decision Trees and Naive Bayes, which were implemented within Tool Rapidminer 5. In conclusion of this work was observed and pointed out the algorithm more suitable for this type of textual data, which was the K-NN algorithm.

Key-Words: Text Mining; Data Mining; Classification

Lista de Figuras

Sumário

1 INTRODUÇÃO

A cada ano que se passa, mais dados estão sendo gerados. Qualquer loja, empresa, possui uma aplicação que realiza várias transações e estas geram dados. Todos esses dados incluem informações valiosas, por exemplo, tendências e padrões, aos quais poderiam ser usados para melhorar decisões empresariais (Gobel & Gruenwald, 1999). Porém com esse número tão elevado de dados e informações armazenadas em inúmeros bancos de dados torna-se difícil enxergar e extrair manualmente as informações ou o conhecimento embutidos nesses repositórios, e nesse contexto surge então os estudos sobre a Mineração de Dados.

As pesquisas nessa área começaram a surgir no fim dos anos 80, primeiramente tratando apenas de dados estruturados, mas com o tempo houve uma evolução nas pesquisas, e atualmente fala-se muito também em Mineração de Dados não-estruturados. Mineração de Textos é um campo multidisciplinar, envolvendo recuperação da informação, análises textuais, extração de informação, clusterização, categorização, visualização, tecnologias de base de dados, e Mineração de Dados (Li ET AL, 2002), que se tornou essencial, pois segundo Tan (1999), 80% das informações de uma organização estão em documentos textuais e 80% do conteúdo on-line está em formato texto (Chen, 2001).

Pode-se dizer então que a extração de conhecimento em dados não-estruturados vem ganhando cada vez mais importância, pois são as maiores fontes de informação para os tomadores de decisão.

Desta forma, o estudo de caso foi realizado com uma base de notícias textuais e objetivou encontrar o algoritmo que classifique de maneira mais eficiente este tipo de base, o qual foi apontado por meio dos cálculos de desempenho (*accuracy*) de cada algoritmo estudado.

1.1 OBJETIVOS:

1.1.1 Objetivo Geral

O principal objetivo deste trabalho é realizar a classificação de textos (Text Mining) em uma base de notícias textuais a fim de selecionar qual dos algoritmos estudados é o mais eficaz na classificação deste tipo de base textual, de acordo com seu desempenho.

1.1.2 Objetivos Específicos

Os objetivos específicos do trabalho são:

- Escolha da base de texto;
- Escolha da ferramenta para a Mineração;
- Estudo dos algoritmos para Mineração de textos;
- Definição dos algoritmos a serem estudados;
- Escolha dos algoritmos a serem aplicados;
- Preparação dos dados para aplicação;
- Aplicação dos algoritmos na base de texto;
- Análise dos resultados.

1.2 JUSTIFICATIVA

Durante os estudos realizados sobre a Mineração em textos (BARION, 2008) e (WIVES, 2002), descreve-se muito sobre a dificuldade de extrair informações e conhecimento em repositórios de dados, principalmente dados não-estruturados. Deste modo fazem-se necessários estudos nesta área de pesquisa.

A presente monografia justifica-se então por trazer resultados de melhores algoritmos para este tipo de base de textos – notícias textuais, ou seja, algoritmos que tragam informações com a maior precisão possível, de maneira que em estudos futuros, torne o processo de descoberta de conhecimento mais rápido por ter condições de se utilizar o algoritmo mais indicado para cada situação.

2 FUNDAMENTAÇÃO TEÓRICA

Com a evolução da computação, foi facilitado o armazenamento de informações, de uma pequena aplicação gerar muitos dados, dados estes muito valiosos. No entanto, na maioria das vezes esses acabam não sendo utilizados pelo fato de estarem ocultos dentro de grandes repositórios das organizações, pois os aplicativos essencialmente utilizados para consultas (planilhas eletrônicas, por exemplo) têm pequena capacidade, e apenas geram relatórios simples. Faz-se necessário então, utilizar ferramentas que auxiliarão a responder perguntas mais complexas como: ‘Quais clientes gostariam de comprar o produto B?’ ou ainda, ‘Se João comprou A e B, e Maria comprou A, seria interessante oferecer o produto B a Maria?

Segundo Gardner (1998), o primeiro passo para tornar possível a mineração de um grande volume de dados, é a realização de Data Warehousing, que tem por objetivo limpar, agregar e consolidar estes dados num repositório para que possam ser analisados por ferramentas OLAP (On-Line Analytical Processing). Essa ferramenta é composta por um conjunto de tecnologias projetadas para dar suporte ao processo decisório através de consultas, análises e cálculos nos dados corporativos (BISPO, 1998).

Com a utilização deste tipo de ferramenta, OLAP, orientada a consulta, ou seja, dirigidas pelo usuário, dificulta que sejam encontrados padrões escondidos nos dados de maneira inteligente, pois o usuário não tem a capacidade para imaginar todas as possíveis relações e associações, testando assim somente algumas hipóteses que gostariam ou não de comprovar.

Diante deste problema para análise dos dados, surgem estudos relacionados ao desenvolvimento de tecnologias automáticas para extração de conhecimento em bases de dados, referenciado na literatura como *Knowledge Discovery in Databases (KDD),* que é o processo de identificação de padrões válidos, novos, potencialmente úteis e compreensíveis obtidos a partir dos dados. (Fayyad, Piatetsky, & Smyth, 1996).

Nos últimos tempos, com o amplo uso de tecnologias avançadas para as bases de dados, as empresas vêm armazenando grandes volumes de dados em seus bancos de dados, e estes são muito valiosos à empresa, porém a maioria delas não faz uso destas ricas informações porque não as enxergam em meio à essas grandes bases. Por esse motivo criou-se a necessidade de ter técnicas que buscam transformar estes dados em conhecimento, e esses são os objetivos da área de KDD.

A “Descoberta de Conhecimento em Bases de Dados” (Knowledge Discovery in Databases, KDD) é um processo que tem por objetivo transformar os dados de empresas dos mais variados ramos, que estão armazenados em suas bases de dados, em conhecimento para ser utilizado pelas mesmas.

Para realizar o processo de KDD, são utilizados dados, que são considerados como matéria prima bruta. Estes passam a ser considerados como informações quando recebem do usuário um significado especial, e finalmente geram o conhecimento quando os especialistas de domínio geram uma regra através das informações contidas nesses dados.

Segundo Carbonell (1987), o conhecimento é definido como a informação interpretada, categorizada, aplicada, revisada, e possui um determinado valor para o usuário.

A Figura 1 representa a relação hierárquica entre dados, informação e conhecimento, considerando-se o volume e o valor que os usuários de níveis decisórios atribuem a cada um dos elementos dessa hierarquia.

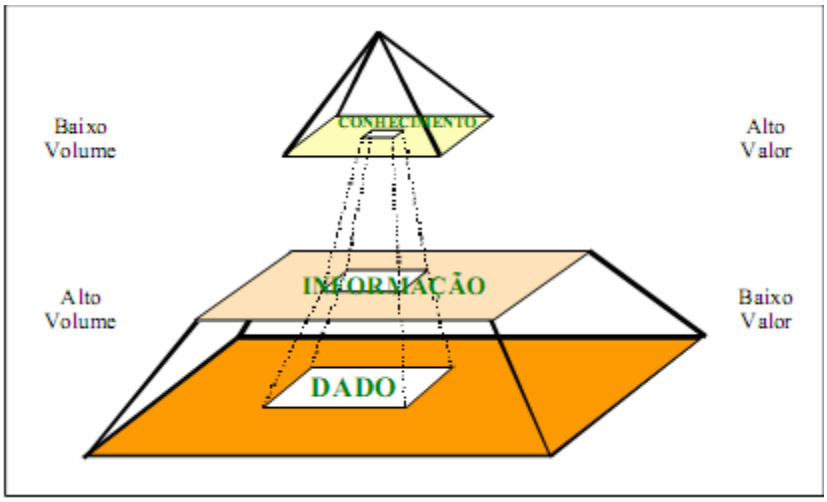

Figura 1 – Pirâmide do Processo de Conhecimento (Rocha, 1999)

Até o ano de 1995, a maioria dos autores consideravam os termos KDD e *Data Mining* como sinônimos, porém o KDD refere-se ao processo de descoberta de conhecimento útil dos dados como um todo, e já a mineração de dados é uma das etapas do Processo de Descoberta de Conhecimento, que ser refere a aplicação de algoritmos para extrair modelos dos dados. (Freitas, 2000).

O processo de KDD, segundo Fayyad (1996), é composto por cinco etapas, que são: Seleção de Dados; Pré-processamento e Limpeza dos dados; Transformação dos Dados; Mineração de Dados (Data Mining); e Interpretação e Avaliação dos resultados.

Segundo Freitas (2000), o conhecimento a ser descoberto deve satisfazer a três propriedades: Deve ser o mais correto possível; deve ser possível de ser compreendido por usuários normais, humanos; e deve ser novo, útil e interessante.

O método utilizado para realizar a descoberta de conhecimento deve apresentar características como: Ser eficiente, genérico e flexível.

2.1 O processo de Descoberta de Conhecimento

A Figura 2 mostra as etapas do processo de Extração de Conhecimento em Bases de Dados, segundo o autor Rocha:

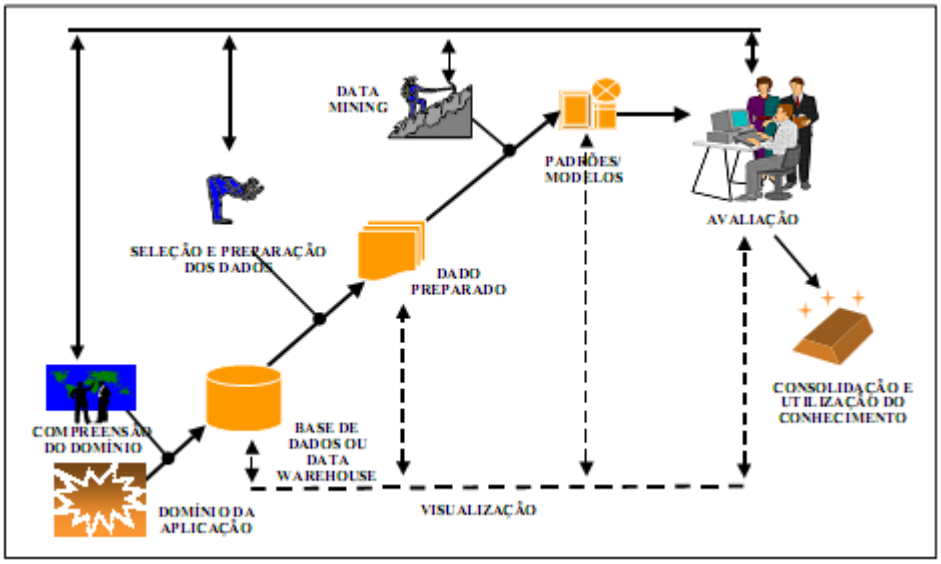

Figura 2– Etapas do processo de Extração de Conhecimento de Bases de Dados (Rocha, 1999)

Identificação do Problema

Nesta fase são definidos os objetivos e as metas que querem ser alcançadas com a mineração. É realizado também o estudo do domínio da aplicação para que sejam definidos além de metas, os critérios de desempenho, se o conhecimento extraído terá que ser compreensível a seres humano, ou se irá gerar somente um modelo caixa-preta, que dá a resposta para outro sistema, entre outras definições (Fayyad, Piatetsky-Shapiro, & Smyth 1996).

Este conhecimento dá suporte também para todas as outras etapas do processo de Extração de Conhecimento.

Na etapa de Pré-processamento ajuda na escolha do conjunto de dados mais adequado ou mais significativo em meio ao grande volume de dados. Na fase de Extração de Padrões, auxilia na escolha do modelo adequado do que vai ser minerado, e por fim na Etapa de Pós-processamento o conhecimento do especialista é importante para saber se o que foi extraído é interessante ao usuário.

2.2 Pré-processamento

Os dados que estão nas bases, geralmente não estão em um formato adequado para mineração, por isso antes da etapa de Extração de Padrões estes dados precisam passar por algumas transformações e serem adequados, assim na fase de pré-processamento passam por mudanças como: Extração e Integração, Transformação, Limpeza, e Seleção e Redução dos dados.

Extração e Integração

Os dados disponíveis podem encontrar-se em diferentes fontes, como planilhas, bancos de dados, arquivos-textos ou Data Warehouse, então, devem ser unificados e integrados em uma única base.

Transformação

Após a extração e integração dos dados, estes precisam adequar-se para que possam ser extraídos os padrões, para isso alguns dados precisam passar por transformações como resumo e transformações de tipo. Por exemplo quando um atributo do tipo ‘data’ é transformado em outro tipo para que o algoritmo de extração de padrões consiga utilizá-lo e ainda normalizações.

Limpeza

A tomada de decisão do sistema será feita baseada nas informações dos dados, estes não podem conter nenhum tipo de erro. Os dados precisam passar por técnicas de limpezas que corrijam os problemas como erros de digitação, ou erro pela leitura dos sensores.

