UNIVERSIDADE ESTADUAL DO CEARÁ

AFRANIO MARTINS SOARES

A MINERAÇÃO DE TEXTO NA ANÁLISE DE CONTAS PÚBLICAS MUNICIPAIS

FORTALEZA - CEARÁ
2010

AFRANIO MARTINS SOARES

A MINERAÇÃO DE TEXTO NA ANÁLISE DE CONTAS PÚBLICAS MUNICIPAIS

Dissertação apresentada ao Curso de Mestrado Profissional em Computação Aplicada da Universidade Estadual do Ceará, como requisito parcial para a obtenção do grau de Mestrado em Computação.

Área de Concentação: Sistemas de Apoio à Decisão.

Orientador: Prof. Dr. Jerffeson Teixeira de Souza.

FORTALEZA-CE
2010

S676m Soares, Afranio Martins

A Mineração de Texto na Análise de Contas Públicas / Afranio Martins Soares. — Fortaleza: UECE / MPCOMP, 2010.

72 f.

Orientador: Prof. Dr. Jerffeson Teixeira de Souza.

Dissertação (Mestrado Profissional em Informática Aplicada) – Universidade Estadual do Ceará, 2010

1. Extração do conhecimento em dados textuais. 2. Mineração de Texto. 3. Classificação de Despesas Públicas. I. Universidade Estadual do Ceará – MPCOMP.

CDD: 001.6

AGRADECIMENTOS

Agradeço a DEUS por tudo que tenho na vida.

RESUMO

O crescente armazenamento de informações em meio informatizado, decorrentes das ações e relações humanas, já não é novidade. Porem, a cada dia novos estudos são direcionados à descoberta do conhecimento existente nestas informações. Nesta seara, a extração do conhecimento em bases de dados textuais tem se mostrado um tema de grande relevância, e, para este fim, destaca-se de maneira especial o processo de “Mineração de Texto”.

A presente dissertação se contextualiza nas atividades de auditorias, desenvolvidas pelos técnicos do Tribunal de Contas dos Municípios do Estado do Ceará-TCMCE. O objetivo fim deste trabalho científico é definir os passos de um processo de mineração de texto, destinado a classificar despesas públicas por objeto de gasto, com base na análise automatizada dos campos de históricos de notas de empenhos, presentes nas prestações de contas dos municípios do Estado do Ceará.

A fundamentação teórica aborda, inicialmente, aspectos legais que incidem sobre classificações de despesas na contabilidade pública nacional. Nas seções seguintes são contemplados temas relacionados ao processo de Mineração de Texto, tais como: Mineração de Dados, Processamento de Linguagem Natural, Etapas do processo de Mineração de Textos, Métodos de Avaliação e Modelo de Projetos para Mineração de Dados.

A solução desenvolvida para a classificação de documentos de despesas seguiu os passos de um projeto de mineração de texto. Utilizando o modelo de projetos CRISP-DM, foram definidas as etapas de um processo de mineração de textos, objetivando classificar documentos de despesas públicas, através da análise do conteúdo textual dos históricos de notas de empenhos. Dentro dos temas descritos, vale destacar “o Entendimento e Preparação dos Dados e a Construção de Modelos de Classificadores”, contemplando, ainda, uma avaliação dos resultados obtidos. As ferramentas utilizadas no processo foram o SGBD Postgresql e o programa WEKA.

Demonstra-se, contudo, que a aplicação de técnicas de mineração de texto é uma solução possível e viável, para a extração de conhecimentos e classificação de documentos de despesas públicas.

Palavras-chave: Auditorias. Mineração. Texto. Classificação.

ABSTRACT

The growing stockpile of information in computerized environment arising from human actions and relations, is not new. However, every day new studies are directed to the discovery of existing knowledge on this information. This field, the extraction of knowledge in textual databases has been a topic of great relevance, and to this end, there is a special way the process of "Text Mining".

This work is contextualized in the activities of audits conducted by technicians from the Municipal Court of the State of Ceará TCMCE. The end goal of this scientific work boils down to defining the steps of a process of text mining, to classify public expenditures by object of expenditure, based on automated analysis of historical notes fields of endeavor, the benefits of these accounts municipalities of Ceará.

The theoretical approach, initially, legal aspects that affect rankings of national public expenditure in the accounts. In the following sections are covered issues related to the process of text mining, such as Data Mining, Natural Language Processing, Process Steps Text Mining, Methods and Evaluation Model for Data Mining Projects.

The solution developed for the classification of the expenditure documents followed in the footsteps of a mining project text. Using the model projects CRISP-DM, defined the steps in a process of text mining, focusing on classifying documents in government expenditure, by analyzing the textual content of historical notes endeavors. Among the topics described, it is worth "Understanding and Data Preparation and Construction of Models Classifiers", comprising also an evaluation of the results. The tools used in the process were the PostgreSQL DBMS and the WEKA program.

Demonstrates, however, that the application of text mining techniques is a feasible and viable solution for the extraction of knowledge and classification of documents for public expenditure.

Keyword: Audits, Mining, Text, Classification.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE GRÁFICOS

LISTA DE SIGLAS E ABREVIATURAS

AUC	“Area Under Curve” (Área abaixo da curva)
CRISP-DM	Cross-Industry Standard Process for Data Mining (Padrão de Processo Industrial para Mineração de Dados)
GNU	General Public License (Licença Pública Geral utilizada em software livre)
PLN	Processamento de Linguagem Natural
RI	Recuperação de Informações
ROC	Receiver Operating Characteristics (Características Operacionais do Receptor)
SGBD	Sistema de Gereciamento de Banco de Dados
SIM	Sistema de Informações Municipais
SOF	Secretaria de Orçamento Federal
STN	Secretaria do Tesouro Nacional
TCMCE	Tribunal de Contas dos Municípios do Estado do Ceará

SUMÁRIO

1. Introdução

1.1. Problematização

Este trabalho científico se insere no contexto das atividades desenvolvidas no Tribunal de Contas dos Municípios do Estado do Ceará – TCMCE, órgão responsável por exercer a tarefa de controle externo junto às administrações municipais do Estado do Ceará.

Dentro das atividades desenvolvidas pelo TCMCE, merecem especial atenção as auditorias contábeis, baseadas na análise técnica dos documentos de receitas e despesas constantes nas prestações de contas municipais.

As prestações de contas municipais são enviadas ao TCMCE através do Sistema de Informações Municipais – SIM. O SIM é um projeto que vem sendo desenvolvido desde 1996, por uma equipe coordenada pelo autor desta dissertação, que definiu os padrões de envio dessas prestações de contas municipais ao TCMCE em meio informatizado. O SIM foi plenamente implantado em todas as administrações municipais do Estado do Ceará desde 2002, tornando-se a ferramenta de suporte para a recepção e armazenamento das referidas contas municipais nos bancos de dados do TCMCE, além de provê os meios para as análises técnicas de auditorias.

Dentro da documentação contábil das unidades gestoras municipais, especificamente na documentação comprobatória de despesas, destaca-se um documento específico denominado de “nota de empenho”. A “nota de empenho” tem a função de formalizar os compromissos de compra de serviços ou mercadorias por parte das administrações municipais em favor de terceiros.

Para se ampliar a noção de importância da “Nota de Empenho” na contabilidade pública, nos casos em que o instrumento de contrato é facultativo, a Lei nº. 8.666/1993 admite a possibilidade de substituí-lo pela “Nota de Empenho da Despesa”, hipótese em que esta representa o próprio contrato.

A nota de empenho indica o nome do credor, a representação e a importância da despesa, além de conter uma codificação conhecida como dotação, que tem a função de informar sobre vários aspectos inerentes à finalidade da despesa. Na nota de empenho, a representação da despesa se faz com o

preenchimento de um campo texto conhecido como “histórico da nota de empenho”, onde se encontra a descrição detalhada do objeto (objeto de gasto) que está sendo adquirido. A extração de informações do conteúdo deste campo é o foco principal da presente dissertação.

Durante os trabalhos de auditoria contábil no TCMCE, normalmente se monta uma estratégia com o objetivo de se direcionar as atenções, com base nos objetos de gastos que originaram as despesas mais relevantes (serviços e mercadorias) e, consequentemente, levando à identificação dos fornecedores que mais se destacam na administração municipal fiscalizada. Atualmente a identificação do objeto de gasto ainda depende, em grande parte, da leitura do “histórico das notas de empenhos”, mesmo tratando-se de dados disponíveis em meio informatizado.

A identificação de objetos de gasto através da leitura tem dificultado consideravelmente os trabalhos de auditoria do TCMCE, considerando o grande volume de documentos de despesas apresentados anualmente nas prestações de contas de seus jurisdicionados. Anualmente o TCMCE fiscaliza as contas de aproximadamente 1.900 unidades gestoras municipais, distribuídas nos 184 municípios do Estado do Ceará.

Além do grande volume de informações a ser analisado nas auditorias realizadas pelo TCMCE, existe outro agravante relacionado com as normas brasileiras, que definem as classificações das despesas das administrações públicas. Através dos códigos das dotações, as referidas normas agrupam em um mesmo “Elemento de Gasto” uma diversidade muito grande de objetos. Para exemplificar tal afirmação, a lista no próximo parágrafo apresenta a relação de itens classificados como “Elemento de Despesa com código 30”, que identifica somente despesas consideradas como aquisição de “Material de Consumo”, conforme texto retirado da Portaria Interministerial 163, STN e SOF, de 04/05/2001:

“Elemento de Despesa 30 - Material de Consumo: Despesas com álcool automotivo; gasolina automotiva; diesel automotivo; lubrificantes automotivos; combustível e lubrificantes de aviação; gás engarrafado; outros combustíveis e lubrificantes; material biológico, farmacológico e laboratorial; animais para estudo, corte ou abate; alimentos para animais; material de coudelaria ou de uso zootécnico;

sementes e mudas de plantas; gêneros de alimentação; material de construção para reparos em imóveis; material de manobra e patrulhamento; material de proteção, segurança, socorro e sobrevivência; material de expediente; material de cama e mesa, copa e cozinha, e produtos de higienização; material gráfico e de processamento de dados; aquisição de disquete; material para esportes e diversões; material para fotografia e filmagem; material para instalação elétrica e eletrônica; material para manutenção, reposição e aplicação; material odontológico, hospitalar e ambulatorial; material químico; material para telecomunicações; vestuário, uniformes, fardamento, tecidos e aviamentos; material de acondicionamento e embalagem; suprimento de proteção ao vôo; suprimento de aviação; sobressalentes de máquinas e motores de navios e esquadra; explosivos e munições; bandeiras, flâmulas e insígnias e outros materiais de uso não-duradouro."

A solução do problema, portanto, requer a elaboração de um meio eficiente para classificar os documentos de despesas, presente nas prestações de contas municipais, com base no conteúdo dos históricos das notas de empenhos.

1.2. Justificativa

O problema apresentado na seção anterior requer uma solução que apresenta algumas dificuldades para sua solução, considerando-se apenas as disponibilidades técnicas convencionais. Quais sejam:

- Extrair conhecimentos, através da análise do conteúdo textual de documentos de despesas em uma base de dados informatizada;
- Descobrir padrões para associar documentos de texto a classes temáticas pré-definidas (objetos de gastos);
- Com base na descoberta de padrões, no tocante ao objeto de gasto de que trata o conteúdo de cada documento, deverá ser feita uma classificação que possibilitará um agrupamento de documentos de despesas e consequentemente de fornecedores.

Os métodos convencionais de consulta a bases de dados não apresentam soluções para este tipo de problema, devido ao nível de complexidade.

Documentos textuais se enquadram em uma forma de representação de informação totalmente diferenciada daquela encontrada nas bases de dados estruturadas, onde as informações são disponibilizadas em forma de matriz, em que as linhas representam instâncias e as colunas atributos.

Com a indisponibilidade de ferramentas específicas para a solução do tipo do problema em questão, foi necessária a elaboração de uma dissertação, baseada na pesquisa de publicações de trabalhos técnicos e projetos executados, que mencionam alguns dos meios utilizados para a descoberta de padrões em documentos textuais, assim como, a classificação automática de documentos através de classes temáticas conhecidas.

De acordo com as publicações pesquisadas, para a solução do tipo de problema abordado são utilizadas técnicas de Mineração de Texto, que envolvem entre outros assuntos: o Processamento de Linguagem Natural, Aprendizado de Máquina, Mineração de Dados, Estatística e Consultas em Bancos de Dados Textuais.

1.3. Objetivos

1.3.1. Objetivo Geral

Classificar despesas públicas através da análise do conteúdo dos campos de históricos das notas de empenhos, utilizando técnicas de mineração de texto.

1.3.2. Objetivos Específicos

Utilizar ferramentas informatizadas para classificar despesas públicas, através da análise do conteúdo dos campos de históricos de notas de empenhos;

Extrair conhecimentos dos campos de históricos das notas de empenhos utilizando técnicas de mineração de texto;

Classificar despesas públicas dentro de uma lista de objetos de gasto pré-definidos, utilizando ferramentas de mineração de texto para analisar os campos de históricos das notas de empenhos;

Avaliar a eficiência de ferramentas de mineração de texto no processo de classificação de despesas, com base na análise automatizada dos campos de históricos de notas de empenhos.

2. Fundamentação Teórica

Este capítulo menciona resumidamente a fundamentação teórica de que trata as normas de contabilidade pública brasileira, confirmando a problematização exposta na seção 1.1, assim como as técnicas relacionadas com o processo de mineração de texto, utilizadas como meio para a solução apresentada na presente dissertação.

2.1. Classificação da Despesa Pública

O objetivo desta seção é comprovar a inexistência de um instrumento legal, dentro das normas atuais da contabilidade pública brasileira, que defina uma classificação para as despesas públicas, a ponto de permitir a identificação de objetos de gastos, sem que seja necessária a leitura dos “históricos das notas de empenhos”. Portanto, na seqüência serão expostas e comentadas apenas as normas federais que tratam da classificação da despesa:

Termos da Portaria Interministerial Nº. 163, de 04 de maio de 2001, da Secretaria do Tesouro Nacional do Ministério da Fazenda e da Secretaria do Orçamento Federal do Ministério de Planejamento, Orçamento e Gestão:

a) 1º. Item da Portaria Interministerial Nº. 163, no Art. 3º: a classificação da despesa, segundo a sua natureza, compõe-se de:

I - categoria econômica;

II - grupo de natureza da despesa;

III - elemento de despesa;

Comentário: A norma acima apresenta classificações para a despesa somente até o nível de elemento de despesa.

b) 2º. Item da Portaria Interministerial Nº. 163, no Art. 3º, parágrafo 3º: “O elemento de despesa tem por finalidade identificar os objetos de gastos, tais como vencimentos e vantagens fixas, juros, diárias, material de consumo, serviços de terceiros prestados sob qualquer forma, subvenções sociais, obras e instalações, equipamentos e material permanente, auxílios, amortização e outros de que a administração pública se serve para a consecução de seus fins”.

Comentário: O dispositivo acima atribui ao Elemento de Despesa a função

de identificar o objeto de gasto da administração pública. Porém, trata de forma homogênea itens abrangentes, como por exemplo: “Material de Consumo”.

c) 3º. Item da Portaria Interministerial Nº. 163, no Art. 3º, parágrafo 5º: “É facultado o desdobramento suplementar dos elementos de despesa para atendimento das necessidades de escrituração contábil e controle da execução orçamentária”.

Comentário: Este dispositivo inclui o “desdobramento suplementar dos elementos de despesa”, demonstrando a necessidade de criar sub-classificações nas despesas (sub-elementos), o que poderia facilitar a identificação dos objetos de gastos de uma forma mais analítica. Porém, este nível de controle ficou facultado, o que fragilizou sua adoção pelas administrações municipais e inviabilizou sua utilização para efeito de fiscalização.

Termos da Portaria Nº. 448, de 13 de setembro de 2002, da Secretaria do Tesouro Nacional do Ministério da Fazenda:

a) 1º. Item da Portaria Nº. 448, no Artigo 1º: “Divulgar o detalhamento das naturezas de despesa, 339030 - Material de Consumo, 339036 - Outros Serviços de Terceiros Pessoa Física, 339039 - Outros Serviços de Terceiros Pessoa Jurídica e 449052 - Equipamentos e Material Permanente, de acordo com os anexos I, II, III, IV, respectivamente, para fins de utilização pela União, Estados, DF e Municípios, com o objetivo de auxiliar, em nível de execução, o processo de apropriação contábil da despesa que menciona”.

Comentário: Esta portaria demonstra claramente a preocupação em definir os sub-grupos das quatro classificações de despesas mais abrangentes, quanto ao objeto de gasto que identificam. Para se ter uma ideia, o supra citado “anexo I” identifica 52(cinqüenta e dois) sub-grupos de objetos para a classificação 339030 - Material de Consumo, porém, não apresenta um código padrão para cada sub-grupo, reforçando o caráter facultativo do desdobramento suplementar dos elementos de despesas, conforme citado anteriormente.

Termos da Portaria Conjunta Nº. 3, da Secretaria do Tesouro Nacional do Ministério da Fazenda e da Secretaria do Orçamento Federal do Ministério de Planejamento, Orçamento e Gestão:

a) 1º. Item da Portaria Conjunta Nº. 3, no Art. 4º: “As alterações da classificação da receita e das despesas orçamentárias, constantes dos Manuais de que trata o art. 1º desta Portaria, observarão o disposto no caput do art. 2º da Portaria Interministerial STN/SOF nº. 163, de 4 de maio de 2001”.

“O código da natureza de despesa orçamentária é composto por seis dígitos, desdobrado até o nível de elemento ou, opcionalmente, por oito, contemplando o desdobramento facultativo do elemento”, conforme figura 2.1:

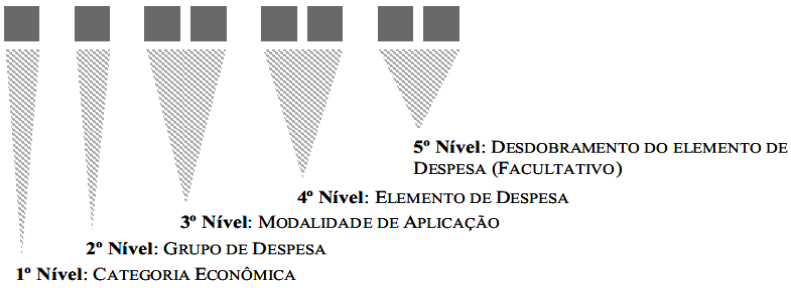


Figura 2.1 – Estrutura do código da Natureza de Despesa orçamentária

b) 2º. Item da Portaria Conjunta no. 3, no Art. 4º, Item 4.4.1.4 ELEMENTO DE DESPESA: “tem por finalidade identificar os objetos de gasto, tais como vencimentos e vantagens fixas, juros, diárias, material de consumo, serviços de terceiros prestados sob qualquer forma, subvenções sociais, obras e instalações, equipamentos e material permanente, auxílios, amortização e outros que a administração pública utiliza para a consecução de seus fins, conforme códigos definidos neste Manual.”

c) 3º. Item da Portaria Conjunta no. 3, no Art. 4º, Item 4.4.1.5, Desdobramento Facultativo do Elemento da Despesa: “Conforme as necessidades de escrituração contábil e controle da execução orçamentária, fica facultado por parte de cada ente o desdobramento dos elementos de despesa”.

Comentário: Esta Portaria Conjunta Nº. 3 foi lançada no objetivo de consolidar as normas pré-definidas para as receitas e despesas públicas através de manuais, padronizando procedimentos para as esferas federais, estaduais e municipais. Ressalte-se, porém, que foram mantidas as determinações das citadas portarias 163 e 448, no tocante a classificação da despesa em nível de objeto de gasto (elemento de despesa).

2.2. A Descoberta do Conhecimento em Bases de Dados Estruturadas

Esta seção explicita aspectos básicos do processo de extração do conhecimento, quando se utiliza bases de dados estruturadas.

Mesmo quando se trata de bases de dados estruturadas, ainda não se percebe na literatura um consenso na definição, nem na estruturação do processo de extração de conhecimento. Entretanto, vários autores sugerem etapas que devem compor o ciclo do referido processo. Resumindo as abordagens de Fayyad et al (1997), Berry e Linoff (1997), Trybula (1997), Han e Kamber (2000) e Kantardzic (2003), observam-se variações na ordem de apresentação das etapas que poderiam ser resumidas no seguinte (SCHIESSL, 2007):

- Compreensão e definição do problema;
- Seleção de fontes de dados;
- Processo de limpeza e adequação dos dados;
- Análise exploratória;
- Redução de variáveis;
- Relacionamento de objetivos;
- Mineração de dados;
- Avaliação ou Interpretação de resultados;
- Transformação do conhecimento adquirido em ação.

A figura 2.2 apresenta um esquema gráfico das etapas descritas acima:

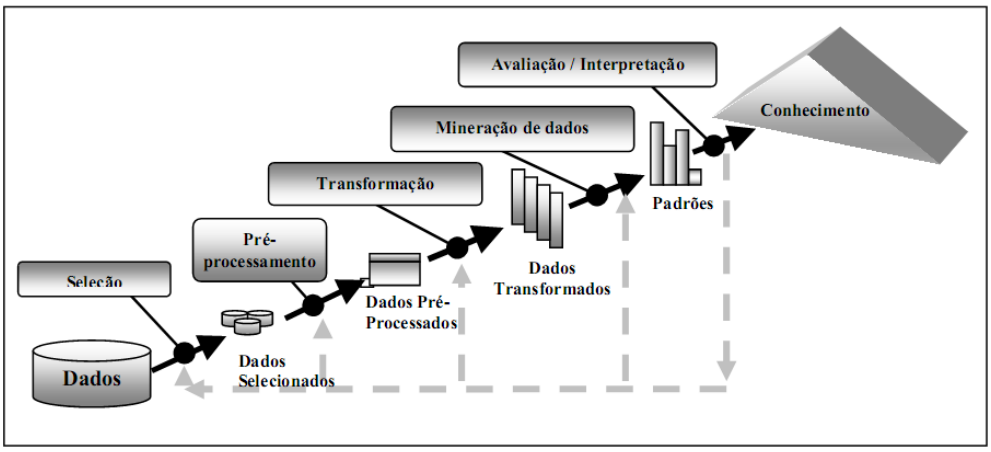


Figura 2.2 : Etapas da Mineração de Dados (Fayyad et al, 1997)

2.3. O Processamento de Linguagem Natural – PLN

A Mineração de texto refere-se à extração de conhecimentos provenientes de textos livres ou não estruturados, contemplando tópicos relacionados com recuperação de informações, destinadas à classificação ou clusterização de textos. PLN é uma tentativa de extrair a representação do significado completo de um texto livre. Isso diz respeito a informações tais como: quem fez o que, com quem, quando, onde, como e por que. PLN normalmente utiliza conceitos linguísticos, classes gramaticais, estrutura gramatical, além de lidar com anáforas e ambiguidades. Neste domínio contemplam-se várias representações de conhecimento, relacionadas com o léxico da palavra, seus significados, suas propriedades gramaticais, regras gramaticais e outros recurso, tais como: ontologia de entidades e ações, ou dicionários de sinônimos e abreviações. (KAO, A.; POTEET, S., 2007)

O Processamento de Linguagem Natural é o conjunto de métodos formais utilizados para analisar textos e gerar informações compatíveis com a escrita em linguagem humana natural. Normalmente computadores estão aptos a compreender instruções escritas em linguagens de computação, mas possuem muita dificuldade em entender comandos escritos em uma linguagem humana. Isso se deve ao fato das linguagens de computação serem extremamente precisas, contendo regras fixas e estruturas lógicas bem definidas, que permitem ao computador saber exatamente como deve proceder a cada comando. Já em um idioma humano uma simples frase normalmente contém ambiguidades, nuances e interpretações que dependem do contexto, do conhecimento do mundo, de regras gramaticais, culturais e de conceitos abstratos (JACKSON, PETER; MOULINIER, ISABELLE, 2007).

Na Mineração de Textos, as técnicas de Processamento de Linguagem Natural são largamente utilizadas, a partir da etapa de pré-processamento, objetivando melhor representar o texto e otimizar ao máximo a utilização de seu conteúdo. São consideradas técnicas de PLN na etapa do pré-processamento: Tokenização, Identificação de Palavras, Identificação de Caracteres Especiais, Normalização, etc.

Vale ressaltar, porem, que na mineração de texto as técnicas de PLN não se aplicam somente nas etapas de pré-processamento, mas, de forma relevante nos próprios procedimentos de extração do conhecimento. A seguir estão descritas

algumas técnicas de PLN utilizadas no processo de mineração de texto (HILLOL; JIAWEY; PHILIP; VIPIN, 2008):

- Extração de Informação (Information Extraction);
- Rastreamento de Tópicos (Topic Tracking);
- Sumarização (Summarization);
- Categorização (Categorization);
- Clusterização (Clustering);
- Correlação de Conceitos (Concept linkage);
- Visualização de informação (Information Visualization);
- Resposta à Pergunta (Question answering);

2.4. O Conhecimento em Bases de Dados Textuais

Este tópico explicita aspectos básicos do processo de extração do conhecimento, quando se utiliza bases de dados textuais, sugerindo a ocorrência de iterações na utilização de técnicas que tratam de dados estruturados, o Processamento de Linguagem Natural e o tratamento de dados textuais (SCHIESSL, 2007).

É comum na bibliografia técnica a afirmação de que as bases de dados textuais apresentam-se de forma não-estruturada. No entanto, possuem uma estrutura implícita que necessita de técnicas especializadas para ser reconhecida por sistemas automatizados. O processamento de linguagem natural (PLN) trata exatamente da descoberta dessas estruturas implícitas, como por exemplo: estrutura sintática (disposição das palavras na frase e a das frases no discurso), estrutura morfológica (estrutura e formação de palavras), estrutura semântica (significação das palavras no contexto). (RAJMAN; BESANÇON, 1997).

A integração dos processos definidos para a extração do conhecimento em base de dados estruturada com as técnicas de PLN resulta, então, na descoberta do conhecimento em bases textuais.

Portanto, o processo de Extração do Conhecimento em Base de Dados Textuais pode ser considerado um conjunto de etapas, que se inicia pelo pré-processamento de informações em formato textual utilizando técnicas de PLN, com o objetivo de transformar tais informações em uma base de dados apta ao

processamento aplicado em dados estruturados.

A figura 2.3 é uma adaptação do modelo de Fayyad et al (1997), onde se consolidam as concepções de autores como Wives e Loh (1999), Dörre et al (1999) e Tan (1999). Técnica de Processamento de Linguagem Natural - PLN foram incluídas na etapa de pré-processamento do processo de mineração de dados:

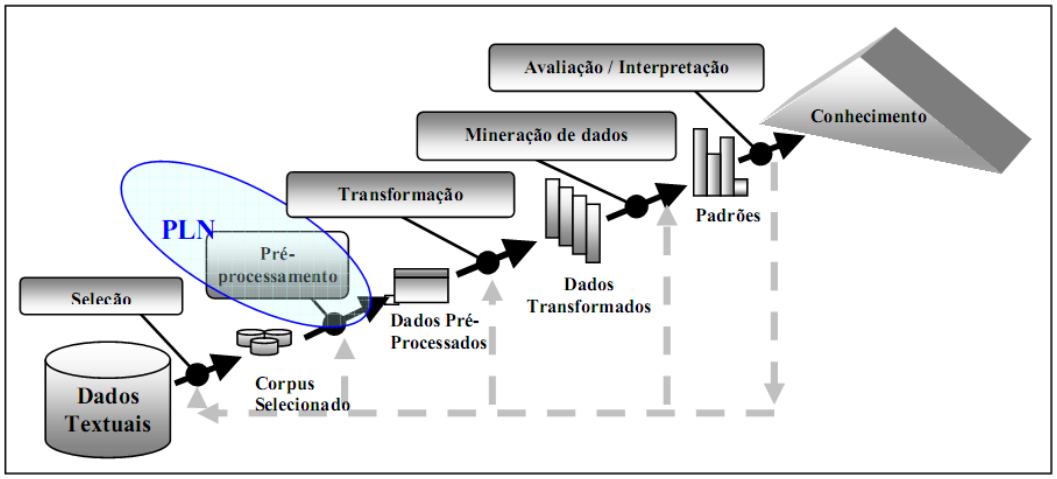


Figura 2.3: Adaptação do modelo de Fayyad para Mineração de Texto.

No escopo da Categorização de Textos existem três aplicações mais comuns: Indexação de Texto, Classificação de Documentos e Filtragem de Texto.

Através de um Sistema de Recuperação de Informações – RI, cada documento de uma coleção é identificado por um ou mais termos-chave que representam seu conteúdo. Com estas informações, o sistema está pronto para recuperar os documentos que atendem aos requisitos das consultas dos usuários, baseadas nos termos utilizados. Os termos-chave passam a pertencer a um vocabulário controlado pertencente ao domínio em que se insere a aplicação. A tarefa de atribuir as palavras de um vocabulário controlado a documentos textuais é chamada “Indexação de texto”.

A “Classificação de Documentos” consiste em rotular cada documento de uma coleção com base em classes temáticas pré-definidas. Por exemplo: as notícias de um jornal podem ser classificadas em Política, Classificados e Esporte.

Um problema de classificação de documentos apresenta várias particularidades, quais sejam: quais documentos pertencem a qual classe, quais critérios devem ser utilizados para identificar cada classe. O processo de

classificação ocorre com a apresentação dos documentos um a um, nunca através de processamento em lote. (FELDMAN e SANGER, 2007)

A Filtragem de Textos pode ser considerada como uma tarefa de classificação, onde existem apenas duas classes temáticas pré-definidas. Um exemplo comum é a checagem de um E-mail para verificar se é ou não “spam”. Um sistema de filtragem pode ser ajustado com base nos resultados obtidos pelo usuário, subtraindo-se iterativamente do sistema as causas dos erros ocorridos, através de técnicas de aprendizado adaptativo. (WEISS, INDURKHYA, ZHAN , DAMERAU, 2005).

Para a maioria dos sistemas de categorização, a Recuperação de Erros (documento que deveriam ter sido classificados em uma classe específica e não foi) e a Precisão de Erros (documentos que não deveriam ter sido classificados em uma classe específica e não foi), são analisados com foco no custo da ocorrência do erro. (FELDMAN e SANGER, 2007).

2.5. Etapas Básicas no Processo de Mineração de Texto

Esta seção contempla as etapas de mineração de texto necessárias para compor um processo de extração automática de conhecimentos úteis, provenientes dos campos de históricos de notas de empenhos que compõem as prestações de contas públicas.

2.5.1. A Coleta de Dados:

A coleta de dados é a primeira tarefa no processo de Mineração de Texto, e tem a função de formar a base de dados textuais a ser utilizada em todo o processo (SCHIESSL, 2007).