Estas técnicas de limpeza podem ser aplicadas utilizando o conhecimento de domínio, ou também independente do domínio. (Batista, Carvalho & Monard 2000),

Seleção e Redução de Dados

O número de exemplos e de atributos que estão nas bases de dados, disponíveis para análise, se for grande demais, pode inviabilizar a utilização do algoritmo de extração de padrões, pelo espaço de memória ou tempo de processamento. Assim pode ser necessário a redução desses dados, que pode ser feita de três maneiras, segundo Weiss & Indurkhya, 1998, que são: redução do número de exemplos, redução do número de atributos, e redução do número de valores de um atributo.

A redução do número de exemplos é feita a partir de amostragem, ou seja, gera-se a amostra representativa de cada conjunto de dados. A maneira mais utilizada para este tipo de redução é a amostragem aleatória (Weiss & Indurkhya, 1998).

É de extrema importância que a amostra seja representativa e a quantidade de exemplos seja suficiente para que os modelos encontrados representem a realidade.

A redução do número de atributos é utilizada para não consumir ou aumentar muito o tempo de busca pela solução. Consiste basicamente em eliminar atributos desnecessários, porém a redução deve ser realizada com cautela, para não excluir um atributo potencialmente útil para o modelo final, e alterar assim a qualidade do conhecimento extraído. (Lee 2000)

A redução do número de valores do atributo é feita através de discretização e suavização dos valores de um atributo. A discretização consiste em substituir um valor por um agrupamento de valores, por exemplo, de 0 a 10. Já a suavização seria pegar um grupo de valores e substituí-lo por um único valor numérico. (Félix, Rezende, Monard, & Caulkins 2000).

2.3 Extração de Padrões

Esta é a etapa onde ocorre realmente a descoberta de conhecimento em meio à base de dados. Compreende a escolha da tarefa que será empregada para minerar os dados, a escolha do algoritmo e a execução desse algoritmo para extrair os padrões.

A escolha da tarefa é feita de acordo com o que se quer saber ou encontrar nos dados. Essa tarefa pode ser:

a) Tarefa Preditiva: Compreende a classificação que consiste na predição de um valor categórico, como por exemplo predizer se o cliente é bom ou mau pagador. Compreende também a regressão, onde o atributo a ser predito consiste em um valor contínuo como, por exemplo, predizer o lucro ou perda em um empréstimo. (Weiss & Indurkhya, 1998).

b) Tarefa Descritiva: Compreende as técnicas de Regras de associação, Agregação, entre outras.

- Regras de Associação:

Representam as combinações que se repetem com alguma freqüência em uma base de dados. É geralmente aplicada a bases de dados que guardam transações de compra e venda. Um exemplo comum é o da analise das transações de um supermercado, a partir desta tem-se a seguinte regra:

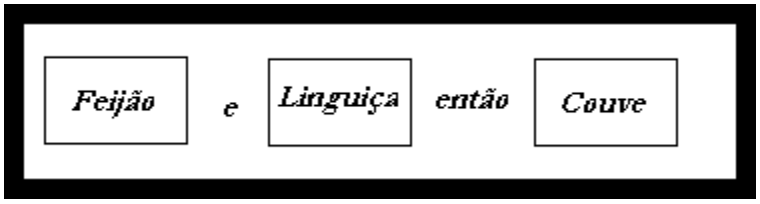

Figura 3 - Regra de associação

De acordo com a Figura 3 pode-se dizer que os clientes que compram feijão e lingüiça tendem a comprar couve. Essas regras de associação possuem um valor de suporte, e de confiança. O suporte, segundo AGRAWAL (1993), num conjunto de dados Z, Sup(Z), representa a porcentagem de transações da base de dados que contêm os itens de Z. O suporte de uma regra de associação $A \Rightarrow B$, $Sup(A \Rightarrow B)$, dado por $Sup(A \cup B)$. Já a confiança desta regra, $Conf (A \Rightarrow B)$, representa, dentre as transações que contém A, a porcentagem de transações que também contem B, ou seja, $Conf (A \Rightarrow B) = Sup(A \cup B) \div Sup(A)$.

Um dos algoritmos mais conhecidos desta técnica, é o *Apriori,* que pode gerar com um grande número de atributos, várias alternativas combinatórias entre eles. Este algoritmo realiza buscas sucessivas em toda a base de dados, mantendo um ótimo desempenho em termos de processamento. (Agrawal & Srikant, 1994).

- Clusterização ou Agrupamentos

Clusterização ou Agrupamento é a classificação não-supervisionada de dados, formando agrupamentos ou clusters (JAIN, 1999).

Um cluster é uma coleção de itens similares que ficam agrupados em um mesmo cluster (segundo um critério pré-fixado anteriormente) e diferentes dos itens dos outros clusters.

Depois de definida a tarefa a ser empregada, define-se o algoritmo para executá-la. Os algoritmos que destacam-se entre os tipos mais freqüentes de representação de padrões são: Árvores de decisão, regras de produção, modelos lineares, modelos não-lineares, modelos baseados em exemplos e modelos de dependência probabilística.

Segundo Kohavi, Sonnerfield, & Dougherty (1996) não existe um único bom algoritmo para todas as tarefas de mineração de dados. Assim pode-se escolher vários algoritmos para obter diversos modelos que serão tratados na etapa de pós-processamento.

Após definido o tipo de tarefa e o algoritmo, ou os algoritmos, aplica-se este aos dados para realmente extrair os padrões. Pode ser necessária a execução desses algoritmos por diversas vezes, dependendo da função escolhida.

2.4 Pós – processamento

Depois de descobertos os padrões é necessário avaliar se o conhecimento extraído é correto, se está de acordo com o conhecimento do especialista, dentre outros. Ao final da extração, os algoritmos encontram muitos padrões, e é necessário averiguar quais destes são interessantes ao usuário (Liu & Hsu 1996).

3 MINERAÇÃO DE TEXTOS (Text Data Mining)

A mineração em bases textuais é definida como uma extração não trivial de informações, não explícitas, de grandes bases textuais, previamente desconhecidas e, potencialmente úteis. (Feldman & Hirsh 1997).

Segundo Passos (2006), a Mineração de Textos é um campo multidisciplinar que envolve conhecimentos das áreas de Informática, Estatística, Lingüística e Ciência Cognitiva.

Minerar textos pode então ser entendida como uma maneira de extrair informações de qualidade/ conhecimento de um texto não-estruturado.

Segundo Tan (1999), 80% das informações de uma companhia estão contidas em documentos textuais, e ainda Chen (2001) diz que 80% do conteúdo on-line está em formato texto. Assim conclui-se que para a tomada de decisão dentro de uma empresa, só são utilizados 20% das informações.

A mineração de textos é também conhecida como uma extensão do *Data Mining*, focada em análise de textos, chamada *Text Data Mining*. Esta surge então da necessidade de descobrir, de forma automática, informações armazenadas em textos (dados desestruturados), para assim utilizá-la na tomada de decisão dentro de uma organização, do mesmo modo que se utiliza os dados estruturados.

3.1 O que é considerado e o que não é considerado Mineração de Textos

O processo de descoberta de conhecimento em textos se baseia em um conjunto de métodos usados para navegar, organizar, achar e descobrir informações em bases textuais. (Feldman & Hirsh 1997).

A Mineração de Textos não é um mecanismo de busca de informações, pois nesta o usuário já sabe o que quer encontrar, por isso não é considerado Mineração, já que esta busca conhecimento novo.

3.2 O processo de Mineração de Textos

Ao se iniciar o processo de Mineração de Textos deve-se levar em consideração uma questão importante: Tipo de abordagem de dados que será utilizada para realizar essa extração de conhecimentos.

3.2.1 Tipos de Abordagens de Dados

Segundo Ebecken & Lopes (2003) os dois tipos de abordagens para mineração de textos são a Análise Semântica e a Análise Estatística, a primeira baseia-se na funcionalidade dos termos do texto, ou seja leva em consideração a seqüência em que os termos estão dispostos no texto, para analisar sua função dentro do mesmo.

Segundo Cordeiro (2005) esta técnica caracteriza-se por verificar a importância de cada termo dentro de sua estrutura nas orações.

Já a análise estatística tem por base o número de vezes que o termo aparece no texto, ou seja, a importância é dada ao termo de acordo com sua freqüência.
Depois de definida essa questão de abordagem, passa-se então a próxima etapa que é a preparação dos dados.

3.2.2 Preparação dos Dados

Nesta etapa são definidos os dados que irão formar a base de textos, em que será realizada a Mineração. Consiste numa consulta que é feita pelo usuário para que documentos importantes sejam encontrados, conhecida como Recuperação da Informação (RI).

Segundo Ebecken (2003), a Recuperação da Informação pode ser considerada o primeiro passo da Preparação dos dados.

Vários métodos de RI foram criados, o que gerou uma taxonomia de modelos (WIVES, 2002), sendo eles:

a) *Modelo Booleano*

Utiliza os conectivos de boole (and, or e not) para realizar as buscas, e como resultado dessas, pode retornar uma intersecção, se utilizado o conectivo *and,* uma união utilizando o conectivo *or* ou ainda uma retirar partes de um conjunto, com a utilização do conectivo *not.*

b) *Modelo Espaço-Vetorial*

Este modelo foi desenvolvido por Gerald Salton, para ser usado num SRI chamado SMART. Neste modelo, os documentos possuem um vetor de termos que são associados a um peso, que indica o grau de importância deste no documento. Os termos que o documento não apresentar recebem peso zero e os outros são calculados através de uma fórmula de identificação de importância. De acordo com o valor do calculo final de seus termos este documento é considerado importante ou não.

c) *Modelo Probabilístico*

O modelo probabilístico recebe esta denominação pelo falo de trabalhar com conceitos da área de probabilidade e estatística. É também conhecido como modelo Bayesiano. Utilizam vetores de termos, porém não apresenta termos com relevância zero, mas sim muito baixa.

Neste modelo, busca-se saber a probabilidade de um documento D, ser ou não, relevante para uma consulta Q.

d) *Modelo Busca-Direta*

Este modelo tem seu uso recomendado para pequenas coleções de documentos, pois realiza a busca por strings nos documentos. Por isso é também conhecido como modelo de busca de Padrões.

e) *Modelo Aglomerado (Clusters)*

Neste modelo, são agrupados documentos similares, formando clusters. Esta similaridade é definida através da quantidade de palavras que os documentos contêm.

Quando é feita uma consulta pelo usuário, o modelo retornará todos os documentos do mesmo grupo.

f) *Modelo Conceitual ou Contextual*

As consultas são feitas levando em consideração o contexto tanto dos documentos, quanto da busca do usuário e não mais somente através dos termos presentes. Cada palavra possui então um grau de importância que varia de acordo com o contexto de cada documento.

Depois de escolhido o modelo mais apropriado passa-se então a fase de indexação e normalização dos textos.

3.2.3 Indexação e Normalização

Segundo Salton e Mcgill (1983) a indexação é o processo pelo qual as palavras contidas no texto são armazenadas em uma estrutura de índices para viabilizar a pesquisa de documentos através das palavras que ele contém, ou seja, consiste em pegar as palavras chaves do texto formando uma estrutura que é denominada índice. Das duas maneiras que indexação pode ser feita - manualmente ou automaticamente – deve produzir o mesmo resultado, sendo este uma lista com os termos chave do documento.

A indexação automática possui as seguintes etapas: Identificação de termos, Remoção de stopwords, Normalização e padronização de vocabulário, Seleção de termos relevantes. Estas formam a estrutura de índice do documento. Observe estas etapas na Figura 4:

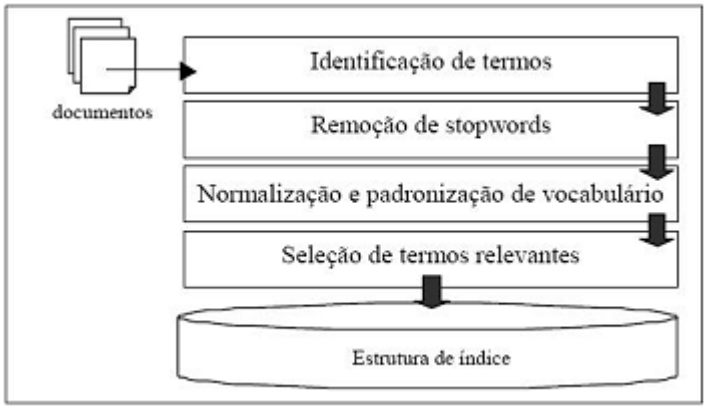

Figura 4 - Etapas do processo de Indexação Automática (WIVES, 2002)

a) Identificação de termos

Dividi-se em: identificação de termos simples e identificação de termos compostos.

Identificação de termos simples: Consiste em identificar as palavras do documento e excluir símbolos e caracteres de controle de arquivo ou de formatação (WIVES, 2002). Podem ocorrer nesta fase também transformações como: converter todos os caracteres para maiúsculo ou minúsculo, padronizar números e datas, eliminar tabulações, correções ortográficas e normalização de vocabulários através de um dicionário de sinônimos.

Identificação de termos compostos: Não deve esquecer-se durante esta fase dos termos que aparecem no texto de forma composta. A identificação desses termos pode ser feita por co-ocorrência, onde o sistema verifica expressões que ocorrem com freqüência nos documentos e apresenta uma lista para validação ou pode-se também utilizar um dicionário de expressões. (WIVES, 2002).

b) *Remoção de stopwords*

Os *stopwords* são palavras menos importantes encontradas na maioria dos textos. São considerados menos importantes porque não influenciam no assunto principal do documento. Estes formam a *stopilist*, uma lista que contem todas as palavras que podem ser removidas do texto-fonte, geralmente composta por artigos, preposições, pronomes, entre outras.