Na etapa inicial da coleta de dados é necessário que se defina claramente que tipo de conteúdo deve ter um conjunto de documentos, para que este se torne relevante ao domínio em que será trabalhado. Outro ponto importante é compatibilizar a forma em que o documento se apresenta em sua fonte com aquela exigida no processo de Mineração de Texto.

2.5.2. A identificação da Palavra:

A definição da unidade básica do texto, denominada de “palavra” ou “termo”, foi uma preocupação percebida por Gean, C. C. e Kaestner, C. A. A.(2004). O texto puro, em seu estado primário e livre de qualquer formatação, é tratado na etapa do pré-processamento, no objetivo de produzir uma representação adequada ao objeto a que se destina.

A identificação de palavras através da análise das seqüências de caracteres no texto foi a abordagem utilizada por Salton (1997). Salton também aconselhou a utilização de dicionários, no objetivo de fornecer uma fonte de comparação com as palavras encontradas no texto, a fim de validar ou possibilitar a correção das expressões que realmente existem, e eliminar as seqüências de caracteres insignificantes. Este método mostra-se muito útil quando os documentos disponíveis apresentam muitos erros gramaticais ou caracteres inválidos (ARANHA 2007).

A utilização de dicionários também facilita a identificação de termos considerados relevantes, quando se deseja utilizar palavras pré-definidas no índice, evitando-se a identificação de palavras desconhecidas. Isto é muito útil quando se trabalha em domínios específicos. Neste contexto também é válida a utilização de um analisador léxico que identifique seqüências de caracteres e forme palavras.

2.5.3. Etapas do Pré-processamento de Dados Textuais:

A etapa inicial da Mineração de Texto é o Pré-processamento, que consiste em uma série de adaptações e transformações executadas em uma coleção de textos, objetivando estruturá-la em uma representação na forma de atributos e valores. De um modo geral, a etapa de pré-processamento tem a finalidade de aprimorar a qualidade da representação dos dados a serem utilizados e organizá-los. As etapas do pré-processamento visam ajustar os dados textuais, a ponto de poderem ser submetidos a algum algoritmo de indexação ou de mineração de dados.

Na prática, o pré-processamento normalmente significa dividir o texto em palavras (tokenização), reduzir palavras à suas raízes (stemming), remover palavras irrelevantes (stopwords) e classificá-las segundo a classe gramatical.

O pré-processamento transforma os textos em uma representação

estruturada, para que os dados possam ser submetidos ao processo como um todo. Porem, durante a transformação dos textos em um formato estruturado, existe a possibilidade de perdas de informações relevantes. Um desafio, nesse caso, é obter uma boa representação, minimizando prejuízos nas informações.

Compreendem etapas típicas do pré-processamento de dados textuais:

a) Tokenização: Identifica e separa as expressões do texto em palavras e é definida pelo reconhecimento de expressões entre marcas de pontuação, tais como: espaços em branco, vírgulas, pontos, etc.. Por padrão cada “token” fica separado através de aspas. A “tokenização” prepara e salva os dados em um repositório para utilização nos processos subsequentes. Exemplo: Ao tokenizar a expressão “a casa é de papel” obtem-se - “a” “casa” “é” “de” “papel”.

b) Limpeza de Caracteres Especiais: é a etapa em que são descartados os caracteres especiais, que não contribuem para a extração do conhecimento. Por exemplo: `, ~, !, @, #, $, %, ^, &, *, (,), +, |, \, /, {, }, [,], :, ;, ?, ', =, ¨, -, §, °, ª, £, ¢, <, >, ¬, }, {, etc.

c) Remoção de “Stopwords”: remove a palavras comuns que não apresentam uma semântica significativa no documento em que se encontram. Normalmente as palavras comuns são constituídas de artigos, preposições, verbos auxiliares, etc, tais como: “que”, “de/do/das”, “o” ou “a”. Após a eliminação das “stopwords” obtém-se uma representação reduzida do texto, mas, ainda em um formato natural. Exemplo: Ao remover “stopwords” da expressão - “a casa é de papel”, obtem-se - “casa” “é” “papel”.

d) Normalização de Variações Morfológicas ou Obtenção de Radicais (stems): esta etapa faz com que expressões textuais afins sejam representadas por um elemento único, portanto com uma semântica única. Normalmente isso acontece com palavras de classe aberta (substantivos, adjetivos, pronomes, verbos e advérbios). Este passo permite uma redução significativa no número de elementos que compõem o texto (ARANHA 2007). Exemplo: As expressões – “terem”, terão”, “teria”, “ter”, “terá” ao serem normalizadas resultam na expressão “ter”.

2.6. Otimizações e Adaptações nos Dados

Os dados textuais normalmente precisam passar por um estágio de transformação, para que possam ser reconhecidos pelos algoritmos disponíveis ao processo de mineração. Isto é, as coleções de textos precisam ser representadas normalmente através de dados numéricos e de forma otimizada, sem que haja prejuízo na qualidade da informação original.

2.6.1. A Representação Quantitativa de Dados Textuais:

Este tópico visa demonstrar algumas formas de representar dados textuais em um formato numérico (SCHIESSL, 2007).

a) Representação de Dados

Uma base de dados estruturada é apresentada normalmente em um formato matricial. Cada linha representa um registro, ou uma instância, ou um caso específico, e cada coluna representa um atributo, ou uma das qualidades que se aplica à informação:

Quadro 2.1: Representação de Dados

NOME	***CPF***	***RG***	***IDADE***
João	212121212	1234	40
Maria	323232323	4321	30
...	...	...	...

No Quadro 2.1 cada linha, ou instância, registra um indivíduo juntamente com as suas informações pessoais e características que o identificam. Cada coluna trata de um tipo de informação, ou atributo específico. Um registro é representado por todos os campos da mesma linha. Isto é, um indivíduo é representado por todos os atributos da linha que o identifica. A extração do conhecimento normalmente é realizada com os dados apresentados desta forma.

b) Representação de um conjunto de Dados Textuais:

Suponhamos que temos a seguinte coleção de documentos:

- Doc.01: A casa é do homem, o cão não é;
- Doc.02: O cão correu para casa;
- Doc.03: O cão é do homem;

O Quadro 2.2 representa o conteúdo dos documentos através de uma notação binária. O “S” identificando a existência do termo no documento, e o “N” sinalizando que o documento não contem aquele termo. De uma forma semelhante à tabela anterior, cada documento é identificado por um registro inteiro, isto é, por todos os atributos de uma mesma linha.

Quadro 2.2: Representação Binária de um Conjunto de Dados Textuais - 1

DOCUMENTO	**TERMOS**									
	A	**Casa**	**é**	**do**	**homem**	**o**	**cão**	**não**	**correu**	**para**
Doc.01	S	S	S	S	S	S	S	S	N	N
Doc.02	N	S	N	N	N	S	S	N	S	S
Doc.03	N	N	S	S	S	S	S	N	N	N

c) Representação binária de um conjunto de Dados Textuais:

O Quadro 2.3 representa uma coleção de documentos através de uma notação binária, onde o “1” identifica a existência do termo no documento, e o “0” sinaliza que o documento não contem aquele termo ou atributo. Se $a_{12} = 1$, então o Atributo “2” ocorre no Documento “1”, Se $a_{21} = 0$, então o Atributo “1” não ocorre no Documento “2”. Cada documento é identificado por um registro inteiro, isto é, por todos os atributos de uma mesma linha.

Quadro 2.3: Representação Binária de um Conjunto de Dados Textuais - 2

DOCUMENTO	***TERMOS***									
	A	**Casa**	**é**	**do**	**homem**	**o**	**cão**	**não**	**correu**	**para**
Doc.01	1	1	1	1	1	1	1	1	0	0
Doc.02	0	1	0	0	0	1	1	0	1	1
Doc.03	0	0	1	1	1	1	1	0	0	0

d) Representação do Conjunto de Dados Textuais por Frequência:

Esta representação baseia-se em síntese na quantidade de vezes que um mesmo termo ou atributo ocorre em um documento. O Quadro 2.4 exemplifica esta abordagem:

Quadro 2.4: Representação do Conjunto de Dados Textuais por Frequência.

DOCUMENTO	***TERMOS***									
	A	**Casa**	**é**	**do**	**homem**	**o**	**cão**	**não**	**correu**	**para**
Doc.01	1	1	2	1	1	1	1	1	0	0
Doc.02	0	1	0	0	0	1	1	0	1	1
Doc.03	0	0	1	1	1	1	1	0	0	0

2.6.2. A Indexação de Dados Textuais:

O processo de indexação visa otimizar o tempo de busca das informações em bases de dados. (Manuais Online do SQL Server 2008, 2009), (POSTGRESQL, 2009).

A indexação do tipo “Full-text” (texto completo) é uma abordagem recentemente adotada em pacotes de bancos de dados e destina-se ao processo de recuperação de informações. O nome comumente dado ao índice de bases de dados contendo textos em linguagem natural é Full-Text Index (índice de texto completo). Muitos sistemas gerenciadores de bancos de dados como Oracle, SQL Server, MySQL e Postgresql já incluem essas funcionalidades em suas versões. No Postgresql o referido índice é utilizado para realizar tarefas de Full Text Search (pesquisa em texto inteiro).

O objetivo fim do processo de pesquisa *Full-text* é obter previamente informações relevantes de uma coleção de dados, para utilizá-las em resposta à necessidade do usuário. Essa necessidade é normalmente expressa por uma consulta que pretende verificar cada registro do banco em busca de uma expressão requisitada. Pela metodologia convencional, o mesmo processo consistiria em abrir cada registro do banco à procura de uma palavra-chave, usando um algoritmo de pesquisa por string (sequência de caracteres). Porem, abrir cada documento em tempo de processamento pode ser custoso, se o volume de documentos for muito grande.

A ideia, portanto, é deixar os dados já organizados de forma a otimizar o tempo de resposta às consultas. Isso é feito extraindo-se as informações das palavras em cada documento e armazenando-as de uma forma vinculada a um processo de indexação, otimizando, sobremaneira, o acesso aos dados durante uma pesquisa. Quando a consulta é feita, só é necessário comparar os dados da consulta com os documentos, com o auxílio de índices, e escolher os documentos considerados relevantes.

2.7. Técnicas Utilizadas na Extração do Conhecimento

Esta seção aborda o aprendizado necessário aos sistemas para a extração do conhecimento em bases de dados textuais, que constitui uma das principais fases da mineração de texto.

2.7.1. Algoritmo de Aprendizado

Os anos 90 trouxeram um novo paradigma que é o da automação na obtenção do conhecimento. A partir de um aprendizado obtido através do reconhecimento de padrões no conteúdo de grandes bases de dados, os sistemas também deverão ter a habilidade de prever, ou predizer, informações úteis provenientes de novas bases de dados. Isto é, os sistemas passam a identificar padrões de forma automática, nos dados disponíveis e ter a capacidade de reconhecer esses padrões em outras bases de dados. São vários os algoritmos utilizados para o aprendizado automático, assim como variam também suas formas de concepção e finalidades.

Preliminarmente, podemos dividir os tipos de algoritmos de aprendizado automático existentes em dois grupos, para os fins desta dissertação, são eles: Algoritmos de Aprendizado Supervisionado e Não Supervisionado.

Os Algoritmos de “Aprendizado Supervisionado” caracterizam-se pela disponibilidade de um conjunto de treinamento, composto por dados de entrada previamente classificados. Ou seja, os dados disponíveis já contemplam a entrada e sua respectiva informação de retorno. Os algoritmos de “Aprendizado Não Supervisionado” sugerem que o sistema aprenda sem que haja um conjunto do tipo “entrada x retorno desejado", considerando para o aprendizado apenas as características e padrões existentes nos dados disponíveis para o processo de extração de conhecimento (MITCHEL, 1997), (ARANHA 2007).

Os problemas de Classificação e Regressão de dados são exemplos que utilizam o Aprendizado Supervisionado. Quanto aos problemas que são resolvidos através de “Aprendizado Não Supervisionado”, podemos citar: Problemas de Clusterização; Problemas de Estimação de Densidade (ZADROZNY, 2009).

Os tópicos a seguir apresentam algumas das técnicas de aprendizado automático, com base no tipo de problema a ser resolvido, são:

- Problema de Classificação: árvores de decisão, algoritmos bayesianos, redes neurais, SVM (Suport Vector Machine);
- Problema de Regressão: regressão linear, árvore de decisão;
- Problema de Clusterização: k-means, EM;
- Problema de Classificação Sensível a Custos: MetaCost, costing;
- Problema de Aprendizado por Reforço: Q-learning, aprendizado por reforço baseado em classificação;
- Problema de Aprendizado Relacional: Redes neurais.

2.7.2. Modelos para Visualização de Dados

Alguns modelos de visualização de dados utilizados na prática, para tomada de decisão, utilizam recursos gráficos para a representação das informações. Isto é, fazem uso da visualização espacial para efeito de apresentação dos dados.

A seguir estão discriminados alguns modelos de visualização dos dados, e comentários sobre as técnicas utilizadas para a obtenção do aprendizado automático (ARANHA 2007):

a) Espaço-Vetorial:

O “Espaço Vetorial” é um dos modelos mais usados em mineração de textos, sendo a classificação automática de documentos a aplicação mais comum. Para esse tipo de aplicação, é necessária uma descrição adequada da semântica do texto.

Através do Espaço Vetorial, os sistemas de recuperação de informações podem aplicar técnicas de redução de dimensionalidade, baseados em métodos matemáticos, objetivando aumentar a eficiência do processamento de dados.

A classificação de documentos pode ser definida, sobre o espaço vetorial, como um caso especial de um problema de classificação supervisionada no contexto do “Reconhecimento de Padrões”. (BERRY, Michael W, 2004)

b) Análise de Correspondência:

Baseia-se na análise de associações entre palavras. O resultado da análise permite que um ser humano interprete visualmente as associações, enxergue conglomerados e assim extraia um conhecimento relevante.

c) Análise de Discriminante:

Este método procura achar as palavras que mais discriminam o conjunto de documentos baseado nos conceitos Bayesianos (com base estatística). A diferença para outras abordagens estatísticas é que essa explicita o conhecimento extraído e determina as palavras e os pesos relativos. (ARANHA 2007):

2.7.3. Modelos de Aprendizado Utilizados nos Algoritmos

Esta seção aborda de forma resumida sobre o modelo de aprendizado de máquina utilizando Árvore de Decisão (IAN H, 2005), (MITCHEL, 1997).

2.7.3.1. Árvore de Decisão

Os algoritmos ID3, C4.5 e J48 (implementação do C4.5 release 8), pertencem a uma família de algoritmos de árvore de decisão, dos mais usados, que utilizam inferências indutivas. Baseiam-se em funções de aproximação que utilizam valores discretos, com robustez suficiente para tratarem de dados com ruídos, e capazes de aprendizado com expressões disjuntivas. Essas árvores procuram um espaço de hipóteses mais expressivo, evitando as dificuldades decorrentes de restrições.

Árvore de decisão classifica as instâncias de cima para baixo. A ideia é seguir a rota de uma folha que vai classificar uma instância. Cada nó representa um atributo e cada galho descendente da árvore representa um possível valor de atributo (exemplo da figura: Sol, Nublado...). O processo inicia-se classificando o primeiro nodo da árvore, testa o atributo especificado para este nodo, depois move-se para baixo através do galho correspondente ao valor do atributo do exemplo. O processo é repetido nas sub-árvores descendentes.