A remoção dos *stopwords* consiste em comparar os termos que compõem o documento em estudo e os termos da *stoplist*, e cada vez que se encontra um igual, é automaticamente eliminado pelos algoritmos de remoção. (FELDMAN, 1995).

A Figura 5 mostra um exemplo de como é feito o teste em um termo do índice para decidir se é ou não um *stopword* e se deve ser eliminado do índice de palavras do documento.

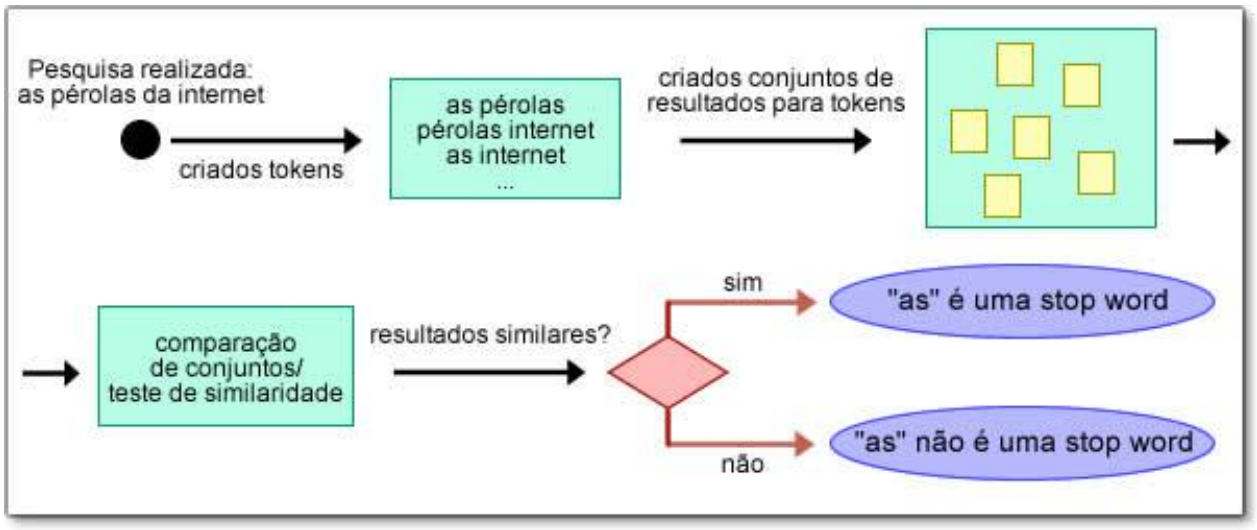

Figura 5 - Diagrama de teste de stopwords

Este processo é necessário para retirar do texto palavras irrelevantes, diminuindo assim o tamanho das estruturas de indexação e facilitando o processo de mineração.

c) *Normalização Morfológica*

Nesta fase do processo são eliminadas as variações morfológicas das palavras, sendo eliminadas os sufixos e prefixos das palavras, através da técnica de lematização ou *stemming*. São eliminadas também características de gênero, número e grau, reduzindo dessa maneira boa parte do índice.

As palavras construção e construir são transformadas em uma mesma cadeia: constru.

Com esta associação torna-se possível fazer com que consultas por palavras com o mesmo radical recuperem os mesmos resultados, o que na maioria das vezes é

valido, pois documentos relacionados com palavras derivadas são relevantes para a mesma coisa. (FELDMAN, 1995)

Depois de realizado o processo de remoção de palavras de ligação (*stopwords*) e remoção de sufixos (*stemming*) da frase: “A destruição das florestas tropicais da Amazonia” ela ficará da seguinte forma “*destru florest tropic amazon”.*

d) Cálculo da Relevância

O calculo da relevância de uma palavra pode ser feito levando em conta: sua freqüência ou sua posição sintática em relação ao texto.

A maneira mais utilizada é a relevância por freqüência, onde cada palavra recebe um peso em relação àquele documento, e este indica a importância da palavra no texto. Este peso pode ser calculado de acordo com sua:

Freqüência Absoluta: Quantas vezes o termo aparece no documento.

Freqüência Relativa: Esta analisa o tamanho do documento, dividi-se a freqüência absoluta do termo pela quantidade total de palavras do texto.

Freqüência Inversa de documentos: Termos que aparecem em poucos documentos possuem um maior grau de importância. Dividi-se o número de vezes que o termo aparece em um documento, pelo número de documentos que ele aparece.

É uma medida estatística usada para avaliar o quão importante uma palavra é dentro de cada texto.

De acordo com estudos, a freqüência inversa de documentos mostra melhores resultados que a Freqüência Relativa e Freqüência Absoluta

e) Seleção de Termos

Corresponde a fase em que são retiradas dos documentos, as palavras mais importantes, de acordo com seu peso, já calculados.

Algumas técnicas para selecioná-las são:

- Filtragem baseada no peso:

É estabelecido um peso limiar (threshold), e são eliminados todos os termos que ficam abaixo deste.

- Seleção baseada no peso do termo:

Depois de filtrados os termos, ainda restam vários termos, então estes passam por uma nova redução, chamada de truncagem. Esta estabelece um número máximo de características a serem utilizadas para caracterizar um documento e todas as outras são eliminadas. (WIVES, 2002). As características devem estar ordenadas de acordo com seu peso, ou grau de importância, para serem selecionadas as *n* primeiras, ou seja as mais relevantes.

- Seleção por análise de Co-ocorrência:

Analisa os pesos dos termos que compõem o documento, levando em consideração os termos que ocorrem em mais de um documento ao mesmo tempo (co – ocorrência), para verificar o grau de relacionamento entre os termos.

Loh et. al (1998), propõe para esta analise as seguintes fórmulas, mostradas nas Figuras 6 e Figura 7.

$$d_{ij} = tf_{ij} \times \log\left(\frac{N}{df_j}\right), \text{ onde:}$$

Figura 6 - Fórmula para analise de Co-ocorrência

N representa o número total de documentos considerados, tf_{ij} é a freqüência da palavra j no documento i e df é a freqüência inversa de documentos.

A segunda fórmula avalia os resultados gerados pela fórmula anterior, detectando as relações entre as palavras. A fórmula é mostrada na figura 7.

$$\text{Co-ocorrência} = \frac{\sum_{i=1}^{n} d_{ijk}}{\sum_{i=1}^{n} d_{ij}}, \text{ para } d_{ijk} = tf_{ijk} \times \log\left(\frac{N}{df_{jk}}\right), \text{ onde:}$$

Figura 7 - Formula para análise dos resultados da Co-ocorrência

tf_{ijk} representa o número de ocorrências de ambas as palavras (j e k) no seu documento i (o menor número de ocorrências entre as palavras deve ser escolhido), d_{ijk} representa o número de documentos (em uma coleção N) no qual as palavras j e k ocorrem ao mesmo tempo. Em alguns casos, duas palavras podem estar relacionadas entre si em mais de um contexto, e em cada contexto deve existir um grau diferente de relação.

- Seleção por análise de linguagem natural:

Aplica técnicas de análise sintática e semântica. A primeira pode ser realizada a partir de um dicionário ou gramática bem definida. Nesta é possível de se influenciar o peso dos termos do documento, a fim de torná-los mais ou menos relevantes.

Segundo WIVES (2002), esse tipo de análise não funciona em 100% dos casos, e a maioria dos Sistemas de Recuperação de Informação não faz uso dela.

A análise semântica é baseada no princípio que as partes mais importantes do documento já possuem demarcação, formatação específica. A utilização deste tipo de abordagem vêm crescendo utilizando novos formatos (HTML e XML, por exemplo, que já permitem incluir algum tipo de marca – *tag*) de armazenamento de textos.

Esta técnica só funciona se a pessoa que elabora os documentos coloque essa marca e isto nem sempre é feito.

f) *Análise dos resultados*

Nesta etapa de análise de resultados verifica-se as respostas do sistema de recuperação de Informações, para saber se este funcionou como deveria.

De acordo com Wives (2002), as métricas mais conhecidas para esta análise são: *Recall, Precisin e Fall-out.*

- Recall

A técnica de Recall mede a capacidade que o sistema apresenta em recuperar os documentos mais relevantes para o usuário de acordo com o termo da pesquisado. Utiliza a seguinte fórmula:

$$recall = \frac{n\text{-}recuperados\text{-}relevantes}{n\text{-}possiveis\text{-}relevantes}$$, onde:

n-recuperados-relevantes: é o número de documentos relevantes recuperados e
n-possiveis-relevantes: é o total de documentos relevantes do sistema.

- Precision

A precision mede a capacidade do sistema de não retornar documentos irrelevantes às consultas do usuário, através da fórmula:

$$precision = \frac{n\text{-}recuperados\text{-}relevantes}{n\text{-}total\text{-}recuperados}$$, onde:

n-recuperados-relevantes: é o número de documentos relevantes recuperados e
n-total-recuperados: é o total de documentos do Sistema.

- Fall-out

Esta técnica analisa a possibilidade do aumento dos documentos irrelevantes poder ser modificada pelo aumento ou diminuição das bases de dados, ou seja, mede a quantidade de documentos irrelevantes, permitindo que se identifique se a quantidade de documentos relevantes permanece a mesma quando o número de documentos varia. Utiliza a fórmula a seguir:

$$fall\text{-}out = \frac{n\text{-}recuperados\text{-}irrelevantes}{n\text{-}possiveis\text{-}irrelevantes}$$, onde:

n-recuperados-irrelevantes: é o número de documentos irrelevantes recuperados
n-possiveis-irrelevantes: é o número total de documentos irrelevantes do sistema.

3.3 Classificação de Textos

A classificação é uma tarefa de mineração, que tem como objetivo classificar textos ou dados, em classes já determinadas, ou seja, vincular um documento específico a um modelo pré-definido.

Os documentos são classificados a partir de características do texto, como termos ou palavras presentes nos documentos. Baseia-se na análise prévia de um conjunto de dados de amostragem ou dados de treinamento, contendo objetos corretamente classificados.

Um modelo de classificação de dados poderia incluir a seguinte regra: *Clientes da faixa econômica B, com idade entre 50 e 60 anos e sexo feminino são maus compradores*. O atributo meta a ser definido por esta classificação é a qualidade do cliente, de bom ou mau comprador, que é definida por regras já definidas que levam em consideração a faixa de idade e sexo que o fazem ser considerados bons ou maus compradores.

Existem inúmero algoritmos para classificação: Decision Tree, Decision Stump, SVM (Support Vector Machine), K-NN (K- Nearest Neighbor) Chaid, Random tree, perceptron, RVM (Relevance Vector Machine) Naives Bayes, entre outros, no entanto para este trabalho em específico são abordados quatro deles: Decision Tree (Árvores de Decisão), Naives Bayes, K-NN e SVM, pois segundo Beppler (2010) & Oliveira (2001) são os mais conhecidos e utilizados.

3.3.1 Árvores de Decisão

Segundo Quinban (1993), métodos de Árvores de Decisão representam um tipo de algoritmo de aprendizado de máquina que utilizam uma abordagem *dividir-para-conquistar* para classificar documentos usando uma representação baseada em árvores.

Com base nos registros do conjunto de treinamento, uma árvore é montada e, a partir desta árvore, pode-se classificar a amostra desconhecida. A classificação de uma nova amostra é feita percorrendo os ramos e nós da árvore de acordo com os valores

dos atributos da amostra desconhecida. Este algoritmo permite uma análise mais detalhada levando em consideração o valor de cada atributo.

Uma árvore de decisão é composta por:

- Nós de decisão, e cada um deles contêm um teste de atributo;
- Ramos descendentes, que correspondem a um possível valor de atributo;
- Folhas que estão associadas a uma decisão; e
- Percursos na árvore (da raiz até a folha), que correspondem a uma regra de classificação.

Na Figura 8 é apresentado um exemplo de arvore de decisão, que expõe se o clima está ou não favorável para jogar tênis.

Figura 8 - Arvore de Decisão - Jogar Tenis (MUNIZ, 1999)

3.3.2 Naives Bayes

O cálculo é baseado no Teorema de Bayes, de Thomas Bayes, um ministro inglês do século XVIII, por isso possui o nome de Naives Bayes.

É um dos mais simples classificadores, utilizando a construção de modelos probabilísticos, e suas probabilidades são estimadas pela contagem da freqüência de cada valor de características para as instâncias dos dados de treino. (Langley, Iba e Thompson, 1992). Dado um novo documento, o classificador estima a probabilidade de este pertencer a uma classe específica, levando em conta o produto das probabilidades condicionais individuais para os valores característicos do documento. (Domingos & Pazzani, 1997).

Ao treinar o classificador Naives Bayes calculamos uma distribuição geradora Pr(d|c) para cada classe C € {-1, 1}. Na fase de classificação simplesmente calculamos

qual a distribuição tem a maior probabilidade de gerar cada documento. Este algoritmo estima a probabilidade de cada classe em função dos valores dos atributos (THOMAS, 1997).

A classe prevista é aquela em que é maior a sua probabilidade condicionada aos valores observados nos atributos, ou seja, pretende-se saber para cada classe *d* a probabilidade *P.*

3.3.3 K-NN

Este tipo de algoritmo é conhecido como algoritmo de aprendizagem preguiçosa, pois não mostra explicitamente um modelo, como o decision tree - que gera a árvore de decisão, dificultando assim o entendimento.

É um classificador baseado em Instância, assim utiliza K pontos mais próximos, (chamado também de Vizinho mais próximo) para realizar a classificação.

É um método para classificar objetos baseados nos exemplos mais próximos dos exemplos de treinamento, para isso leva em consideração suas características. Utiliza a idéia básica do vizinho mais próximo: *Se ele anda com um pato, grasna como um pato, então provavelmente ele é um pato.* (TAN, STEINBACH & KUMAR, 2005).

3.3.4 SVM

O algoritmo de aprendizagem de máquina SVM, tem como objetivo a determinação de limites de decisão que produzam uma separação ótima entre essas classes por meio de minimização de erros. Consiste em uma técnica computacional de aprendizado para problemas de reconhecimento por padrão (VAPNIK, 1995).

O processo de treinamento consiste em treinar um classificador de forma que este aprenda um mapeamento $x \rightarrow y$ por meio de exemplos (classes) de treinamento {xi, yi} de forma que a máquina seja capaz de classificar um exemplo (x, y) ainda não visto que siga a mesma distribuição de probabilidade (P) dos exemplos de treinamento.

Segundo SMOLA et al(1999), os classificadores gerados por uma SVM em geral alcançam bons resultados de generalização. A capacidade de generalização de um classificador é medida por sua eficiência na classificação de dados que não pertençam ao conjunto utilizado em seu treino.

3.4 Métodos de Validação

A tarefa de classificação consiste em construir um modelo de algum tipo que possa ser aplicado em dados não classificados visando categorizá-los em classes. (Harrison, 1998). Deste modo, para que se possa realizar a classificação de um conjunto de dados é necessário que o algoritmo seja anteriormente treinado com um conjunto de dados já classificados, chamado conjunto de treinamento, sendo assim, geralmente usa-se parte da base para treinar o algoritmo e o restante para realização de testes. Essa separação de dados em conjunto de teste e treinamento é uma parte importante da avaliação de modelos de Mineração. Geralmente quando se particiona um conjunto de dados em um conjunto de treinamento e um conjunto de testes, a maior parte dos dados é usada para treinamento e o restante, ou seja, a menor parte é usada para testes.

Depois que um modelo for processado usando o conjunto de treinamento, testa-se este modelo fazendo previsões contra o conjunto de testes.

Nesta seção são apresentados os conceitos de Validação utilizando dois dos métodos mais utilizados para avaliar algoritmos de classificação, Bootstrap e Validação Cruzada, pois permitem determinar qual o melhor algoritmo em termos de taxa de acertos para um conjunto de dados específico.

3.4.1 Bootstrap

A Validação Bootstrap foi proposta por Efron e Tibshirani (EFRON e TIBSHIRANI, 1993) é considerada uma das melhores ferramentas para estimar o desempenho de um conjunto de dados pequeno.

As amostras são aleatoriamente distribuídas entre os conjunto de treinamento e teste, durante a execução e não são duplicadas amostras, sendo assim as amostras usadas no teste não serão escolhidas para treinamento. No entanto podem ser repetidas dentro de um mesmo conjunto.

Na próxima execução os dados são redistribuídos e novamente escolhidos aleatoriamente, podendo ocorrer sobreposição dos dados usados na execução anterior, o que não ocorre na validação cruzada.

Esta distribuição pode ser observada na Figura 8, que possui um exemplo de bootstrap.

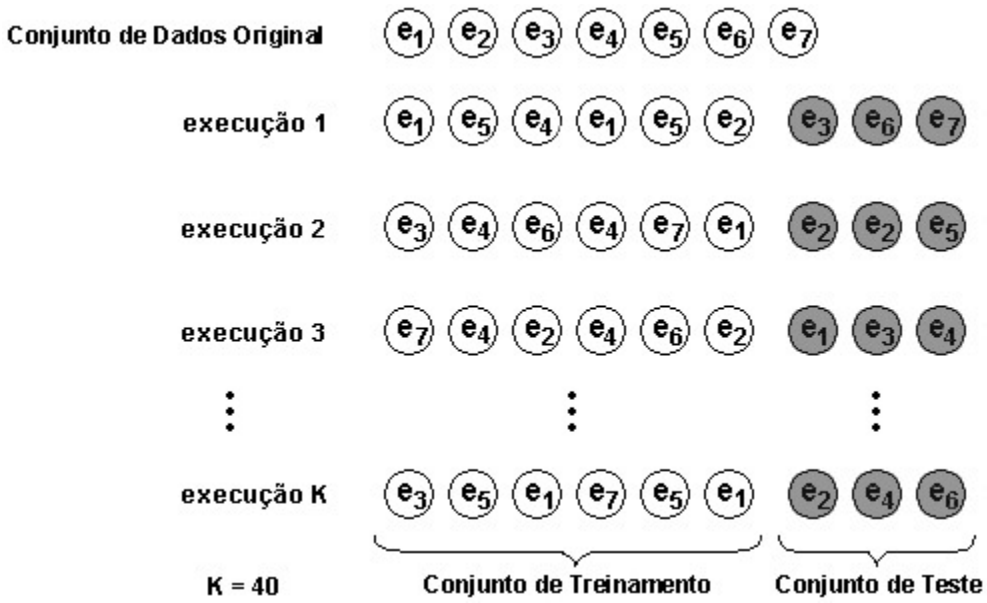

Figura 9 - Divisão de amostras Bootstrap (LOPES, 2003)

3.4.2 Validação Cruzada

Segundo Kohavi (1995), a Validação Cruzada é uma das técnicas mais aplicadas na verificação da estabilidade dos modelos. Este tipo de validação consiste em dividir o conjunto de amostras em: Conjunto de treinamento de conjunto de testes, como mostra a Figura 9.

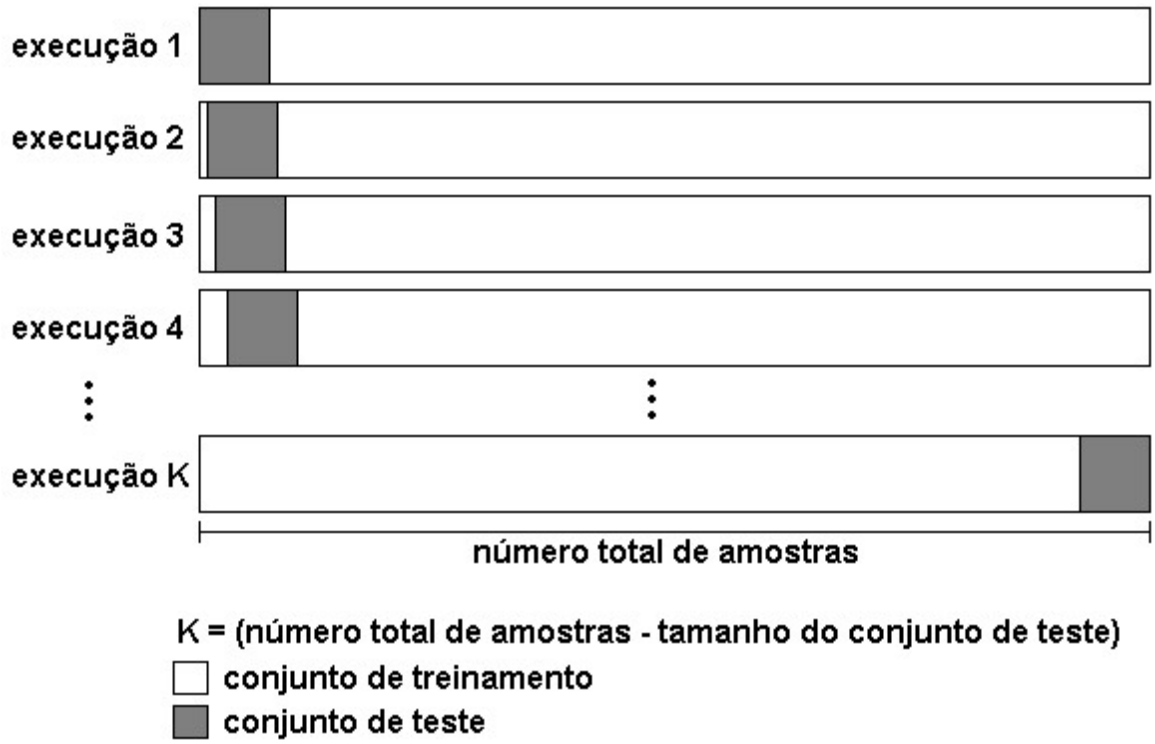

Figura 10 - Divisão das amostras usadas pela validação cruzada (LOPES, 2003)

A grande diferença deste tipo de validação em relação à validação bootstrap é que todas as amostras são usadas para treinamento e teste, como pode ser observado na figura 9. Na primeira execução o conjunto de teste é constituído pelas primeiras vinte e duas amostras e o treinamento pelas restantes. Na segunda execução, o conjunto de teste contém desde a segunda até a vigésima terceira amostra e o conjunto de treinamento as restantes, sendo este processo repetido até que tenham sido validadas todas as amostras do conjunto.

4 DESENVOLVIMENTO

Esta seção explicará passo a passo como foi a realizada a Mineração de Textos da base e como foram realizados os testes com os algoritmos, dentro da Ferramenta

4.1 Materiais e Métodos

Inicialmente este trabalho apresenta uma revisão bibliográfica de fontes secundárias, que são as principais bibliografias já publicadas referentes ao tema de estudo (Mineração de textos), tais como jornais, revistas, artigos, livros, monografias, dissertações, teses, entre outros.

De acordo com o critério de classificação de Gonsalves (2001), a monografia possuirá a seguinte configuração metodológica:

a) Segundo o objetivo:

É uma pesquisa exploratória, que tem por finalidade compreender o fenômeno que está sendo estudado.

b) Segundo o método de procedimento de coleta utilizado:

É um estudo de caso, pois analisará uma unidade específica de estudo, e realizará um exame profundo sobre o problema investigado.

c) Segundo as fontes de informação:

É uma pesquisa bibliográfica e documental, pois se utiliza fundamentalmente das contribuições de diversos autores sobre um determinado assunto, e de matérias que não receberam ainda nenhum tipo de tratamento analítico.

4.2 Base de textos

A base de textos utilizada para o desenvolvimento do trabalho é composta de corpos de textos, notícias que formam três categorias, sendo elas: Política, Esportes e Educação.

Esta base foi formada utilizando-se as notícias publicadas no site da UOL Notícias (http://noticias.uol.com.br), no site Terra Notícias (http://noticias.terra.com.br), e ainda no IG Noticias (http://ultimosegundo.ig.com.br) entre os meses de março a novembro de 2010. Foram montados arquivos contendo cada notícia separados em grupos de acordo com sua classe: Política, Esportes e Educação.

A coleta de notícias em cada site foi feita de forma aleatória, e a quantidade de notícias de cada um dos sites é de aproximadamente 50% do site da UOL, 25% do Terra e 25% do site de notícias IG. A quantidade de notícias de cada assunto foi dividida igualmente: 40 notícias de cada categoria, totalizando a base.

A base compõe-se de 120 arquivos de textos, distribuídos entre as três categorias, nos quais foram aplicadas as tarefas de limpeza e transformados em dados para serem classificados nas categorias: política, esportes e educação.

4.3 Ferramenta: Rapidminer

A ferramenta utilizada para realizar os processos de limpeza e seleção de termos, e também para realizar a mineração, é o *Rapidminer*, em sua versão 5, que é uma ferramenta *open-source* criada na Alemanha, anteriormente chamada *Yale.* Possui interface gráfica ao usuário e scripts baseados em XML, e um interpretador para KDD.

É desenvolvida sob a plataforma Java, facilitando assim a integração com outras aplicações sob esta arquitetura. Possui incorporada toda a biblioteca Weka e pode ser integrada com bibliotecas Java Database Connecttivity (JDBC), possibilitando a conexão diretamente ao banco de dados, para aplicação de algoritmos de Mineração de Dados.

A Figura 10 mostra o ambiente de trabalho da ferramenta Rapiminer.