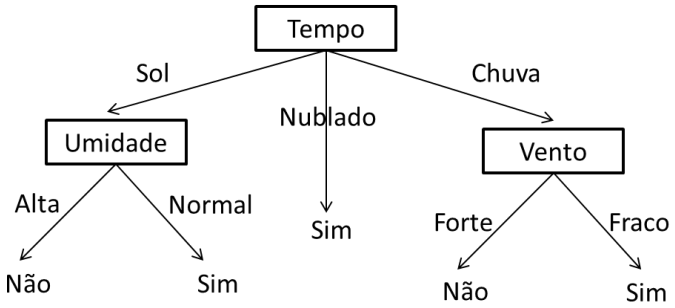


Figura 2.4 – Representação de uma Árvore de Decisão (adaptado de Mitchel, Tom M.)

Em geral uma árvore de decisão representa uma disjunção de conjunções de restrições de valores de atributos de instâncias. Cada caminho ou rota em uma árvore até a folha corresponde a uma conjunção de testes de atributos, e a própria árvore representa a disjunção dessas conjunções. Por exemplo, a árvore acima representa a seguinte expressão:

$$(\ Tempo = Sol \cap Umidade = Normal\ )$$
$$\cup\ (\ Tempo = Nublado\ )$$
$$\cup\ (\ Tempo = Chuva \cap Vento = Fraco\ )$$

Iniciamos o algoritmo de Árvore de Decisão com a seguinte questão: Que atributo deve ser testado na rota inicial da árvore? Para responder, cada instância de atributo é avaliada usando um teste estatístico para avaliar quão bem ela classifica o exemplo de treino. Uma propriedade estatística, conhecida como “Information Gain”, mede a qualidade em que um atributo tem seus exemplos de treino separados de acordo com suas classificações previamente conhecidas, resultando na escolha de uma melhor forma de prosseguir no crescimento da árvore de decisão.

Um outro conceito a ser considerado é a “Entropia”, que mede o grau de impureza das classificações de uma coleção de exemplos e também faz parte do processo de cálculo do “Information Gain”.

Considerando a “Entropia” como a medida de impureza de uma coleção de exemplos de treino, é possível definir a medida de efetividade de um atributo em classificar dados de treino através do “Information gain”, que corresponde à medida da redução esperada de “entropia” causada pelo particionamento de exemplos de acordo com os atributos.

Durante a utilização de uma árvore de decisão para a solução de um problema, é importante a verificação do nível de precisão nos resultados obtidos a cada etapa do crescimento da árvore. Esta medida de precisão tem uma relação direta com um resultado denominado de “Overfitting”.

Dado um espaço de hipótese H, a hipótese h ϵ H é chamada de “overfit” dos dados de treino, se lá existir alguma hipótese alternativa h' ϵ H, tal que, h tem menos erros que h' sobre os exemplos de treino, mas h' tem menos erros que h sobre todas as instâncias em geral.

A figura 2.5 demonstra que a precisão cresce para os dados de treino, porém a partir do 25º nodo da árvore, aproximadamente, os testes realizados com os dados de teste apresentam uma linha decrescente na precisão quando se considera os dados gerais, demonstrando a ocorrência do “overfit”.

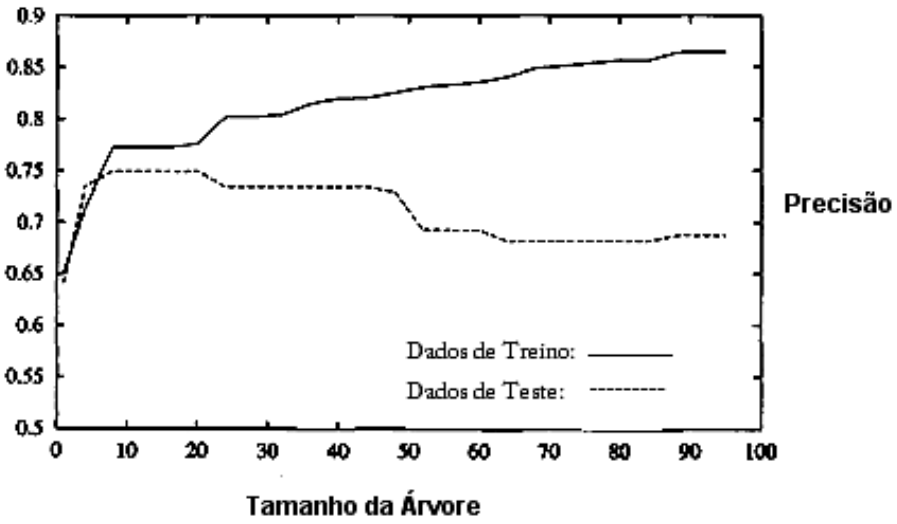


Figura 2.5: Representação gráfica do “Overfitting”

A ocorrência do “overfit” leva à necessidade de se buscar formas de evitá-lo. Existem várias formas de se evitar “overfit” em aprendizado através de árvores de decisão, senão vejamos:

- Parar o crescimento da árvore, antes que ela atinja o ponto máximo em que os dados de treino são classificados com perfeição;
- Deixar a árvore atingir o “overfit” depois podar a árvore.

Observação: Embora a primeira abordagem pareça mais vantajosa, a segunda tem apresentado melhores resultados. (IAN H, 2005)

Este método de aprendizagem, que é considerado supervisionado, está contido entre os mais populares algoritmos de inferência indutiva, e tem sido aplicado com sucesso a uma ampla gama de tarefas de aprendizagem, tais como: Classificação e Regressão.

2.7.4. Métodos para Avaliação de Resultados

A avaliação é a única forma de se conhecer a eficiência de um sistema que se propõe a apresentar resultados, com base em um modelo de aprendizado. O processo de avaliação se torna mais fácil quando estamos lidando com um algoritmo de aprendizado supervisionado, onde se pode comparar, por exemplo, uma classificação de dados apresentada por um sistema com uma classificação previamente considerada correta para os mesmos dados. Os modelos de avaliação de resultados variam de simples comparações de valores, que resultam em percentuais de erros sugeridos em um processo de classificação, a processos mais elaborados, onde se indica o percentual de certeza através de ferramentas estatísticas, contemplando-se "intervalos de confiança", com base em médias, variância e, se for o caso, aplicando distribuições estatísticas normal, qui-quadrada, etc... (IAN H, 2005).

No processo de seleção de um método de avaliação devem ser considerados pontos de extrema relevância, tais como: a quantidade de dados disponíveis para se elaborar o processo de avaliação; o custo de um método contemplar com habilidade o erro de classificação em um domínio específico (ex: um erro em um jogo de azar não pode ter o mesmo custo de um erro de diagnóstico médico).

Com base nos objetivos desta dissertação, serão abordados dois métodos de avaliação, que objetivam demonstrar taxas de erros esperados em um processo de classificação de instâncias: Cross Validation e Matriz de Confusão – Curva ROC.

2.7.4.1. Cross Validation (Validação Cruzada)

Para implementar a avaliação o método “hold-out”, conceitualmente, reserva-se parte dos dados disponíveis da coleção para “treino”, parte para “teste” e parte para “validação”. Mas, na prática o método utiliza 1/3 dos dados disponíveis

para teste e o restante para treino.

Durante a separação dos “dados de treinamento” e “dados de teste” devemos ter o cuidado de mantermos a representatividade. Por exemplo: Se tivermos uma coleção onde 30% dos dados indica o branco e 70% indica o preto, devemos separar os dados de treinamento e os dados de teste com as mesmas proporções. Este procedimento é chamado de “stratified holdout” (Hold-out Estratificação).

Uma forma geral de reduzir algum vício, ou erro, proveniente da manipulação dos dados decorrente dos procedimentos de “hold-out”, é repetir todo o processo várias vezes com diferentes sub-exemplos do que temos disponível. Assim, 2/3 dos dados é randomicamente selecionado para treino, possivelmente com estratificação, e o restante será utilizado para teste. As taxas de erros das diferentes iterações farão uma média aritmética para produzirem um erro sobre o total.

Na técnica de “Cross Validation”, uma variante do “hold-out”, decide-se sobre um número fixo de partição de dados. Por exemplo, divide-se os dados em 3(três) partições iguais. Utiliza-se 2/3 dos dados para treino e 1/3 para dados de teste, de forma que cada uma das três partições será utilizada uma vez como teste. Esta técnica é chamada de “threefold cross-validation” (validação cruzada com três partições). Caso seja utilizada a estratificação, esta técnica será chamada de “stratified threefold cross-validation” (validação cruzada, estratificada em três partições).

Uma prática comum para predizer a taxa de erro é utilizar 10(dez) partições em vez de 3(três). Os dados são divididos aleatoriamente em 10(dez) partições, usando a estratificação, de forma que cada décimo obtido será utilizado uma vez para teste e o restante para treino. Em cada uma das dez experiências será gerada uma taxa de erro. Serão obtidas 10(dez) taxas de erros no final. Então, será calculada a média aritmética dos 10(dez) erros encontrados que representará o erro estimado total. Tecnicamente, avalia-se que em situações onde a disponibilidade de dados é limitada, o padrão é utilizar o Stratified 10-Fold Cross-Validation ”(validação cruzada, estratificada em dez partições).

2.7.4.2. Curva ROC (Matriz de confusão)

ROC - Receiver Operating Characteristics (Características Operacionais do Receptor) é uma técnica para visualizar, avaliar, organizar e selecionar classificadores baseado em suas performances. Para realizar estas análises, gráficos ROC podem mostrar o limiar entre taxas de acertos e taxas de erros dos classificadores.

Considerando-se um conjunto de amostras, onde uma instância pode assumir valores no conjunto “p”, “n”, “positive” e “negative” respectivamente. Tendo-se um classificador e uma instância, pode-se obter 4 situações. Se a instância é “positive” e é classificada como “positive”, conta-se como “true positive”; se é classificada como “negative”, conta-se como “false negative”. Se a instância é “negative” e é classificada como tal, conta-se como “true negative”; se é classificada como “positive”, conta-se como “false positive”. Portanto, através de um classificador e um conjunto de instâncias pode-se construir uma “matriz de confusão” de 2 por 2, no caso de 2 classes. Esta matriz serve como base para muitas métricas que podem ser aplicadas à classificação. A Figura 2.6 mostra a matriz de confusão.

		Classe Verdadeira	
		p	**n**
Classificada Como	**p**	*True Positive*	*False Positive*
	n	*False Negative*	*True Negative*
	Totais	**P**	**N**

Figura: 2.6: Matriz de Confusão gerada a partir de um conjunto de instâncias.

O conjunto de equações demonstra as métricas que podem ser usadas a partir da matriz de confusão. Destacam-se as métricas “tp rate” (taxa de verdadeiro positivo) e “fp rate” (taxa de falso positivo) que servirão como base para a construção do espaço ROC (área limite entre a curva ROC e o eixo das abcissas). A métrica “precision” (precisão) é a taxa de acerto do usuário, isto é, quantas instâncias de uma determinada classe o classificador acertou. A métrica “accuracy” (acurácia) representa a taxa de acerto de todo o classificador, isto é, a razão entre a soma dos acertos das duas classes e o número total de instâncias.

- *tp rate (taxa de verdadeiro positivo) = recall (retorno) = TP/P;*
- *fp rate (taxa de falso positivo) = FP/N*
- *precision (precisão) = TP/(TP+FP)*
- *Fmeasure (Medida F) = 2(precision+1)/recall*

Antes de definir como são construídos os gráficos ROC, tem-se que definir o espaço em que estas curvas serão representadas. Os gráficos ROC são bidimensionais, onde no eixo Y plota-se o valor de “tp rate”(taxa de verdadeiro positivo) e no eixo X o valor de “fp rate” ”(taxa de falso positivo). A Figura 2.7 mostra um gráfico ROC simples, somente com classificadores discretos. Classificadores discretos são aqueles que geram como saída somente uma classe. Estes classificadores fornecem um par (“fp rate”, “tp rate”) correspondendo as coordenadas de um ponto no espaço ROC. Muitos pontos são importantes no espaço ROC. O ponto inferior esquerdo (0,0) representa uma estratégia que nunca gera uma classificação positiva; como um classificador que não comete erros “falso positivos”, mas também não classifica nenhum “verdadeiro positivo”. A estratégia oposta, de incondicionalmente gerar “verdadeiro positivo” é representada pelo ponto superior direito (1, 1).

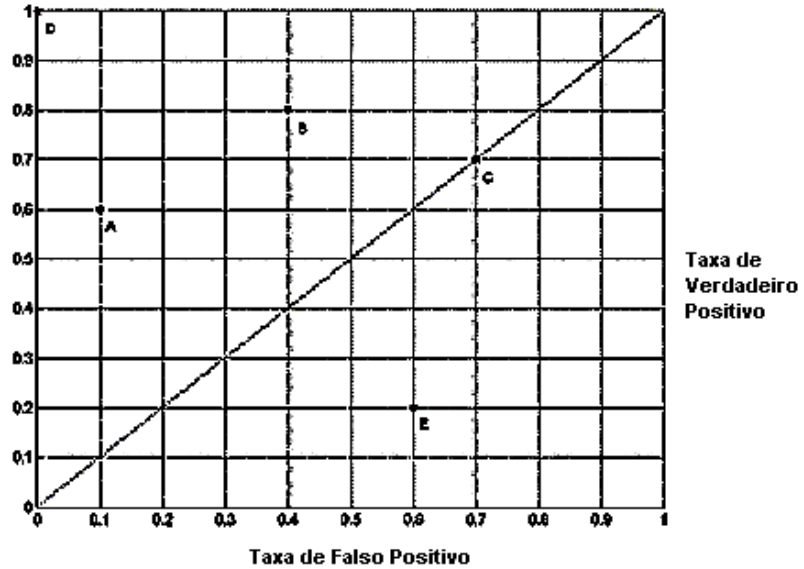


Figura 2.7: Gráfico ROC mostrando 5 classificadores discretos.

Alguns classificadores, tais como “Naive Bayes” ou “Redes Neurais”, naturalmente produzem uma probabilidade ou valor para cada instância que representa o grau de representatividade da classe sobre aquela instância. A partir destes classificadores, podem-se gerar classificadores discretos com uma simples aplicação de um limiar.

Cada limiar produzirá um conjunto distinto de pontos no espaço ROC.

A figura 2.8 mostra um exemplo de uma curva ROC produzida a partir de 20 instâncias, utilizando a limiarização do conjunto de teste:

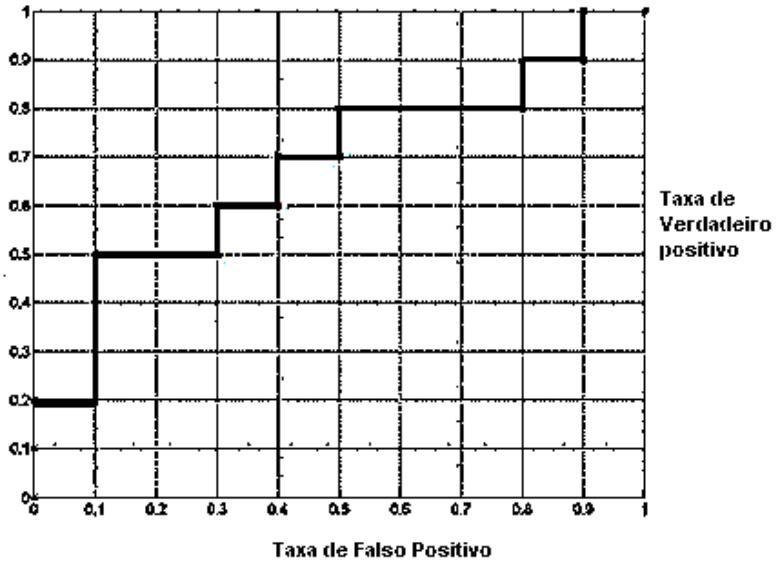


Figura 2.8: Função Degrau

Qualquer curva ROC, que é gerada por um número finito de instância, produz uma “função degrau” no espaço ROC. Quanto maior o número de instâncias, mais contínua a curva fica. É importante ressaltar que cada ponto criado no tempo “t” do processo de geração da curva ROC, depende do ponto “t – 1”.

Uma curva ROC é uma demonstração bidimensional da performance de um classificador. Para comparar classificadores é preciso reduzir a curva ROC a um valor escalar. Um método comum para realizar esta redução é calcular a área abaixo da curva ROC (AUC – “Area Under Curve” - Área abaixo da curva). Como a AUC é uma porção da área do quadrado unitário (espaço ROC), seus valores vão de 0.0 à 1.0. Entretanto, como classificadores piores que os aleatórios não são encontrados no espaço ROC, não existem classificadores com AUC menor que 0.5 (0.5 é a área de um classificador aleatório).

A Figura 2.9 mostra a área abaixo de duas curvas ROC “A” e “B”. O classificador “B” possui uma área maior e, portanto, tem uma melhor performance média. É possível que em algumas regiões do espaço ROC um classificador seja melhor que outro. Na Figura 2.9 o classificador “B” é geralmente melhor que “A” exceto em “fp rate” > 0.6 onde o “A” leva uma pequena vantagem.

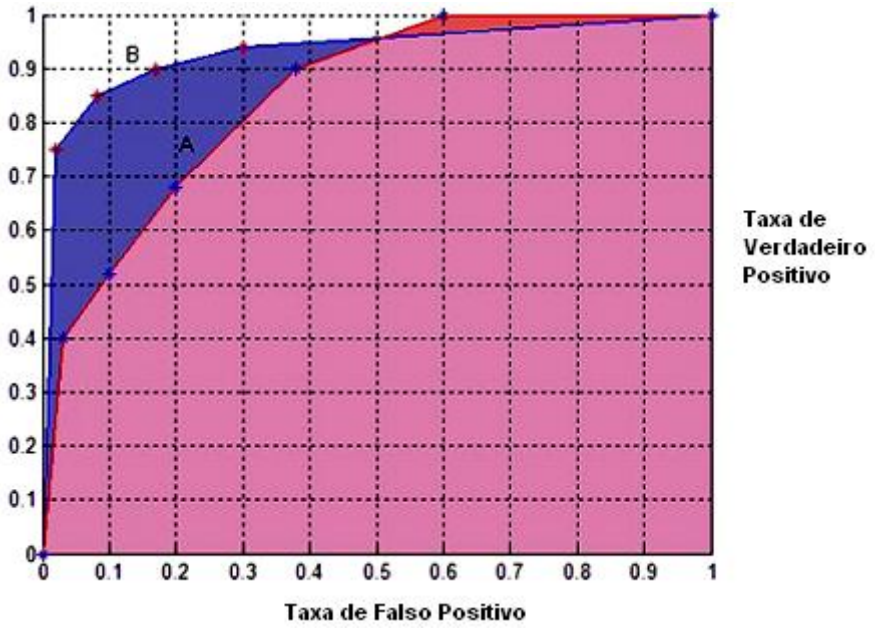


Figura 2.9: Comparativo das áreas abaixo de duas curvas ROC

2.8. O Modelo CRISP-DM

O CRISP-DM (Cross-Industry Standard Process for Data Mining) é um modelo de processo criado em 1996, destinado à elaboração de projetos de Mineração de Dados. O CRISP é descrito hierarquicamente através de um conjunto de tarefas divididas em quatro níveis de abstração, do mais geral para o mais específico: Fases, Tarefas Genéricas, Tarefas Especializadas e Processo de Instâncias. (CRISP, 2009)

O primeiro nível é constituído de seis Fases: Entendimento do Negócio, Entendimento dos Dados, Preparação dos Dados, Modelagem, Avaliação do Modelo e Implantação. A seqüência das fases pertence a um ciclo que é percorrido a cada revisão, sempre que necessária. Cada fase depende de uma anterior.

A Figura 2.1 mostra as fases de um processo de mineração de dados. A sequência das fases não é rígida. A constante revisão das fases é sempre necessária. Cada fase depende de uma anterior. As setas indicam as mais importantes e mais frequentes dependências entre fases:

Figura 2.10 – Fases de um Processo de Mineração de Dados pelo CRISP-DM

O segundo nível, Tarefas Genéricas, é chamado genérico porque pretende ser o mais abrangente possível para cobrir todas as situações possíveis de se aplicar à mineração de dados. Este nível foi projetado para ser o mais completo e estável possível: completo porque deve cobrir todos os processos da mineração de dados e suas possíveis aplicações, estável porque o modelo deve ser válido para as técnicas atuais e futuras. O Quadro 2.5 descreve as tarefas genéricas para cada fase de um processo de mineração de dados.

O terceiro nível, Tarefas Especializadas, descreve como ações de Tarefas Genéricas devem ser encaminhadas em situações específicas.

O quarto nível, Processo de Instâncias, funciona como um registrador, documentando ações, decisões e resultados de um dado processo de mineração de dados em execução.

O Quadro 2.5 apresenta uma descrição sucinta dos processos definidos pelo CRISP-DM:

Quadro 2.5: Tarefas genéricas nas fases de um processo de mineração de dados.

Processo de Mineração de Dados					
Entendimento do Negócio	**Entendimento do Dados**	**Preparação dos Dados**	**Modelagem**	**Avaliação do Modelo**	**Implantação**
Determinar o objetivo do Negócio	Coleta Inicial dos Dados	Seleção de Dados	Seleção da Técnica de Modelagem	Avaliar os Resultados	Plano de Implantação
Avaliar a situação	Descrição dos Dados	Limpeza dos Dados	Gerar o Modelo de Teste	Revisar Processos	Plano de Monitoramento e Manutenção
Determinar os objetivos do Data Mining	Exploração dos Dados	Construção dos Dados	Construir Modelo	Determinar Próximos Passos	Produzir Relatórios Finais
Produzir o Plano do Projeto	Verificação da Qualidade dos Dados	Integração dos Dados	Avaliar Modelo de Teste		Revisar Projeto
		Formatação dos Dados			

3. A Classificação de Despesas Através da Mineração de Texto

Esta seção apresenta as etapas para a extração de conhecimentos e classificação de despesas públicas, aplicando-se técnicas de mineração de textos. Inicialmente os históricos de notas de empenhos foram analisados visualmente por um especialista em auditorias contábeis, em seguida foi executada a primeira classificação dos dados utilizando-se funções de pesquisa em texto do SGBD Postgresql. Daí, os dados foram novamente classificados e avaliados através de algoritmos de mineração de texto do programa WEKA. A apresentação de todo o processo foi baseada no modelo CRISP-DM.

3.1. Entendimento do Negócio

3.1.1. O objetivo do Negócio

Esta pesquisa se contextualiza nas atividades do Tribunal de Contas dos Municípios do Estado do Ceará - TCMCE, Órgão responsável por exercer o controle externo no âmbito dos municípios do Estado do Ceará. O controle externo contempla as atividades de orientação, acompanhamento e fiscalização dos municípios.

A fiscalização aos municípios ocorre basicamente por meio de auditorias nas prestações de contas municipais. Todo o processo de autoria do TCMCE é suportado pelo Sistema de Informações Municipais – SIM. O SIM é utilizado para definir os padrões de prestações de contas municipais em meio informatizado, além de provê os meios para o armazenamento do conteúdo dessas prestações de contas, e fornecer o suporte às atividades de fiscalização, provendo ferramentas para a análise documental.

Considerando o grande volume de dados históricos armazenados no SIM, bem como as dificuldades operacionais decorrentes da necessidade de analisar o conteúdo das prestações de contas municipais, torna-se imprescindível a utilização de uma ferramenta capaz de extrair o conhecimento dos dados armazenados no TCMCE de forma automática, a fim de otimizar o direcionamento das operações de auditorias.

3.1.2 – Avaliação da situação

Para desenvolver o presente processo de mineração de texto foram utilizados os seguintes recursos computacionais:

Programas:

- Sistema Operacional Microsoft Windows XP Professional, service pack 3;
- Microsoft Access 2003 – para migração de dados do SQL 6.5 para SGBD PostgreSQL 8.3.7;
- Banco de Dados Postgresql 8.3.7 – para armazenamento de dados, manipulação de dados e pré-processamento de dados;
- Interface Gráfica SQL Manager versão 4.4.0.5 para PostgreSQL – para Manipulação de dados;
- WEKA 3.7 – na implementação de algoritmos para aprendizado de máquina em processos de Mineração de Dados;

Equipamentos:

- Notebook Dell Vostro 1510, Intel Core 2 Duo, 4 BG de memória RAM, Disco de 250 GB;

Base de Dados:

- A base de dados utilizada foi a do Sistema de Informações Municipais - SIM, dados já auditados do exercício de 2007, cujo acesso foi devidamente autorizado pela Presidência do Tribunal de Contas dos Municípios do Estado do Ceará - TCMCE.

Pessoal:

- Para as tarefas de migração de dados do SGBD SQL Server 6.5 para o Postgresql e suporte em programação, técnicos de informática da Diretoria de Tecnologia de Informação do TCMCE forneceram valioso suporte;
- O acesso aos dados e sua análise foram facilitados, pelo fato do autor desta dissertação ser também o responsável pelo projeto SIM desde sua concepção e por trabalhar como analista no assunto.

Avaliação de resultados:

- Os resultados obtidos na implementação proposta pela presente dissertação foram apresentados em forma de avaliação, contemplando-se os critérios de classificação e as taxas de acertos, resultantes dos testes realizados nos documentos de despesas públicas classificados, através de técnicas de mineração aplicadas à análise automatizada do conteúdo textual dos históricos das notas de empenhos.

3.1.3. Objetivos da Mineração de Texto

Classificar despesas públicas através da análise do conteúdo dos campos de históricos das notas de empenhos, utilizando técnicas de mineração de texto, e avaliar o processo de classificação.

Esta medida facilitará consideravelmente o agrupamento de despesas e consequentemente de fornecedores, o que é básico e imprescindível para otimizar o planejamento dos trabalhos das auditorias desenvolvidos pelo TCMCE.

3.2. Entendimento dos Dados

3.2.1. A Coleta dos Dados

Os dados foram coletados da base de dados do Sistema de Informações Municipais – SIM, instalado no Tribunal de Contas dos Municípios do Estado do Ceará - TCMCE. A base de dados do SIM é estruturada, constituída de várias tabelas integradas, porem, para os fins desta dissertação fez-se necessário apenas o conteúdo dos dados da tabela de notas de empenhos denominada “Empenhos_tb”.

No ato da extração dos dados da tabela “Empenhos_tb” surgiu uma dificuldade relacionada com a incompatibilidade entre SGBD´s - a base de dados original do SIM estava armazenada em um banco de dados Microsoft SQL Server 6.5 e foi migrada para o banco de dados Postgresql 8.3.7, instalado no notebook utilizado para a pesquisa. Para solucionar este problema foi necessária a utilização do gerenciador de dados Microsoft Access 2003, para intermediar o processo de migração - os dados da tabela “Notas de Empenhos” foram transferidos do MS SQL-

Server 6.5 para o MS-Access e, em seguida, importados do MS-Access para o banco de dados Postgresql.

Os dados foram mantidos no mesmo formato que estavam originalmente, até serem pré-processados para a etapa da mineração de texto, mesmo tendo sofrido uma migração de uma plataforma proprietária, pertencente à Microsoft, para o SGBD Postgresql 8.3.7, que pertence a uma plataforma de software livre.

3.2.2. A Descrição dos Dados

Cada registro da tabela “Notas de Empenhos” do SIM contem as informações de uma “Nota de Empenho” específica, que foi emitida por uma administração municipal. A tabela “Notas de Empenhos” do banco de dados contem 34(trinta e quatro) campos distintos, porem, para os fins desta pesquisa, cabe destacar os seguintes:

Quadro 3.1: Campos da tabela Notas de Empenhos

Código da Tabela	Tipo de Campo	Descrição do Campo
cd_municipio	char(3)	Código do Município
cd_elemento_od	char(8)	Código da natureza de despesa do empenho
de_historico_ne1	varchar(255)	Histórico da nota de empenho
de_historico_ne2	varchar(255)	Histórico da nota de empenho

O conteúdo dos campos “de_historico_ne1” e “de_historico_ne2” foi utilizado diretamente no processo de mineração de texto.

3.2.3. A Exploração dos Dados

Durante a verificação inicial do conteúdo dos campos “de_historico_ne1” e “de_historico_ne2”, constatou-se a existência de uma grande variedade de tipos de despesas, o que dificultaria a operacionalização do processo de mineração de texto. Decidiu-se, então, reduzir o conjunto de dados para se contemplar documentos com base em uma “natureza de despesas” específica - “material de consumo”.

Vale salientar que este procedimento de restringir a análise dos dados, com base na natureza de despesa, é uma prática operacional comum nas atividades de auditoria. Portanto, é comum se focalizar a análise de dados em certos tipos de despesas, tais como: Material de Consumo, Material Permanente, Obras e Serviços

de Engenharia.

O presente processo de mineração de texto, portanto, utilizou-se de uma coleção de dados que representa as notas de empenhos de despesas relacionadas à “natureza de despesa 30”, que identifica gastos com “material de consumo”. As notas de empenhos foram extraídas de prestações de contas originadas de 15(quinze) municípios considerados de grande, médio e pequeno porte, localizados em regiões distintas do Estado do Ceará.

Dentro dos documentos relacionados como material de consumo definiu-se, ainda, que a classificação seria aplicada no sentido de se identificar os empenhos destinados às aquisições dos seguintes “objetos de gastos”: Gêneros Alimentícios, Combustível, Material de Expediente, Peças Automotivas, Material para Obras, Material de Limpeza, Medicamentos e Outros.