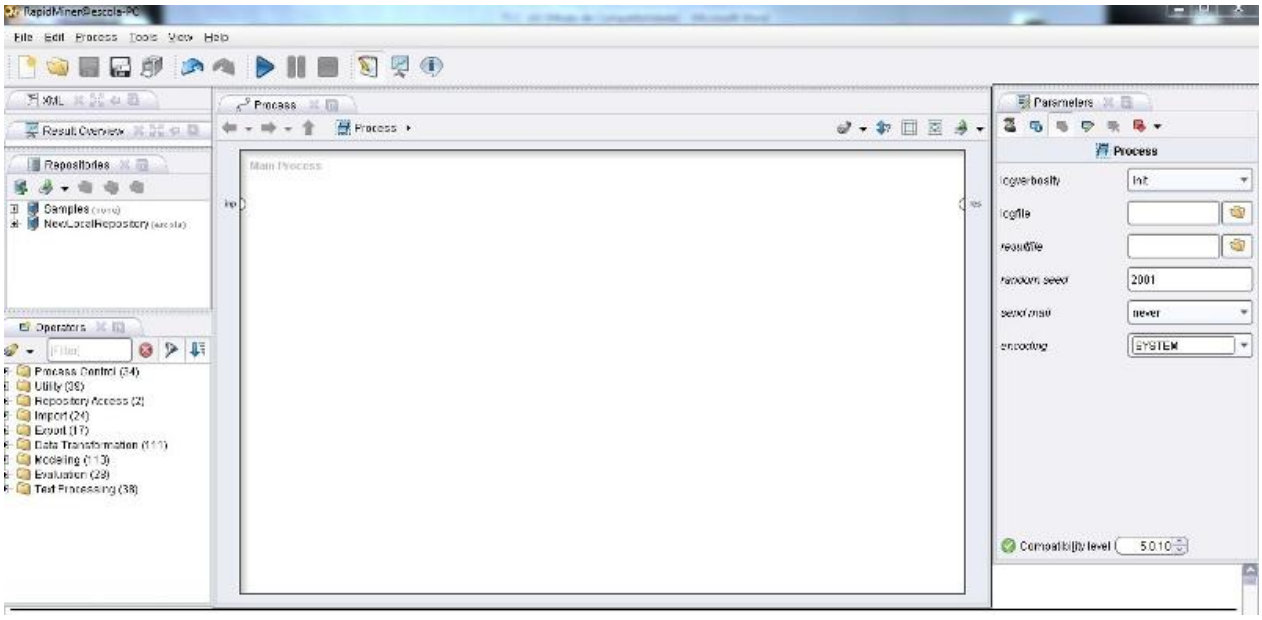

Figura 11 - Ferramenta Rapidminer

Pode-se observar na figura acima, que sua interface facilita a sua utilização, pois é intuitiva, os componentes e funções da Ferramenta estão bem categorizados, o que ajuda muito na criação de processos para sua utilização.

Esta ferramenta foi escolhida, pelo fato de ser uma ferramenta livre, e de fácil entendimento de suas funções, além das vantagens de se poder integrar diretamente ao Banco de Dados.

4.4 Tipo de abordagem dos dados

O tipo de abordagem de dados utilizada neste trabalho, é a Análise Estátistica, levando em consideração, a freqüência do termo, dentro dos documentos analisados.

4.5 Pré-processamento

Antes de realizar a etapa de extração de conhecimento da base textual, foi necessário que esta passasse por alguns mecanismos de limpeza e seleção dos dados mais importantes dos textos que compunham a base. Esta etapa é fundamental para

remover ruídos e preparar os dados para em seguida ser aplicada as técnicas de extração de conhecimento.

Para isto foi montado um processo utilizando componentes da ferramenta Rapidminer. Este processo compôs-se dos seguintes passos:

4.5.1 Leitura dos dados textuais

O primeiro passo para realizar o pré-processamento da Base Textual foi realizar a leitura dos textos que compõem a base, sendo utilizado para este processo o componente que faz essa leitura, denominado *Process Documents from files,* como mostrado na Figura 12. O componente recebeu as entradas de dados (textos) através de uma lista que aponta para os diretórios onde estão os textos. Neste componente é também especificado, como vai ser criado o vetor de palavras, que pode ser por Freqüência Absoluta, Freqüência Relativa e Freqüência inversa do termo, já especificado anteriormente na seção que trata das etapas da Mineração de Textos. Neste caso foi especificada a criação do vetor de acordo com a Freqüência inversa do termo, pelo fato de vários estudos apontarem para este, como mais eficiente forma de criar os vetores.

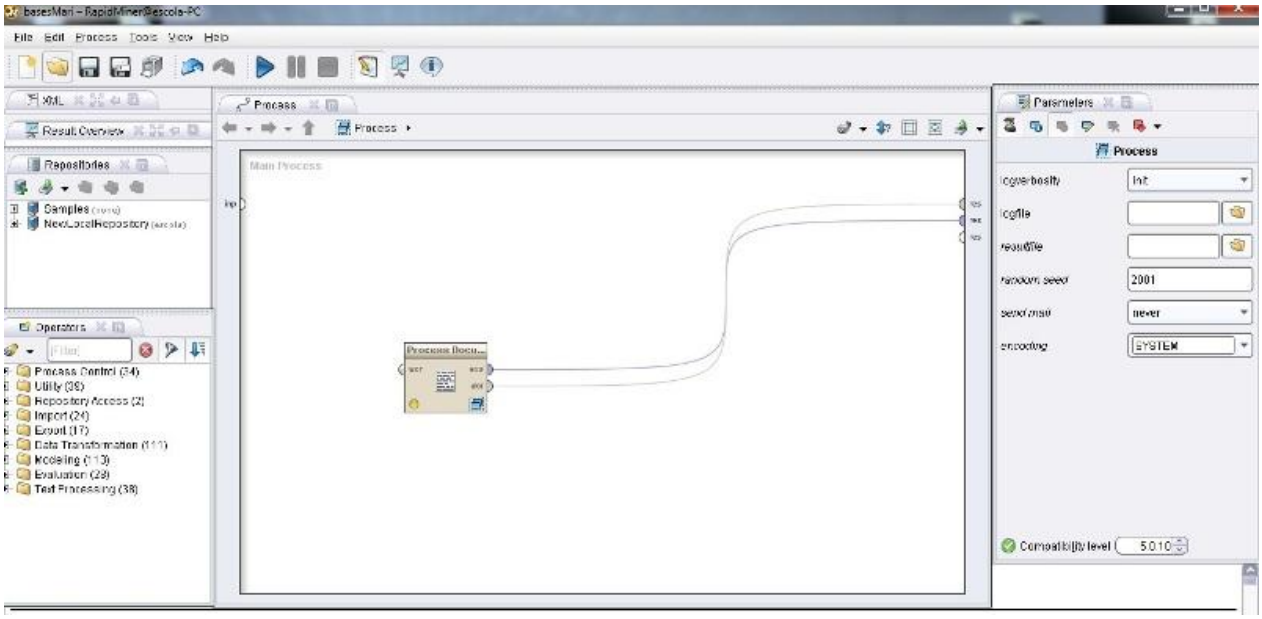

Figura 12 – Componente de leitura da base textual

Os próximos passos deste processo ficam encapsulados no componente que recebe os dados textuais. Sendo eles descritos nas próximas subseções:

4.5.2 Divisão do texto em termos

Depois de já lidos os textos, fez-se necessário dividir os textos em palavras para a formação do vetor. Nesta fase do processo foi utilizado o componente *Tokenize,* que separa o texto em palavras para a formação do vetor, este levará em consideração para demarcar cada palavra, as *non-letters* (não letras), ou seja, os espaços presentes no texto, gerando assim uma lista de palavras conhecida como índice.

4.5.3 Padronização dos caracteres

Nesta etapa o objetivo é padronizar os caracteres do texto, transformando todos para maiúsculo ou minúsculo, utilizando o componente *Transform Cases.*

Este passo é muito importante, pois na hora de realizar a remoção de termos do vetor, só serão removidas as palavras que estiverem com a grafia idêntica à passada na lista de parâmetros para remoção.

Neste caso específico, todos os termos foram transformados para minúsculo.

4.5.4 Remoção de stopwords

Este passo remove os caracteres de menor importância ou mais comuns a todos os textos em português, utilizando-se o componente *Filter Stopwords Dictionary,* que importa uma lista, conhecida como stoplist, que é padrão para qualquer processo de remoção destas palavras, não sendo diferenciada para cada tipo de texto. Esta lista é composta por pronomes, artigos, preposições, advérbios e pontuação. Os termos desta lista foram comparados com os termos do índice criado anteriormente, assim foram eliminadas as palavras comuns a todos os textos, que não agregam importância no significado dos textos, restando assim somente as palavras que realmente dão significado aquele texto.

4.5.5 Normalização Morfológica

Esta etapa consiste em eliminar prefixo e sufixo, características de gênero, número e grau, restando somente o radical de cada palavra. Este passo é importante para que cada palavra seja colocada no índice apenas uma vez, pois, por exemplo, se um texto contiver as palavras cabelo e cabeleireiro, será extraído apenas o radical delas, sendo assim, restará um único termo, que faz referencia as duas palavras citadas, diminuindo assim o tamanho do vetor e acelerando o processo de busca de informações.

Para realização desta normalização utilizou-se o componente *Stem (Snowball)*, com o atributo de linguagem para Português, pois a base é composta por textos em Português.

Para melhor entendimento deste processo, na Figura 13, é demonstrada, a disposição dos componentes na ferramenta.

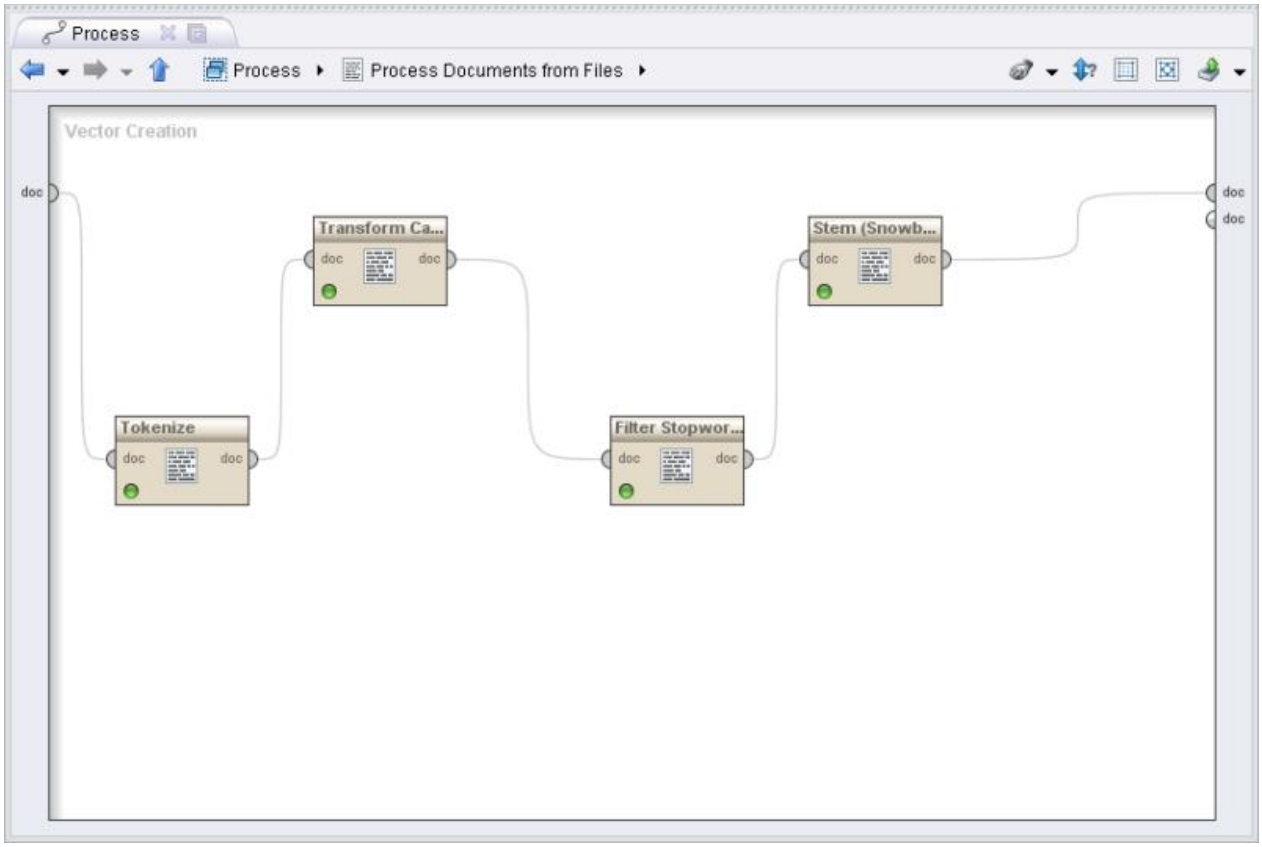

Figura 13 - Indexação e Limpeza da base

O resultado de todo este processo, gera a ExampleSet, que mostra a base pré classificada em Educação, Esporte e Política e a WordList, uma tabela contendo as seguintes colunas: Word (que são as palavras), Total Occurrences (quantas vezes determinado termo foi encontrado na base), Document Ocurrences (em quantos documentos apareceu), Política (quantas vezes o termo foi encontrado nos textos de política), Esporte (quantas vezes apareceu nos textos de esporte) e Educação (quantidade de vezes que foi encontrado nos textos de educação). A Figura 14 mostra

o ExampleSet gerado pela ferramenta, e a Figura 15 apresenta um exemplo de WordList.