O passo seguinte da etapa de exploração dos dados foi identificar nas notas de empenhos quais os “termos” que poderiam ser usados, tanto para vincular, quanto para desvincular os documentos de despesa aos tipos de “objetos de gastos” previamente definidos. Para isso, foi feita uma análise reiterada no conteúdo dos campos de histórico de cada nota de empenho selecionada. O Quadro 3.2 apresenta o resultado da análise e lista os termos que foram considerados “vinculantes” e “desvinculantes” a cada objeto de gasto:

Quadro 3.2: Termos considerados Vinculante e Desvinculantes por Objeto de Gasto

Objetos de Gasto:	
Termos Vinculantes	Termos Desvinculantes
Gêneros Alimentícios	
acucar, adocante, garrafao, mineral, bebida, cafe, carne, frango, cereal, cha, condimentos, fruta, legume, refrigerante, refrigerantes, suco, tempero, verdura, lanche, merenda, alimenticio, alimento, alimentar, alimentação, perecivel, peixe, refeicão, refeicoes, cesta, garrafoes, almocos, arroz, feijao, macarrao, cajuina.	veiculos, expediente, limpeza, higiene, lancha, sanitaria, pneu, hidraulico, eletrico

Combustível	
combustiveis, gasolina, alcool, gas, glp, querosene, diesel, lubrificante, graxa.	limpeza, didatico, musica, bomba, chao, condicionado
Material de Expediente	
expediente, carimbos, apagador, apontador, borracha, caderno, caneta, cartolina, classificador, clipe, cola, colchete, corretivo, envelope, estencil, estilete, extrator, fita, giz, grafite, grampeador, grampos, lapis, lapiseira, papel, percevejo, perfurador, registrador, regua, selos, tesoura, transparencia, suprimento, formulario, receituario.	tonner, toner, pecas, medalha, telefone, pneus, computador, odontológico, impressora, paiva, medicamentos, pelicula, limpeza, mudas, hidrômetro, armarios, toalha, construcao, refeicoes, lanches, eletrico, cimento, veiculos, generos, vales, gliter, valvula, fundo, agua.
Peças Automotivas	
pecas, velocimetro, veiculo, placas, automoveis, automotivel, automotivo, pneu, rolamento, automotivo, moto, motocicleta, carro, trator, freio,	combustivel, gas, alcool, aluguel, gasolina, telefonica, computador, pedra, copiadora, predio, equipamento, madeira, exposicao, ponte, digital, informatica, televisores, abastecimento, acompanhamento, impressra, musicais, mesa, elastico, cozinha, piscina, singer, impressora, grades, roupas, fotocopiadora, internet, oxigenio, duplicador, titanio, centrifugas, tonner, mae, grama, medalhas, paciente, Epson, hp, acude, inaugurar
Material para Obras	
construção, ceramica, telha, instalacoes, eletrica, hidraulica, tinta, ferragem, torneira, eletroduto, cimento, tijolo, fechadura, lampada, pintura, janela, porta, portões, ferrolho, caibros, ripas	odontologico, mola, veiculo, automovel, duplicadora, expediente, limpeza, plaqueta, tecido, impressora, cartucho, copiadora, carro, portaria
Material de Limpeza	
etilico, anticorrosivo, balde, capacho, cera, cesto, desinfetante, desodorizante, detergente, escova, espanador, esponja, estopa, flanela, lustra, mangueira, naftalina, panos, limpeza, limpeza, higiênico, higiene, higienização, removedor, rodo, sabão, sabonete, saco, saponaceo, caustica, toalha, vassoura, lavanderia	alimenticio, generos, caneta, dental, gestantes, tubérculo, placas, esterco, bagana, serigrafia, piscina, odontologico, podacao, capinagem, balancas, tecidos, combustível,

Medicamentos	
medicamento, medico, ambulatorial, soro, vacina, vacinação, agulhas, cânulas, cateter, compressa, gaze, dreno, hospitalar, oxigenio, laboratorial, esparadrapo, cirúrgicos, bisturi, seringa, termometro, clinico, amalgama, anestésico, odontologico, odontologica, raio, platina, seringa, traumatologia, sugador, farmacia, medicina, comprimidos.	caes, eutanazia, bovino, animal, solda, higiene, higienizacao, placa, isopor, tinta, reforma, impressos, graficos, veiculo, expediente, tecido, equipamento, copiadora, gerador, impressora, informatica, lampada, formulario, fundo, instalação, botas, limpeza, eletrica, hidraulica, fardamento, lavanderia, gas, glp, tapete, computador, copa, cozinha, paes, pao, gado, jurídica, aftosa, colchao, condicionado, frutas, refeitório, uniforme, domestico, pintura, propaganda, leite, saco, bordado
Outros	
(*pertencem a classe de "outros" todos os documentos que não se vinculam aos objetos definidos acima*)	*(não apresenta termos desvinculantes)*

Os termos utilizados no Quadro 3.2 não apresentam acentuação e cedilhas, no objetivo de se manter a compatibilidade com o formato original dos dados utilizados.

3.2.4. Verificação da Qualidade dos Dados

Inicialmente, observou-se que o conjunto de dados selecionado contem todas as classes de objetos de gastos pré-definidos, o que é muito favorável ao processo de reconhecimento de padrões no processo de mineração de texto.

Analisando-se, porem, o conteúdo dos históricos das notas de empenhos, constatou-se a ocorrência de vários fatores que impactaram desfavoravelmente no processo de reconhecimento de padrões, dificultando, inclusive, o processo de classificação automática. O Quadro 3.3 apresenta um resumo desses fatores, dividindo as ocorrências por documentos de despesas previamente classificados, conforme objetos de gastos pré-definidos:

Quadro 3.3: Fatores impactantes no reconhecimento de padrões por Objetos de Gastos .

Medicamento:

- Despesas para adquirir outros produtos citando termos relacionados com medicamentos. Ex: ...aquisição de “formulários” gráficos para controle de “medicamentos”...
- O mesmo empenho para adquirir gêneros alimentícios e medicamento;

Alimento:

- Empenho para adquirir outros produtos apresentando termos relacionados com alimento. X: aquisição de “pratos descartaveis” para “refeicoes” no hospital;
- O mesmo empenho adquirindo material de limpeza e gêneros alimentícios;
- Empenho para adquirir gêneros alimenticios para animais;
- Erros ortográficos impedindo a identificação de termos. Ex: “Garraoe” no lugar de “Garrafoes”; “alimentcios” no lugar de “alimentícios”
- O mesmo empenho adquirindo material de limpeza e gêneros alimentícios;
- O mesmo empenho adquirindo material de expediente e gêneros alimentícios;
- Termos com significados distintos cujo radical apresenta o mesmo conteúdo. Ex: Lanche e Lancha;
- Termos com significados distintos cujo radical apresenta o mesmo conteúdo. Ex: bebê e bebe;
- O mesmo empenho adquirindo medicamento e cesta básica;
- O mesmo termo com significados distintos. Ex; agua e agua sanitária)
- Erros ortográficos. Ex: “a?ucar” no lugar de “acucar”)
- O mesmo empenho para adquirir alimento e material para obras;

Combustível e Lubrificante:

- O mesmo empenho adquirindo combustível e produto vinculado a alimento. Ex: gas e água;
- O mesmo empenho adquirindo peças automotivas e combustível. Ex: camara de ar e gasolina;
- O mesmo empenho adquirindo combustível e produto vinculado a alimento. Ex: gas e água;
- Empenho com termos vinculados a combustível referindo-se a outro produto. Ex: “gas” medicinal, que corresponde a oxigênio;
- O mesmo empenho adquirindo peças automotivas e combustível. Ex: oleo lubrificante e camara de ar;
- Uso de termos vinculados a combustível referindo-se a outros produtos. Ex: “gas” R 134 para gelágua; gas medicinal que corresponde a oxigênio)
- O mesmo empenho adquirindo gêneros alimentícios e combustível. Ex: alimento e gás de cozinha; água mineral e GLP;
- O mesmo empenho para adquirir combustível e material de limpeza. Ex: Gás e sabão;

Material de Expediente:

- Empenho para adquirir material de expediente e equipamento, sendo que o segundo é termo excludente do primeiro. Ex: aquisição de “switch” (equipamento) e “fita” para impressora;
- Empenho com termos vinculados a material de expediente referindo-se a outro produto. Ex: aquisição de material de “expediente” para a secretaria de “saúde”;
- Empenho apresentando termos vinculados a material de expediente referindo-se a outros produtos. Ex: aquisição de “carpete” na cor “grafite”;
- Empenho para adquirir material de expediente e equipamento, sendo que o segundo é termo excludente do primeiro. Ex: aquisição de “switch” (equipamento) e “fita” para impressora;

Material de Limpeza:

- O mesmo empenho adquirindo material de limpeza e material de expediente);
- Empenhos utilizando termo vinculado à limpeza, referindo-se a outro objeto. Exemplos: ... desbloqueio e “limpeza” de celular; aquisição de “pa e inchada” para “limpeza” de terreno);
- O mesmo empenho adquirindo material didático e material de limpeza;

Material de Construção:

- O mesmo empenho adquirindo material de construção e material de expediente;
- Empenho apresentando termos vinculados a material de obra, referindo-se a outro objeto. Ex: aquisição de camisas polo fio 30 verde;
- O mesmo empenho adquirindo material de construção e material de limpeza;
- Empenho apresentando termo vinculado a material elétrico (obras), referindo-se a serviço de fornecimento de energia elétrica. Neste caso a despesa não se trata nem mesmo de material de consumo e sim de serviço;
- Empenho apresentando termos vinculados a material de obra, referindo-se a outro objeto. Ex: aquisição de “mola” “hidráulica”;
- Empenho apresentando termos vinculados a material de obra, referindo-se a outro objeto. Ex: aquisição de “pecas para computador” em virtude de problemas “eletricos”;
- Empenho apresentando termos vinculados a material de obra, referindo-se a outro objeto. Ex: aquisição de plaquetas para controle de patrimonio pintadas com tinta indelével;
- Empenho apresentando termos vinculados a material de obra, referindo-se a outro objeto. Ex: aquisição de vidro elétrico para veiculo;
- Empenho apresentando termos vinculados a material de obra, referindo-se a outro objeto. Ex: aquisição de “bomba” “hidráulica”;
- Empenho apresentando termo cujo radical é vinculado a material de obras, referindo-se a outros objetos. Ex: concessão de suprimento de fundos, conforme “portaria”)
- O mesmo empenho adquirindo material de construção e material de expediente)
- Empenhos com erros ortográficos. Ex: ... aquisição de “simento” para obras...
- Empenho apresentando termos vinculados a material de obra, referindo-se a outro objeto. Ex: aquisicao da NBR 7229, com as Normas Tecnicas para projeto, “construção” e operacao de sistemas de tanques sépticos ...

Peças Automotivas:

- Empenho apresentando termos vinculados peca automotiva, referindo-se a outro objeto. Ex: aquisição de “pecas” para “central telefônica”;
- Empenho apresentando termos vinculados a material de pecas automotivas, referindo-se a outro objeto. Ex: “aquisição de pedras” com “placas” de latão;
- Empenho apresentando termos vinculados a material de pecas automotivas, referindo-se a outro objeto. Ex: aquisição de “pecas” para maquina copiadora;
- Empenho apresentando termos vinculados a material de pecas automotivas, referindo-se a outro objeto. Ex: aquisição de pecas para rede de internet;
- Empenho apresentando termos vinculados a material de pecas automotivas, referindo-se a outro objeto. Ex: aquisição de “lampadas” para “veículos”;
- Empenho apresentando termos vinculados a material de pecas automotivas, referindo-se a outro objeto. Ex: aquisição de material para exposição de “pecas” de “confecção”...;
- Empenho apresentando termos vinculados a material de pecas automotivas, referindo-se a outro objeto. Ex: “aquisição de “pecas” pra “maquina digital”;
- O mesmo empenho adquirindo pecas automotivas e combustivel);
- Empenho apresentando termos vinculados a material de pecas automotivas, referindo-se a outro objeto. Ex: “aquisição de “material expediente” para controle diário de “veículos”...;
- Empenho apresentando termo vinculado a pecas automotivas, referindo-se a serviço de lavagem de veículos. Neste caso a despesa não se trata nem mesmo de pecas automotivas (material de consumo) e sim de serviço;
- Empenho apresentando termo vinculado a pecas automotivas, referindo-se a serviço de balanceamento de rodas de veículos. Neste caso a despesa não se trata nem mesmo de pecas automotivas (material de consumo) e sim de serviço;
- Empenho apresentando termos vinculados a material de pecas automotivas, referindo-se a outro objeto. Ex: “aquisição de “placa” de “impressora”...;
- Empenho apresentando termos vinculados a material de pecas automotivas, referindo-se a outro objeto. Ex: “aquisição de “placa” para manutenção de adutoras...;
- Empenho apresentando termos vinculados a material de pecas automotivas, referindo-se a outro objeto. Ex: ..aquisicao de material para uso na manutencao na “maquina” de lavar “carro” pertecente a Secretaria...;
- Empenhos apresentando erros ortográfico. Ex: ... aquisição de pecas para “automotivs”;
- Empenho apresentando termos vinculados a material de pecas automotivas, referindo-se a outro objeto. Ex: aquisição de “placa” para “equipamentos de som”...;
- Empenho apresentando termo vinculado a pecas automotivas, referindo-se a serviços mecânicos em veículos. Neste caso a despesa não se trata nem mesmo de pecas automotivas (material de consumo) e sim de serviços de terceiros;

Outros:

- Empenhos classificados como “outros” por apresentar classes de objetos de gasto com termos desvinculantes simultaneamente. Ex: empenhos adquirindo Material de Limpeza e Material de Expediente; empenhos adquirindo gêneros alimentícios, material de limpeza e material de expediente;
- Empenhos classificados como “outros” por apresentar termos que não podem ser utilizados como vinculantes por serem muito abrangentes. Ex: aquisição de material de ...., aquisição de diversos itens de ..., aquisição de material de manutenção ..., despesas para abastecimento ....;

- Empenhos classificados como “outros” devido a erros ortográficos. Ex: material de “expediente” em vez de “expediente”; aquisição de “refeicos” em vez de “refeições”; aquisição de “oxidenio” em vez de “oxigenio”; material de “limpesa” no lugar de “limpeza”; aquisição de “medicamemntos" no lugar de “medicamentos”; gêneros “alimemticios” em vez de “alimenticios” ;

Observação: Os termos do Quadro 3.3 que se encontram sem acentuação e cedilhas mantêm o padrão original dos dados.

O percentual de registros que apresenta fatores influentes, na qualidade da classificação automática dos dados de despesa, atinge a média geral de 5,4%, conforme demonstra a Tabela 3.1.

Tabela 3.1: Percentual de ocorrências de fatores impactantes no reconhecimento de padrões por Objetos de Gastos

Ocorrência de Fatores Impactantes (%)	Alim.	Autom	Comb.	Exped	Limp.	Medic.	Obra	Outro	Média
	3,8	4,7	3	4,7	6	6	3	11,9	5,4

Outros fatores constatados que influenciaram negativamente no resultado da análise de qualidade dos dados selecionados são:

- A presença de termos irrelevantes para o reconhecimento de padrões, também conhecido como “stopwords” – fato comum em qualquer base de dados textual;
- A presença de termos numéricos e termos com caracteres e números.

3.3. Preparação dos Dados

Esta seção apresenta as fases do processo de preparação dos dados, contemplando as etapas de seleção, limpeza, construção, integração e formatação dos dados.

3.3.1. Seleção dos Dados

Os registros utilizados para o processo de mineração de texto foram selecionados do conteúdo dos campos “de_historico_ne1” e “de_historico_ne2”, constantes dos dados da Tabela de “Notas de Empenhos” do SIM.

Inicialmente foram selecionados os dados inerentes às despesas com “material de consumo”. Para isso foi elaborado um filtro, através de uma consulta na linguagem SQL e executada no banco de dados Postgresql, onde foram selecionados os registros que tinham o campo caractere “cd_elemento_od” contendo o quinto e sexto dígitos preenchidos com o valor numérico “30”.

Objetivando manter uma boa representatividade dos dados utilizados no processo de mineração, foram selecionados documentos de despesas originados de 15(quinze) municípios - considerados de pequeno, médio e grande porte e localizados em regiões distintas do Estado do Ceará. O filtro de dados por município também foi executado através de uma consulta na linguagem SQL, no banco de dados Postgresql, utilizando-se códigos específicos de municípios.

Como resultado final da implementação dos critérios de seleção, foram utilizados no processo de mineração de texto um total de 14.072(catorze mil e setenta e dois) registros ou instâncias de documentos de despesas.

3.3.2. A Limpeza dos Dados

O processo de limpeza de dados foi realizado através das funções do banco de dados Postgresql, mais especificamente nas funções de “Full text Search” (pesquisa em texto completo).

O primeiro passo do processo de “Full Text Search” para manipular dados textuais é transformar os termos dos documentos em “Tokens” (termos representados individualmente). Daí, cada documento será representado pelos “tokens” que o compõe.

Os “parsers” (analisadores do postgresql) de pesquisas de texto são responsáveis pela divisão de documentos de textos naturais em “tokens”, assim como a identificação de cada tipo de “token” (exemplo: palavras, números, símbolos (?, !, %, $, #, @) , endereço de e-mail, etc). Os tipos de “tokens” possíveis são definidos pelo “parser” que foi adotado. É importante salientar que um “parser” não altera o texto como um todo, ele simplesmente identifica limites com base na estrutura dos termos presentes no texto analisado.

As funções de “Full Text Searching” do Postgresql possibilitam ainda outras etapas do pré-processamento de documentos, como descrito a seguir:

- Analisar documentos em forma de “tokens”: Identificar diferentes tipos de “tokens”, para que eles possam ser tratados de forma diferente. As classes de “tokens” dependem das particularidades de cada aplicação específica. Portanto, o Postgresql possibilita a pré-definição das classes de “tokens”, conforme aplicação.
- Converter “Tokens” em “Lexemas”: Um lexeme é uma seqüência de caracteres tal qual um ‘token’, mas, que foi devidamente normalizado de modo a padronizar as diferentes formas de apresentação da mesma palavra. A normalização consiste quase sempre em converter letras maiúsculas para minúsculas, em remover sufixos e remover símbolos. Esta propriedade também permite implementar buscas para encontrar a mesma palavra, sem precisar pesquisar todas as suas variantes possíveis. Esta etapa também contempla a eliminação de “Stopwords”, que são palavras tão comuns que se tornam irrelevantes para efeito de classificação de documentos. Em suma, enquanto “tokens” são fragmentos de texto de um documento, “lexemas” são palavras normalizadas e consideradas úteis para a indexação e busca (consultas). O PostgreSQL utiliza dicionários para executar esta conversão.

No Postgresql os dicionários também se mostram como importantes ferramentas para o processo de limpeza dos dados. Os dicionários possibilitam uma refinada camada de controle na normalização e transformação de “tokens” em lexemas. Com dicionários adequados é possível:

- Definir palavras que não devem ser indexadas (palavras que não devam ser consideradas nas consultas);
- Mapear sinônimos para uma única palavra utilizando Dicionário ISPELL;
- Mapear frases para uma única palavra utilizando um Dicionário TESAURO;
- Mapear diferentes variações de uma palavra de uma forma

diferenciada usando um dicionário ISPELL;

- Mapear diferentes variações de uma palavra de uma forma diferenciada usando regras SNOWBALL STEMMER .

No portgresql a definição do tipo de “parser” e dos dicionários a serem utilizados é feita através de uma “configuração” - a “text search configuration” (configuração de pesquisa em texto). O postgresql fornece algumas formas de configurações pré-definidas e também possibilita que estas sejam ajustadas, ou totalmente construídas.

A limpeza dos dados implementada na presente dissertação baseou-se na utilização de um dicionário denominado de “Snowball Stemmer”, e de uma adaptação de uma configuração denominada “portuguese”, ambos fornecidos como padrão pelo SGBD Postgresql. Esta configuração consiste em utilizar o modelo de “Dicionário Simples”, que converte todo o texto, já transformado em “tokens”, para letras minúsculas. O passo seguinte é verificar cada “token” de cada documento, confrontando-os com o conteúdo de um dicionário de “stopwords”. Os “tokens” que tiverem uma correspondência no dicionário serão excluídos, e o restante retornará em forma de “lexema” (ou token normalizado). O outro dicionário utilizado foi o “portuguese_stem”, que tem a função de reduzir as formas variantes de uma expressão para uma raiz, ou “stem”, utilizando a ortografia do idioma específico.

As adaptações efetuadas na configuração “portuguese” fornecida pelo Postgresql foram as seguintes:

- Ampliação da lista de termos do arquivo de Stopwords com base nas expressões contidas na coleção de documentos. Para isso foi utilizada uma função estatística disponível no Postrgresql, que retorna o número de repetições de cada termo na coleção de documentos. Os termos menos expressivos, isto é, que mais se repetiam nos documentos, foram adicionados na lista de stopwords para serem desconsiderados. Vide linhas de comando abaixo, no formato SQL, utilizando função estatística do Postgresql:

```
SELECT* FROM ts_stat('SELECT historico_representa FROM empenhos_tb where cd_elemento_od LIKE \'____30__\'')ORDER BY nentry DESC, ndoc DESC, word LIMIT 900;
```

- Um outro ponto adaptado na configuração “portuguese” foi a seleção de alguns tipos de “tokens” para serem desconsiderados durante a normalização, por serem irrelevantes para o processo de mineração. A tabela a seguir discrimina os tipos de “tokens” que foram desconsiderados e suas respectivas descrições:

Quadro 3.4: Tipos de “tokens” desconsiderados na metodologia

Tipos Desconsiderados	Descrição
numword	palavras, letras e dígitos
asciihword	palavras hifenizadas, todos os ASCII
hword	palavras hifenizadas, dodas as letras
numhword	palavras hifenizadas, letras e dígitos
hword_asciipart	parte de palavras hifenizadas, todos os ASCII
hword_part	parte de palavras hifenizadas, todas as letras
hword_numpart	parte de palavras hifenizadas, letras e dígitos
email	endereços de E-mails
url	URL
url_path	URL path
file	File or path name
sfloat	Notação científica
float	Notação decimal
int	Inteiro com sinal
uint	Inteiro sem sinal
version	Número da Versão

- A linha de comando SQL a seguir foi implementada no banco de dados Postgresql, para selecionar os tokens a serem desconsiderados:

```
ALTER TEXT SEARCH CONFIGURATION nome_configuração  DROP MAPPING
FOR email, url, url_path, numword, asciihword, hword, numhword,
hword_asciipart, hword_part, hword_numpart, sfloat, float, int,
uint, version;
```

3.3.3. A Construção dos Dados

O processo de construção dos dados foi executado pelo SGBD Postgresql, através de funções e tipos de dados específicos, destinados a dar suporte ao processo de “Full Text Searching”.

No SGBD Postgresql o tipo de dado “tsvector” representa um documento em um formato adequado para pesquisas, enquanto o tipo de dado “tsquery” representa o formato adequado para consultas. A função “to_tsvector” é a responsável por normalizar cada termo constante na coleção de documentos a ser utilizada no processo de mineração, transformando cada documento para o tipo “tsvector”, enquanto que a função “to_tsquery” é a responsável por normalizar os termos de consultas, transformando-as no tipo “tsquery”. Em síntese, um processo de “Full Text Searching” baseia-se na busca de correspondências existentes entre o objeto de consulta, constante em uma função de consulta “tsquery”, e os dados disponibilizados por uma função de representação de dados do tipo “tsvector”.

A mesma “configuração” do Postgresql (“portuguese”), da fase de “limpeza dos dados”, foi utilizada pelas funções “to_tsvector” e “to_tsquery”, objetivando selecionar o “parser” (analisador) e os dicionários que atuaram no processo de “full text search”. Dentro dessa sintaxe, também se integram outros operadores e funções destinados a definir as condições a serem atendidas no processo de pesquisa.

Vale salientar que todo o processo de análise, tokenização e limpeza de dados acontecem no ato da execução das funções “to_tsvector” e “to_tsquery”.

A etapa da construção dos dados contemplou inicialmente a criação de uma tabela denominada “empenhos_tb”, destinada a armazenar apenas os dados utilizados no processo de mineração. A tabela “empenhos_tb” foi atualizada através de um processo simplificado de exportação e importação de dados.

Na tabela “empenhos_tb” foi criada uma coluna denominada de “historico_representa”, com dados do tipo “tsvector”, destinada a armazenar o conteúdo concatenado das colunas “de_historico_ne1” e “de_historico_ne”2, extraído da base original do SIM. O processo de atualização de dados da coluna “historico_representa” contemplou simultaneamente os seguintes passos:

- A concatenação do conteúdo das colunas “de_historico_ne1” e “de_historico_ne2” e gravação do resultado na coluna “historico_representa”;
- A análise e tokenização do conteúdo da coluna “historico_representa”;
- A transformação dos “tokens” produzindo os lexemas (com base na configuração adaptada do Postgresql para Full Text Search);

- A limpeza dos dados .

A seguir, a linha de comando no formato SQL utilizada para criar uma coluna do tipo “tsvector”:

```
ALTER TABLE empenhos_tb ADD COLUMN historico_representa tsvector;
```

A seguir as linhas de comando no formato SQL para executar todo o processo de atualização da coluna “historico_representa”. A função “strip”, objetiva excluir pesos ou endereços de “tokens” e a função “coalesce” trata os campos nulos.

```
UPDATE empenhos_tb SET historico_representa = strip ( to_tsvector
('portuguese'                                                    ,
COALESCE(de_historico_ne1,'')||COALESCE(de_historico_ne2,'')));
```

O passo seguinte foi identificar as classes de cada documento, com base no objeto de gasto descrito em seus conteúdos, para que a coleção de documentos pudesse ser utilizada também no processo de treinamento e aprendizado de máquina, já que utilizamos o modelo supervisionado. Lembrando que as classes utilizadas para agrupar os documentos foram: Gêneros Alimentícios, Combustível, Material de Expediente, Peças Automotivas, Material para Obras, Material de Limpeza, Medicamentos e Outros.

Esta classificação inicial, realizada através de uma análise visual de dados, foi executada por um especialista, com uma experiência de 28(vinte oito) anos na área de auditorias em contas públicas que, por coincidência, é o autor desta dissertação. Utilizando-se o suporte de funções de pesquisa em textos, disponíveis no SGBD Postgresql, foram classificados 14072 (catorze mil e setenta e dois) registros de documentos de despesas, de acordo com os “termos vinculantes e desvinculantes a objetos de gastos”, conforme apresentado no Quadro 3.2.

Nesta ocasião, foi criado o campo “classes”, do tipo “texto”, na tabela “empenhos_tb”, destinado a armazenar a identificação das classes dos documentos. Daí, através de uma “query” (consulta) o Postgresql pesquisou o conteúdo da coluna “historico_representa” e, baseado na identificação de “termos vinculantes e desvinculantes a objetos de gastos”, fez a atualização do campo “classes”.

A Tabela 3.2 apresenta as frequências resultantes das classificações nos dados selecionados para o processo de mineração de texto:

Tabela 3.2: Frequência das Classificações por Objetos de Gasto

Objetos de Gastos	Frequencia Absoluta	Frequência Relativa %
Alimento	1106	8,1
Combustível	1665	12,4
Expediente	1266	9,5
Automotiva	1627	11.4
Obras	726	5,0
Limpeza	822	5,7
Medicamento	1200	8,2
Outro	5660	39,7
Total	14072	100,0

A seguir, um trecho das linhas de comandos, no formato SQL, destinadas a identificar a classe com base no conteúdo de cada documento. Observe que após a cláusula “then” estão as definições de cada classe:

```
UPDATE empenhos_tb SET classes =
CASE
  WHEN
   cd_elemento_od LIKE'____30__' AND
   historico_representa@@to_tsquery('portuguese','((combustivel)|(
   gasolina)|(alcool)|(querosene)|gas)|(diesel)|(lubrificante)|(gr
   axa))&(!limpeza)&(!didatico)&(!musica) & (!bomba)')
  THEN 'combustivel'
  WHEN
   cd_elemento_od  LIKE'____30__'  AND    historico_representa  @@
   to_tsquery                                          ('portuguese',
   '((acucar)|(adocante)|(garrafao)|(mineral)|(bebida)|(cafe)|(car
   ne)|(cereal)|(cha)|(condimentos)|(fruta)|(legume)|(refrigerante
   )|(suco)|(tempero)|(verdura)|(merenda)|(alimenticio)|(alimento)
   |(alimentar)|(alimentacao)|(perecivel)|(refeicao)|(refeicoes)|(
   cesta))& (!veiculos)')
  THEN 'alimento'
  ....
```

O passo final da construção dos dados foi a concatenação do conteúdo da coluna “historico_representa” com o conteúdo da coluna “classes”. Para isso foi criado o campo “texto_classes”, do tipo texto, na tabela “empenhos_tb”. Daí, o campo “texto_classes” foi atualizado com o resultado da concatenação dos campos “historico_representa” e “classes”.

Linhas de comandos no formato SQL para atualizar o campo *texto_classes*:

```
UPDATE empenhos_tb SET texto_classes = historico_representa :: text||','||classes;
```

3.3.4. A Integração dos Dados

O processo de integração de dados foi bastante simplificado, considerando que toda a coleção de documentos foi retirada de uma só tabela - “Notas de Empenhos”, resumindo-se à concatenação de campos, conforme citado na etapa de “Construção de Dados”, quais sejam:

- A concatenação das colunas “de_historico_ne1” e “de_historico_ne2”*;*
- A concatenação dos campos “historico_representa” e “classes”

Finalmente, os dados que foram utilizados no processo de mineração de texto ficaram disponíveis no campo “texto_classes” da tabela “empenhos_tb”.

3.3.5. A Formatação dos Dados

A formatação dos dados foi feita através do programa WEKA, que disponibiliza ferramentas destinadas à mineração de dados, desenvolvido no Departamento de Ciência da Computação da Universidade Waikato (Nova Zelândia). O WEKA é um software livre, sob licença GNU, implementado inteiramente na linguagem de programação Java.

O WEKA dispõe das interfaces gráfica e de linhas de comando, por intermédio das quais é possível implementar, dentre outras etapas do processo de mineração, o pré-processamento e a classificação de dados.

No WEKA, o pré-processamento é feito através de algoritmos que implementam filtros e conversores, que são responsáveis pela formatação dos dados para que possam ser utilizados posteriormente pelos algoritmos de mineração. Por padrão, o WEKA para implementar seus algoritmos prescinde que os arquivos de dados tenham os seguintes tipos de extensões: “.arff” , “.csv” , “.data” , “.names” e “.bsi”.

Na fase de formatação dos dados do presente processo de mineração, o primeiro passo foi configurar uma conexão do programa WEKA com o banco de dados Postgresql, para que as funções do WEKA utilizassem automaticamente os

dados da tabela “empenhos_tb”. Para isso, alguns passo foram necessários previamente para se chegar na conexão desejada, senão vejamos:

- Extrair o arquivo “DatabaseUtils.props.postgresql” do arquivo “WEKA.JAR”, que por padrão é salvo na pasta de instalação do WEKA;
- Salvar o arquivo “DatabaseUtils.props.postgresql” na pasta “Home do Usuário”. (Para se descobrir onde se localiza a pasta home do usuário, basta utilizar na console do Windows a variável de ambiente “%USERPROFILE%”);
- Salvar o “driver JDBC” na pasta home do usuário. Este “driver” é responsável pelo processo de conexão lógica do programa WEKA ao banco de dados Postgresql;
- Renomear o arquivo “DatabaseUtils.props.postgresql” para “DatabaseUtils.props”. Este procedimento visa manter o padrão de arquivos reconhecidos pelo WEKA ;
- Editar o arquivo “DatabaseUtils.props” - Dentro do arquivo “DatabaseUtils.props”, no local onde aparece o conteúdo “# database URL”, alterar o conteúdo da linha existente para: “jdbcURL=jdbc:postgresql://localhost:5432/mestrado_db”.Este comando utiliza o “driver JDBC” para iniciar o processo de conexão do WEKA com a base de dados “mestrado_db” do Postgresql, onde está a tabela “empenhos_tb”.