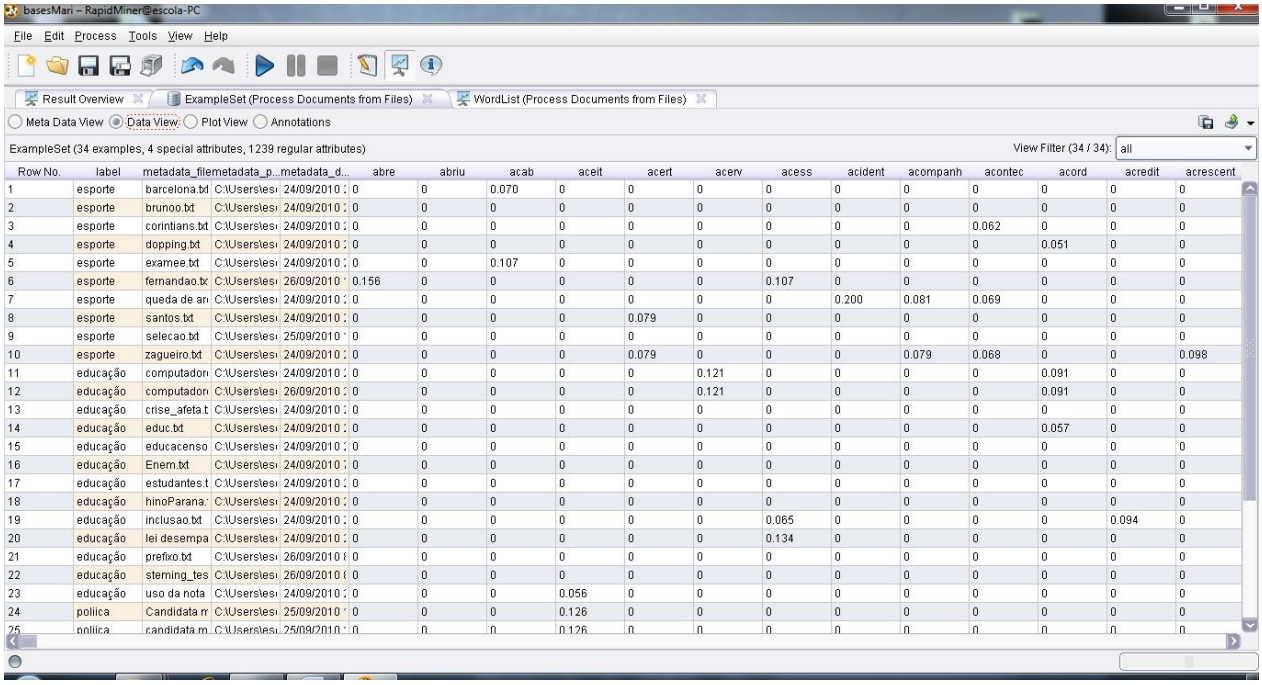

Figura 14 – ExampleSet

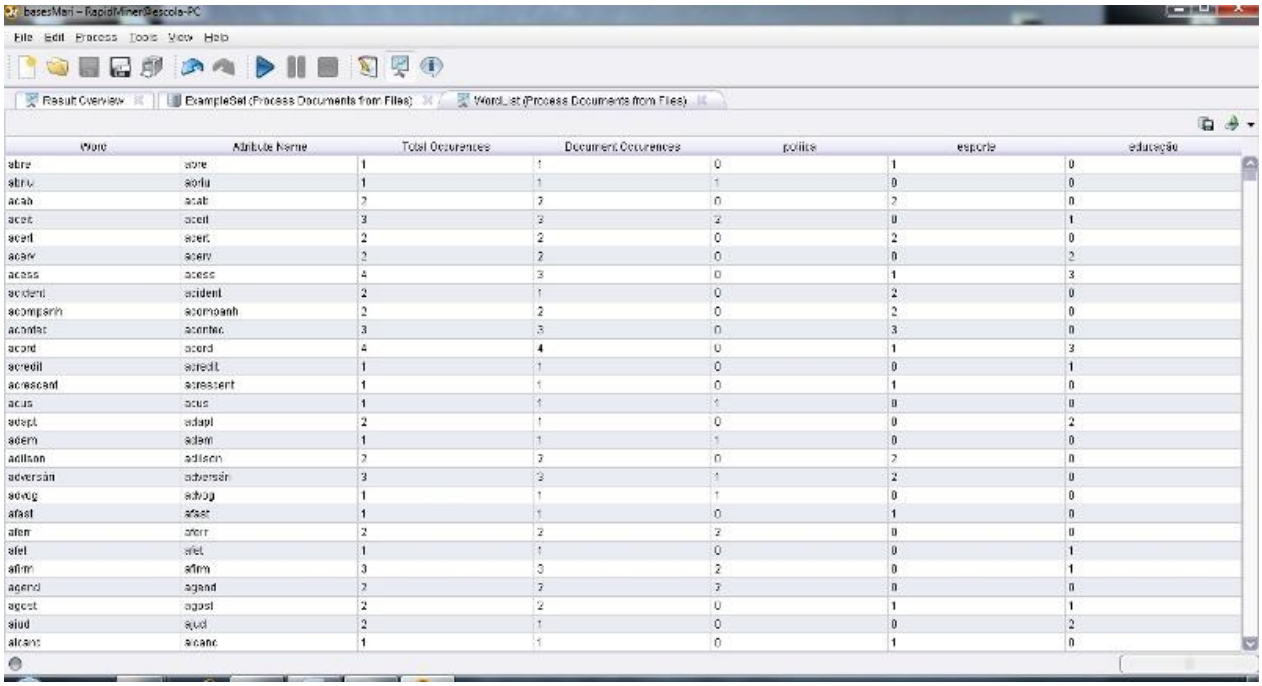

Figura 15 – WordList

4.6 Mineração e Validações

Depois de realizado o pré-processamento, o passo seguinte foi a etapa de Mineração de Dados. A mineração foi realizada também no Rapidminer 5, e consiste em um processo automático de mineração e validação da base. Os tipos de validações utilizadas foram: Validação Bootstrap e Validação Cruzada.

4.6.1 Validação Bootstrap

O processo é montado conforme se pode ver na Figura 16. Como entrada (*Retrieve)* foram usados os dados da ExampleSet gerado pelo pré-processamento da base, que liga-se ao componente de Validação Bootstrap, *Bootstrapping Validation*. Este operador de validação executa várias amostragens bootstrap (amostragem com reposição) sobre o conjunto de entrada e treina um modelo sobre essas amostras. As demais amostras, ou seja, aquelas que não foram incluídas compõem um teste em que o modelo é avaliado.

Neste operador é especificado o número de validações que o algoritmo deve executar, no caso deste trabalho optou-se por cinco validações.

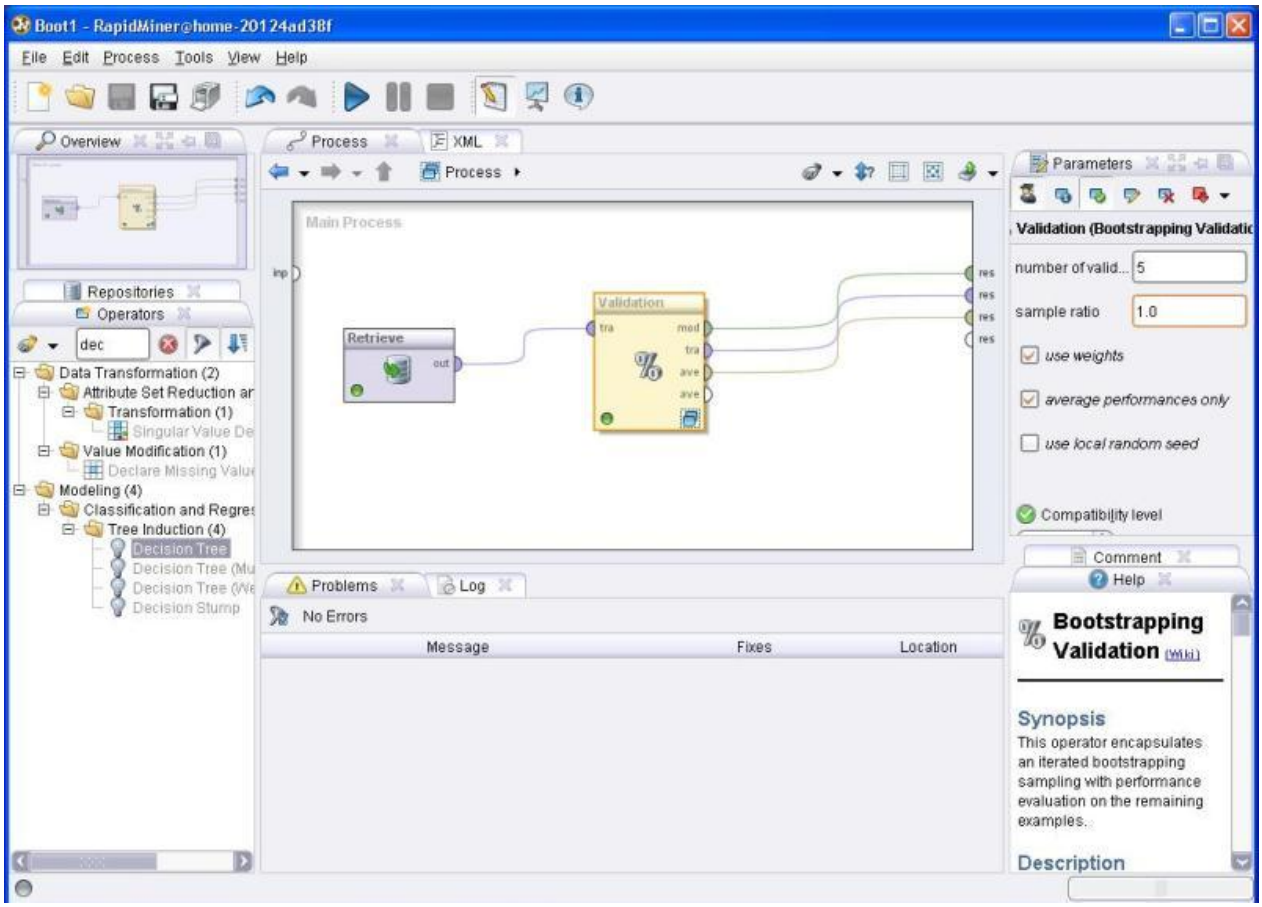

Figura 16 - Processo Bootstrapping

Dentro do componente de validação na parte de Treino (*training)* é colocado o algoritmo - no caso da Figura 17 mostra o *Decision Tree,* mas foram realizados testes também com o *K-NN, SVM e o Naives Bayes* - e no Test (*testing)* é aplicado o modelo com o *Apply Model* ligado ao componente *Performance,* que fornece uma lista de valores de desempenho determinados automaticamente, do algoritmo testado.

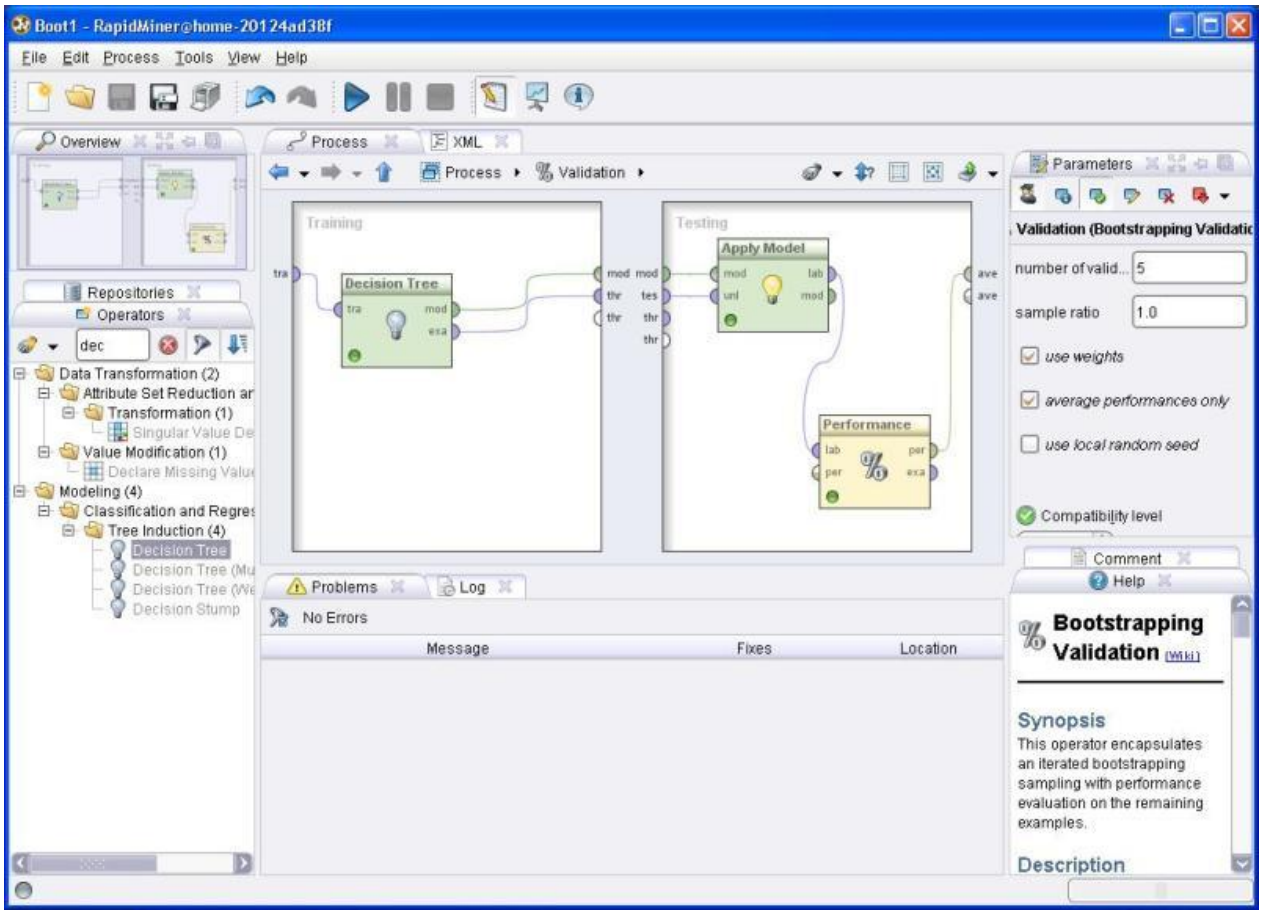

Figura 17 - Processo de validação

4.6.2 Validação Cruzada

O processo de Validação Cruzada foi montado na ferramenta Rapidminer utilizando-se do componente de Validação Cruzada, *X-Validation*, e como atributo deste componente foi especificado o número de validações como sendo 5 vezes, como mostra a Figura 18:

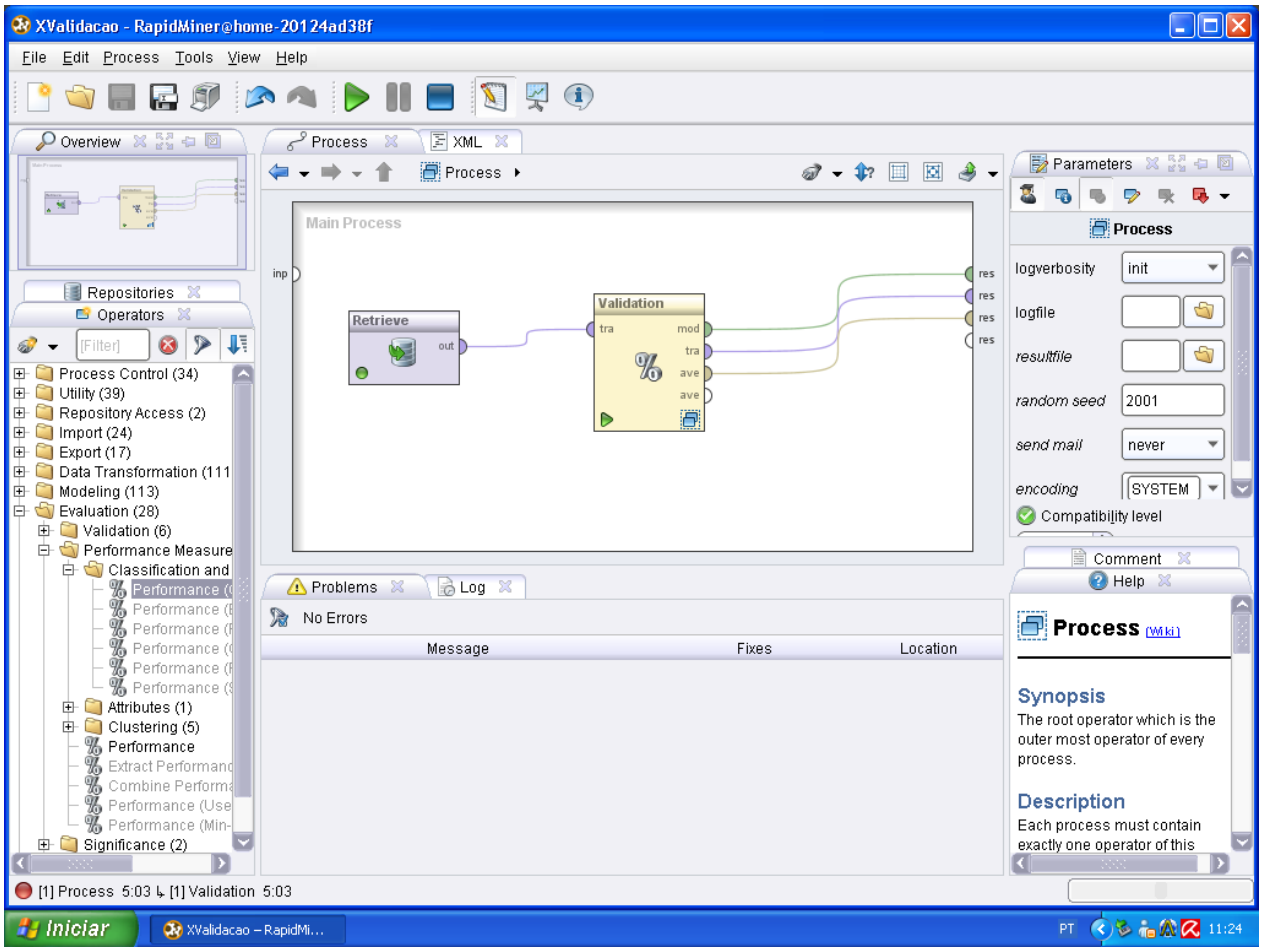

Figura 18 - Validação cruzada

Dentro do componente de validação cruzada é montado um processo interno, que pode ser observado na Figura 19, onde na parte de treino foi especificado o algoritmo. Foram realizados testes com *o Decision Tree, K-NN, SVM e Naives Bayes.* Na seção de testes é aplicado o modelo com o componente *Aplly Model*, e este é ligado ao componente *Performance,* para mostrar os índices de precisão dos algoritmos.

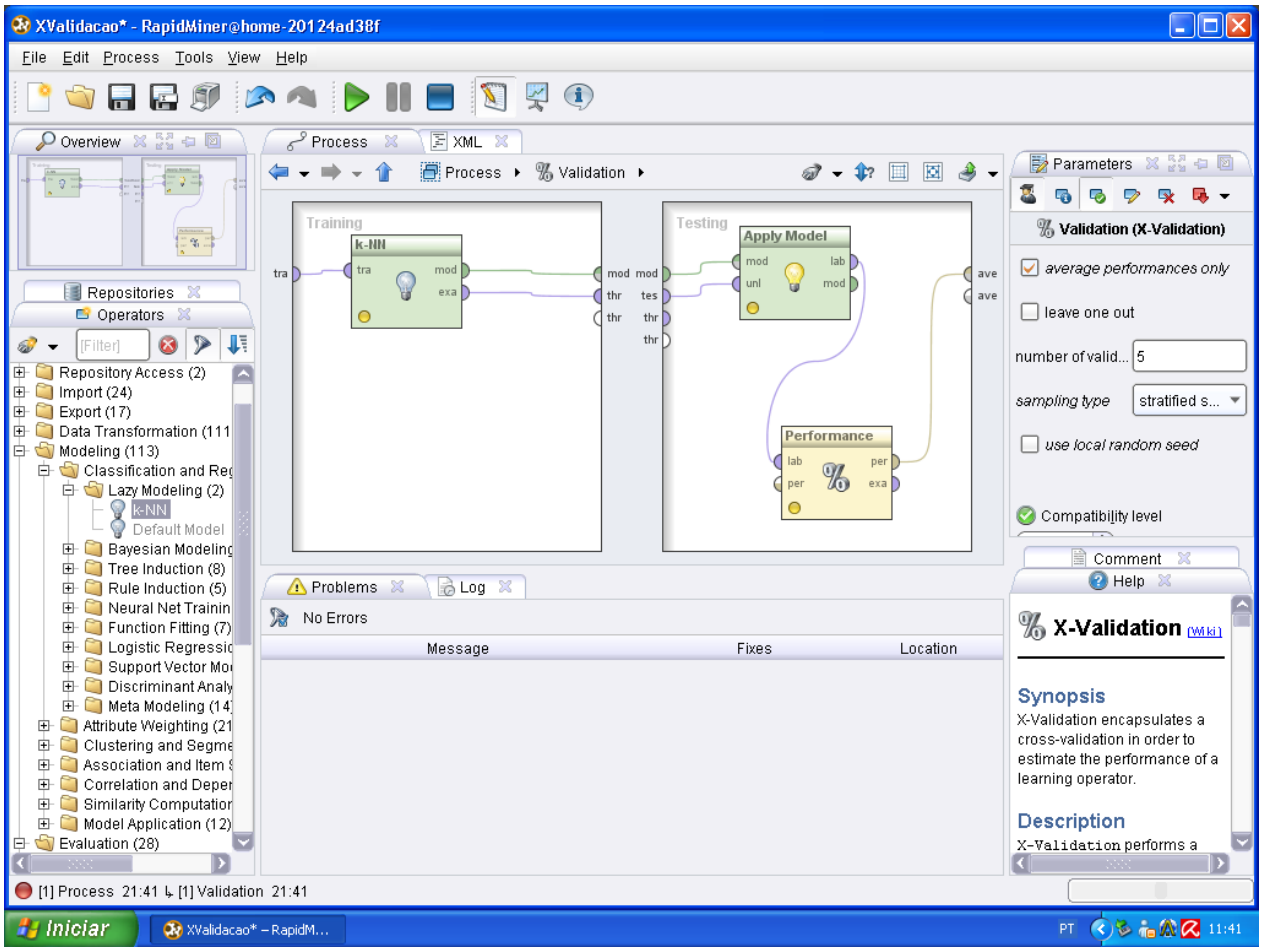

Figura 19 - Processo interno de validação

O processo da Figura 16 e 18 geram as respostas para o operador de validação *Bootstrapping Validation* (figura 16) *e X-Validation* (figura 18), e este gera como saída o desempenho de cada um dos algoritmos, numa tabela com os acertos de cada classe, como mostra a figura abaixo:

Table View Plot View

accuracy: 80.91% +/- 9.49% (mikro: 80.98%)

	true Politica	true Esporte	true Educacao	class precision
pred. Politica	43	1	8	82.69%
pred. Esporte	8	45	0	84.91%
pred. Educacao	12	2	44	75.86%
class recall	68.25%	93.75%	84.62%	

Figura 20 – Exemplo de demepenho do modelo

5 RESULTADOS

Os testes de classificação foram feitos da seguinte forma: A base total é composta por 120 documentos, sendo assim foram separadas aleatoriamente bases menores, de 30, 60, 90 e 120 registros, para a realização dos testes com cada um dos algoritmos estudados.

Foram realizadas testes de Validação Bootstrap e Validação Cruzada, e os resultados dos dois tipos de validação foram parecidos.

A Figura 21 mostra os índices de acertos de cada algoritmo, na Validação Bootstrap, de acordo com os tamanhos de base testados:

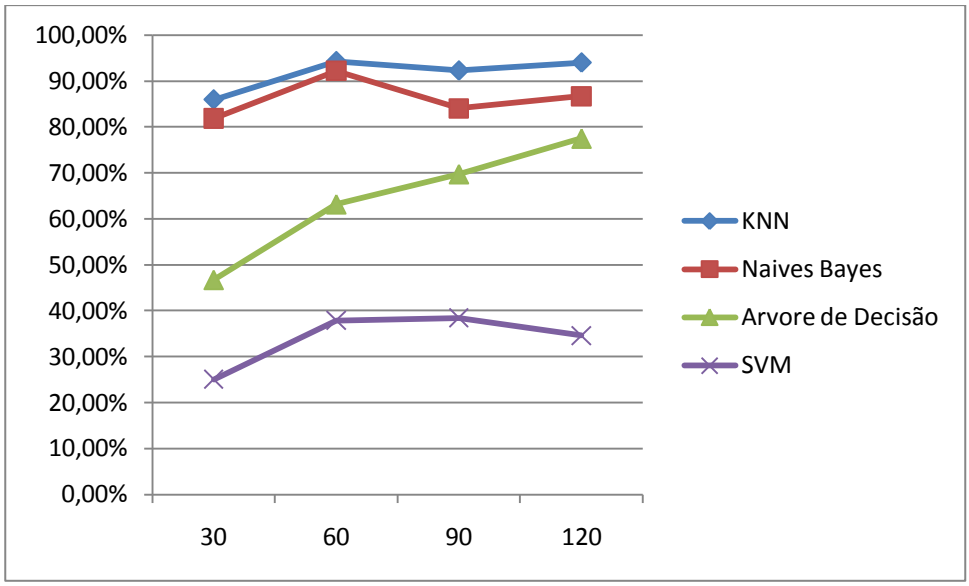

Figura 21 - Índices de precisão dos algoritmos com a validação Bootstrap

O quadro abaixo mostra os índices de acertos dos algoritmos, em cada categoria de notícias, utilizando a Validação Bootstrap:

Quadro 01 – Percentuais de acerto dos algoritmos através da Validação Bootstrap

K-NN	Documentos	Política	Educação	Esportes	**Desempenho Geral**
	30	80%	86,67%	94,12%	**85,96%**
	60	90%	96,43%	97,37%	**94,30%**
	90	94,12%	87,50%	96,23%	**92,26%**
	120	92,41%	92,31%	98,33 %	**94,01%**
Naives Bayes	Documentos	Política	Educação	Esportes	**Desempenho Geral**
	30	93,75%	80%	76,19%	**81,86%**
	60	94,12%	90,91%	92,31%	**92,23%**
	90	83,67%	83,87%	84,21%	**84,06%**
	120	88,73%	83,75%	87,88%	**86,69%**
Decision Tree	Documentos	Política	Educação	Esportes	**Desempenho Geral**
	30	47,09%	39,09%	54,12%	**46,76%**
	60	80,95%	45,40	70,73%	**63,19%**
	90	65,31%	86,54	61,19%	**69,74%**
	120	67,35%	78,57%	97,96%	**77,47%**
SVM	Documentos	Política	Educação	Esportes	**Desempenho Geral**
	30	26,32%	24,24%	5%	**25,07%**
	60	50%	30,19%	46,15%	**38,85%**
	90	49,06%	44%	31,11%	**38,44%**
	120	40%	37,70%	21,95%	**34,62%**

Pode-se observar que para os algoritmos K-NN, SVM quanto para o Naives Bayes, o maior índice de precisão se deu com bases de 60 documentos. Já o decision Tree aumentou a precisão de acordo com o aumento da base.

A Figura 22 mostra os índices de acertos de cada algoritmo, na Validação Cruzada, de acordo com os tamanhos de base testados:

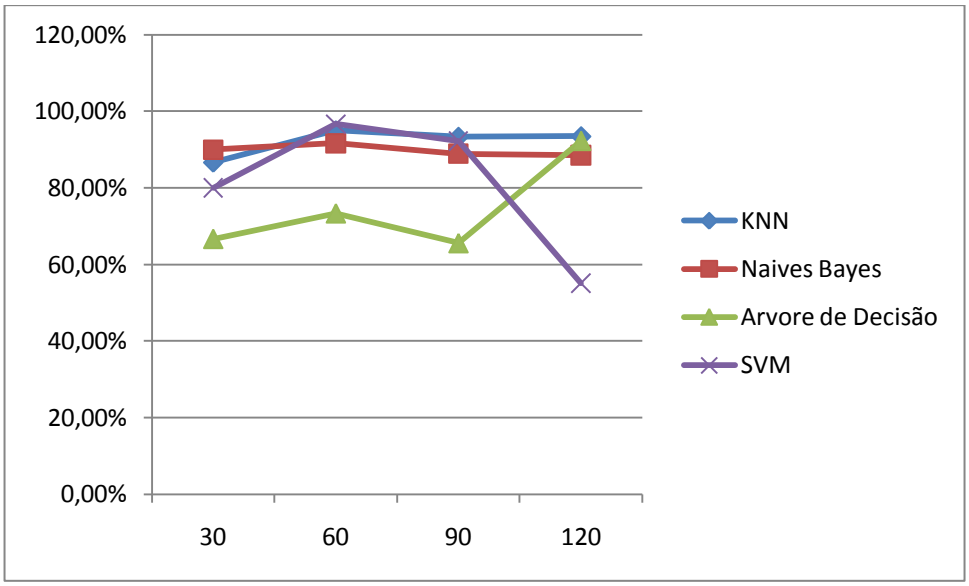

Figura 22 - Índices de precisão dos algoritmos com a validação cruzada

O quadro a seguir mostra os índices de acertos dos algoritmos, em cada categoria de notícias, utilizando a Validação Cruzada. E de acordo com os dados da desta, pode-se observar que nos algoritmos Naives Bayes, K-NN e SVM o maior índice de acertos, foi com a base de 60 registros, da mesma maneira ocorrida com o Bootstrap.

Analisando-se o gráfico da Figura 22, que mostra os resultados da Validação Cruzada, em relação à Figura 2, Validação Bootstrap, nota-se uma grande diferença em relação ao algoritmo SVM. Isto, ocorreu provavelmente pelo fato de na hora do treinamento os documentos escolhidos pelo bootstrap aleatoriamente foram muito diferentes do conjunto usado para teste. E já na validação cruzada, todos os documentos são utilizados tanto para treino como para testes, ocasionando maiores percentuais de acertos do algoritmo.

Quadro 02 – Percentuais de acerto dos algoritmos através da Validação Cruzada

K-NN	Documentos	Política	Educação	Esportes	**Desempenho Geral**
	30	75%	88,89%	100%	**86,67%**
	60	90,48%	95%	100%	**95%**
	90	90,32%	90,62%	100%	**93,33%**
	120	90,24%	95,24%	94,87%	**93,47%**
Naives Bayes	Documentos	Política	Educação	Esportes	**Desempenho Geral**
	30	100%	81,82%	90,91%	**90%**
	60	100%	86,36%	90,48%	**91,67%**
	90	89,29%	82,35%	96,43%	**88,89%**
	120	89,47%	84,44%	92,31%	**88,53%**
Decision Tree	Documentos	Política	Educação	Esportes	**Desempenho Geral**
	30	50%	66,67%	100%	**66,67%**
	60	59,38%	76,92%	100%	**73,33%**
	90	50%	73,33%	95,65%	**65,56%**
	120	54,84%	53,70%	78,38%	**61,57%**
SVM	Documentos	Política	Educação	Esportes	**Desempenho Geral**
	30	100%	62,50%	100%	**80%**
	60	95,24%	100%	95,24%	**96,67%**
	90	96,43%	83,33%	100%	**92,22%**
	120	41,18%	68,29%	84,85%	**55,10%**

Observa-se também que neste tipo de validação, o algoritmo Naives Bayes, com 30 e 60 documentos mostrou um índice de 100% de acerto para as bases de política. Isto pode ter ocorrido pelo fato, de mesmo essas notícias terem sido coletadas

aleatoriamente, sem escolha de tema específico, por ser ano eleitoral, as notícias de política em sua maior parte estão relacionadas às eleições e candidatos de 2010 e isto pode ter influenciado o algoritmo.

O algoritmo de Árvore de Decisão também demonstrou 100% de acertos com base de 30 e 60 registros de Esportes e analisando a WordList gerada a maior parte das notícias que foram utilizadas neste teste se relaciona com futebol, o que pode ter induzido o algoritmo pela similaridade de termos.

Nota-se também, que o algoritmo SVM apresentou uma queda considerável da base de 90 para a base de 120 documentos.

A Figura 23 demonstra o tempo de resposta de cada algoritmo, em minutos:

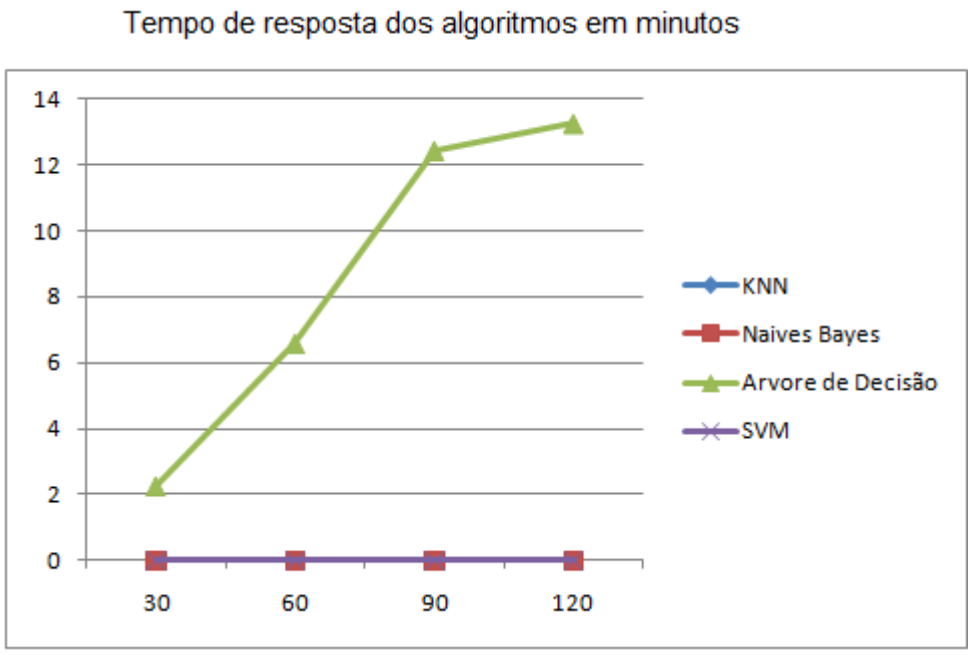

Figura 23 - Tempo de resposta de cada algoritmo

De acordo com a figura 23, que demonstra o tempo de resposta de cada algoritmo para processamento da base pode-se notar que os algoritmos KNN, SVM e Naives Bayes apresentam um processamento e uma resposta muito rápida, menos de um segundo de processamento. Já o algoritmo de Árvores de Decisão, para processar as mesmas bases, mostraram alto tempo de processamento.

6 CONCLUSÕES

Depois de analisados os resultados das Validações, levando-se em conta o percentual da precisão de cada algoritmo e também seu tempo de resposta, pode-se concluir que o algoritmo mais indicado para este tipo de base textual – Notícias – é o algoritmo K-NN, chegando a aproximadamente 95% de acertos, e também um baixo tempo para processamento, conclui-se então que este algoritmo é indicado para este tipo de base textual. Deve-se, contudo, lembrar, que este algoritmo de classificação, utiliza a teoria do vizinho mais próximo, que leva em consideração o mais parecido, ou que tenha as características mais parecidas com o que se deseja classificar, e o fato das Bases Textuais terem sido extraídas de poucos lugares, e ainda os textos de Política tratar em sua grande maioria das Eleições e Candidatos de 2010, pode ter exercido alguma influencia na hora de realizar a classificação.

É importante então ressaltar, que os resultados não foram induzidos, no entanto pelo fato das notícias terem sido extraídos de poucas fontes, e estas serem semelhantes, isto pode ter influenciado de alguma forma nos resultados alcançados, afinal os textos retirados de um mesmo site, e escrito pelo mesmo colunista, utiliza termos parecidos. Sendo assim, para se ter um resultado mais confiante seria necessário que a base fosse formada a partir de fontes o mais distintas.

O algoritmo Naives Bayes apresentou pequena diferença nos índices em relação ao K-NN, pode-se assim concluir que se usado para a Mineração deste tipo de dados não-estruturados apresentará resultados satisfatórios.

O algoritmo de Árvores de Decisão apresentou baixo rendimento e ainda um tempo muito alto para processamento em relação aos outros algoritmos de classificação testados, e um aumento neste tempo de acordo com o aumento no tamanho da base, sendo assim conclui-se que a utilização deste algoritmo é inviável para este tipo específico de base textual.

7 REFERÊNCIAS BIBLIOGRÁFICAS

BARION, E. C. N. *Mineração de Textos: Text Mining.* Revista de Ciências Exatas e Tecnologia. Valhinhos/Sp v. 3, n.3, dez/2008.

BEPPLER, M. et ai (2005). *Apilcação de Text Mining para extração de Conhecimento Jurisprudencial.* Disponível em: http://www.unesc.net/sulcomp/05/art081sulcomp2005.pdf Acesso em: 12/04/2010

CASTRO, S. A.; GONÇALVES, P.R.; CAZARINI, E. W. *O uso de OLAP na estratégia de vendas de uma indústria de calçados alavancando a cadeia de suprimentos.* XXIV Encontro Nacional de Engenharia de Produção, Florianópolis, v1, p.02-04, 2004.

COELHO, F. L. *Classificação semi-automática de monografias.* 2008. 71f. Trabalho de conclusão de curso (Graduação em Ciência da Computação) – Centro Universitário Feevale. Novo Hamburgo. 2008.

GONÇALVES, E. C. (2005) *Regras de Associação e suas medidas de interesse objetivas e subjetivas.* Disponível em: http://www.dcc.ufla.br/infocomp/artigos/v4.1/art04.pdf Acesso em: 21/03/2010

LOH, S; WIVES, L; FRANIER, A. *Recuperação semântica de documentos textuais na internet.* Programa de Pós-Graduação em Computação – Universidade Federal do Rio Grande do Sul, 2004.

LOPES, F.M. *Um modelo perceptivo de Limiarização de imagens digitais.* 2003. 129f. Dissertação(Mestrado em Informática) – Universidade Federal do Paraná. Curitiba, 2003.

MORAIS, E. *Contextualização de Documentos em Domínios Representados por Ontologias Utilizando Mineração de Textos.* 2007. 113f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Goiás. Goiás, 2007.

NAGLIS, D. L. *Mineração de Dados: Uma aplicação na base de dados de artigos de periódicos científicos das áreas de informação (ABCDM).* 2008. 155f. Monografia (Especialização em Engenharia Elétrica) – Faculdade de Tecnologia da Universidade de Brasilia. Brasília, 2008.

OLIVEIRA, A.O.C.; PEREIRA, L.M.R.; SILVA, M.W.S.*Extração de conhecimento a partir de dados estruturados e não-estruturados. Belém/PA. 2001.*

REIS, A.P. *A dinâmica de Aprendizagem em Arranjos Produtivos Locais: Um estudo das Redes de Conhecimeto das Pequenas e Médias Empresas de Software na Construção de suas Capacitações.* 2008. 258f. Tese (Doutorado em Engenharia de Produção) – Universidade de São Paulo, 2008.

ROMÃO, W. et AL (1999). *Extração de Regras de Associação em C&T: O algoritmo Apriori.* Disponível em: http://www.abepro.org.br/biblioteca/ENEGE1999.A0901.pdf Acesso em: 25/03/2010

WIVES, L. *Tecnologia de descoberta de conhecimento em textos aplicada a inteligência competitiva.* Exame de Qualificação EQ-069, PPGC-UFRGS, 2002.