Concluída a fase preliminar de conexão do WEKA ao banco Postgresql, foi possível iniciar o processo de formatação dos dados. Esta etapa contemplou os seguintes passos:

- Extração de dados dos campos “texto_classes” e “classes”, da tabela “empenhos_tb” do banco Postgresql;
- Conversão dos dados da coleção para arquivos no formato “.arff ”, que é um formato reconhecido pelo WEKA;
- Gravação dos dados em arquivos no padrão reconhecido pelo WEKA;

Para executar a etapa de formatação dos dados, foi utilizada a interface de linha de comando “SimpleCLI” do WEKA, onde foram inseridas as seguintes linhas de comandos na linguagem Java:

```
java weka.core.converters.DatabaseLoader -user postgres -password
8110100 -Q "SELECT classes,texto_classes FROM
mestrado_db.public.empenhos_tb" > data/arquivo_destino.arff
```

Observe-se que o comando acima concluiu o processo de conexão ao banco, quando passou para o banco os parâmetros de “user” (usuário) e “password” (senha). Também implementou uma consulta no formato SQL, através do comando de seleção de dados “Select”. Daí, o conteúdo retornado foi redirecionado para o arquivo “arquivo_destino.arff” na pasta “...WEKA \ dados”.

O passo seguinte do processo de formatação foi a conversão do arquivo “arquivo_destino.arff” para o formato de “vetor”, e o redirecionamento do resultado para o arquivo “arquivo_destino_vector.arff”, na pasta “...WEKA\dados”. Este procedimento foi necessário para que o WEKA pudesse aplicar os algoritmos de classificação em base de dados textuais. Para isso também foi utilizada a interface de linha de comando “SimpleCLI” do WEKA. Vide a seguir a linha de comando:

```
java weka.filters.unsupervised.attribute.StringToWordVector -i
data/ arquivo_destino.arff -o data/ arquivo_destino_vector.arff -c
last
```

A etapa final da formatação foi a reordenação do arquivo “arquivo_destino_vector.arff”, para transferir o atributo de classe para o final de cada vetor, ou instância, e redirecionar o retorno obtido para o arquivo “arquivo_destino_vector_reorder.arff”, na pasta “...WEKA\dados”. O WEKA foi parametrizado para reconhecer como o atributo de classe aquele que se localiza no final de cada instância, por este motivo foi aplicada esta reordenação. Para reordenar, também foi utilizada a interface de linha de comando “SimpleCLI” do WEKA, com o seguinte conteúdo:

```
java weka.filters.unsupervised.attribute.Reorder -R 2 -last,first
-i data / arquivo_destino_vector.arff -o data /
arquivo_destino_vector_reorder.arff
```

3.4. Modelagem

Esta seção apresenta as etapas de criação de modelos de classificação de dados textuais, utilizando técnicas de aprendizado de máquina. Através dos recursos disponíveis no programa WEKA, os dados previamente classificados por objetos de gastos foram submetidos a algoritmos, que executaram os processos de

classificação e avaliação dos resultados.

As técnicas de modelagem, selecionadas para o processo de mineração de texto, foram originadas das funções disponíveis na interface gráfica do programa WEKA identificada como “Explorer”. O “WEKA Explorer” permite acessar dados diretamente de um arquivo, de uma base de dados, ou de um endereço de internet, além de possibilitar a geração de dados para experimentos. Outra funcionalidade é o fornecimento de filtros e conversores destinados às etapas de pré-processamento de dados. Mas, o ponto máximo do “WEKA Explorer” é a disponibilização de vários algoritmos que se destinam à Classificação, Clusterização, Regras de Associação e Seleções de Atributos, que atuam de forma integrada dentro do processo de mineração.

O “WEKA Explorer” disponibiliza várias seções onde se encontram as configurações e funções disponíveis para o processo de mineração de dados.

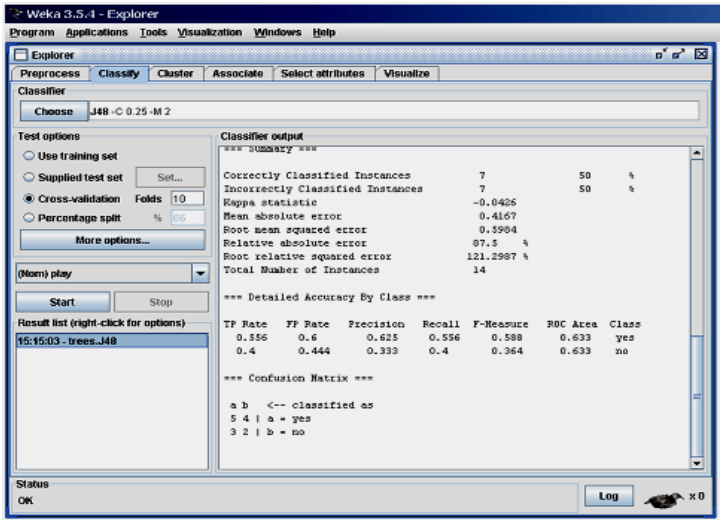


Figura 3.1: Tela do WEKA, na seção de Classificação de Dados

A seguir serão abordadas algumas opções de configurações do WEKA, que foram parametrizadas para o processo de classificação de dados da presente dissertação:

- A Seleção do Classificador:

  Antes que o classificador seja selecionado é necessário que os dados a serem utilizados já estejam previamente formatados e acessados pelo WEKA. Daí, através do botão “choose” é possível selecionar o classificador desejado. Estão disponíveis várias representações de

algoritmos destinados à classificação – Classificador bayesiano, Árvore de decisão, Redes Neurais, Regras de Associação, Algoritmo Preguiçoso e outros. O “Explorer” somente disponibiliza para seleção os classificadores compatíveis com o tipo de dado disponível para o processo de classificação.

- As Propriedades do Classificador:

  Para cada modelo de classificador o “Explorer” disponibiliza várias propriedades, que podem ser parametrizadas de acordo com as características dos dados e da tarefa a ser realizada. Para isso, basta acessar a caixa “choose” com o botão direito do mouse.

- As opções de Teste:

  Para testar a qualidade do classificador selecionado, o “WEKA” disponibiliza quatro opções de teste. Quais sejam:

  “Use training set” : o classificador é avaliado sobre sua qualidade em predizer a classe dos casos em que foi treinado;

  “Supplied test set” : o classificador é avaliado sobre sua qualidade em predizer a classe de um conjunto de instâncias carregadas a partir de um arquivo;

  “Cross-validation”: o classificador é avaliado pelo *cross-validation*, utilizando o número de folds que estão inscritos no campo texto *folds*.

  “Percentage split”: O classificador é avaliado sobre o quão bem ele prediz uma certa percentagem dos dados que foi utilizado para o teste. A quantidade de dados considerada depende do valor inscrito no campo “%”.

- O Atributo de Classe:

  Os classificadores no WEKA são concebidos para serem treinados e para predizerem uma única "classe" de atributo, que é o objetivo da predição. Alguns classificadores só podem aprender classes nominais, outros só podem aprender classes numéricas (como nos problemas de regressão) e outros ainda podem aprender ambos.

Por padrão, a classe é definida para ser o último atributo dentro da estrutura dos dados. Porem, o “Explorer” permite treinar um classificador para prever um atributo diferente através da caixa “*Test options”.*

- Os Resultados Apresentados pelo Classificador:

  Uma vez configurado o classificador, opções de teste e classes, o processo de aprendizagem poderá ser iniciado através do botão “Start”.

  Quando o treinamento é concluído, a área de saída do classificador (“Classifier output”) é preenchida com um texto descrevendo os resultados de treinamento e testes, e uma nova entrada aparece na caixa “Result list”, que serve para identificar um modelo e seus resultados de treino e teste.

  Ao final de um processo de classificação, são apresentados vários resultados na janela de saída do classificador (“*Classifier outpu”t*). Quais sejam:

  “Run information”: uma lista de informações relacionadas com o esquema de aprendizagem - relação de nome, instâncias, atributos e modelo de teste que foram envolvidos no processo.

  Classifier model: uma representação textual do modelo de classificação que foi aplicado sobre o treino completo dos dados.

  “Summary”: uma lista de estatísticas resumindo o quão preciso foi o classificador na previsão da verdadeira classe das instâncias no tipo de teste escolhido.

  “Detailed Accuracy By Class”: um detalhamento por classe, da precisão da predição do classificador.

  “Confusion Matrix”: mostra quantas instâncias foram atribuídos a cada classe, assim como os resultados obtidos na classificação no formato de matriz de confusão.

  “Source Code”: esta seção lista o código fonte Java utilizado no processo de classificação, se tiver sido selecionada a opção “Output source code” na caixa de diálogo “More options”.

- O campo *“Result list”* (Lista de Resultados):

  Após treinar vários classificadores, a lista de resultados conterá várias entradas, onde cada entrada identifica um modelo e seus resultados de treino e teste.

  Ao selecionar um classificador, através do botão direito do mouse, se tem acesso a várias opções relacionadas com as seguintes ações: visualizar resultados através de listagens ou recursos gráficos, salvar resultados, salvar modelos, utilizar modelos previamente construídos, entre outras.

3.4.1. Seleção da Técnica de Modelagem

No objetivo de implementar a classificação de dados textuais, considerando as características dos dados utilizados, através da interface “WEKA Explorer” foi selecionado o classificador J48, baseado em árvore de decisão.

3.4.2. A Geração dos Modelos e Resultados Apresentados pelo Weka

Ao se iniciar o processo de classificação, o WEKA acessa os dados dos arquivos pré-formatados (Ex: “arquivo_destino_vector_reorder.arff”), gera os modelos de classificadores usando o algoritmo J48 e em seguida os avalia, concluindo todas as etapas de classificação de forma sequenciada e integrada.

Para a geração dos modelos de classificadores, foram utilizados os dados construídos previamente e apresentados no tópico “3.3.3. A Construção dos Dados”. Porem, considerando-se a necessidade de um estudo no comportamento do algoritmo J48 para classificar “dados textuais”, foram extraídas, da coleção de 14.070 documentos de despesas, cinco amostras aleatórias de 500, 1000, 3000, 6000 e 9000 registros e uma amostra de 12000 registros. Saliente-se, ainda, que as amostras mantiveram as mesmas frequências relativas de cada objeto de gasto, apresentada na Tabela 3.2 “Frequência das Classificações por Objetos de Gasto”.

A seguir, a linhas de comando no formato SQL utilizada para gerar as amostras aleatórias supra citadas:

```
INSERT INTO public."tabela_destino"
SELECT
historico_representa,
classes,
texto_classes
FROM tabela_origem
WHERE classes = 'objeto_gasto'
ORDER BY RANDOM()
LIMIT valor_numerico
```

Objetivando alcançar os melhores resultados na geração dos modelos de classificadores, com base nos resultados de diversos experimentos, adotou-se os seguintes parâmetros para o algoritmo J48 :

- “binarySplits – false” (a opção “false” não usa divisões binárias em atributos nominais na construção das árvores);
- “confidenceFactor – 0,5” (optou-se por 0,5 o fator de confiança utilizado para a poda, pois quanto menor o valor maior a incorrência de poda);
- “debug – false” (a opção “false” rejeita a saída de informações adicionais pela console);
- “minNumObj – 2” (optou-se por 2 o número mínimo de instâncias por folha);
- “numFolds – 3” (optou-se por 3 a quantidade de dados usada para a redução de erro por poda. Uma partição é usado para poda, o resto para o crescimento da árvore;
- “reducedErrorPruning” – false (a opção “false” não usa a redução de erro por poda no lugar da poda C.4.5);
- “saveInstanceData” – false (a opção “false” não salva os dados de treinamento para a visualização);
- “seed – 1” (a opção 1 define as “seeds” utilizadas para randomizar os dados quando a redução de erro por poda é utilizada);
- “subtreeRaising – true” (a opção “true” considera as operações de subárvore quando utilizar a poda);
- “unpruned – false” (a opção “false” admite a realização da poda);
- “useLaplace – false” (a opção “false” define que a contagem de folhas da árvore não seja arredondada com base em Laplace).

Os quadros a seguir apresentam os resultados do processo executado pelos classificadores gerados pelo algoritmo J48 (WEKA), através do percentual de registro que foi classificado corretamente em cada amostra:

a) Amostras de 500 registros:

Tabela 3.3: Percentuais de registros classificados corretamente por Classificador - 500

Amostra	A	B	C	D	E	Média	Desv.P.
Percentuais de acertos (%)	44	47,6	45,5	45	46,8	45,8	1,17

Tabela 3.4: Médias de Registros classificados corretamente por Objeto de Gasto - 500

Resultados	Alim.	Automot	Comb	Exped	Limp	Medic	Obra	Outro
Nº Registros Total	41	62	48	57	25	28	41	198
Média de Acertos	5	3	4	6	2	10	0	198
Média de Acertos(%)	11,2	5,2	9,2	11,2	6,4	36,4	0	100
Desvio P.das Médias	31,02	---						

b) Amostras de 1000 registros:

Tabela 3.5: Percentuais de registros classificados corretamente por Classificador - 1000

Amostra	A	B	C	D	E	Média	Desv.P.
Percentuais de acertos (%)	48,1	49,1	49,5	50,5	50,6	49,6	0,84

Tabela 3.6: Médias de Registros classificados corretamente por Objeto de Gasto - 1000

Resultados	Alim.	Automot	Comb	Exped	Limp	Medic	Obra	Outro
Nº Registros Total	81	124	95	114	50	57	82	397
Média de Acertos	16	13	22	20	4	23	1	397
Média de Acertos(%)	19,3	10,7	22,7	17,4	8	40,7	1,5	100
Desvio P.das Médias	29,51	---						

c) Amostras de 3000 registros:

Tabela 3.7: Percentuais de registros classificados corretamente por Classificador - 3000

Amostra	A	B	C	D	E	Média	Desv.P.
Percentuais de acertos (%)	57,1	58,1	58,4	58,3	57,1	57,8	0,53

Tabela 3.8: Médias de Registros classificados corretamente por Objeto de Gasto - 3000

Resultados	Alim.	Automot	Comb	Exped	Limp	Medic	Obra	Outro
Nº Registros Total	243	373	284	341	150	170	247	1192
Média de Acertos	76	82	123	112	31	105	13	1192
Média de Acertos(%)	31,2	21,9	43,4	32,9	20,4	62	5,3	100
Desvio P.das Médias	27,7	---						

d) Amostras de 6000 registros:

Tabela 3.9: Percentuais de registros classificados corretamente por Classificador - 6000

Amostra	A	B	C	D	E	Média	Desv.P.
Percentuais de acertos (%)	64	64	63,4	63,1	64,1	63,7	0,36

Tabela 3.10: Médias de Registros classificados corretamente por Objeto de Gasto - 6000

Resultados	Alim.	Automot	Comb	Exped	Limp	Medic	Obra	Outro
Nº Registros Total	487	745	567	682	301	340	494	2384
Média de Acertos	201	212	336	286	93	265	47	2384
Média de Acertos(%)	41,2	28,4	59,3	41,9	30,8	77,9	9,4	100
Desvio P.das Médias	27,32	---						

e) Amostras de 9000 registros:

Tabela 3.11: Percentuais de registros classificados corretamente por Classificador - 9000

Amostra	A	B	C	D	E	Média	Desv.P.
Percentuais de acertos (%)	66,8	66,6	66,6	66,6	66,4	66,6	0,12

Tabela 3.12: Médias de Registros classificados corretamente por Objeto de Gasto - 9000

Resultados	Alim.	Automot	Comb	Exped	Limp	Medic	Obra	Outro
Nº Registros Total	730	1116	852	1024	451	510	741	3576
Média de Acertos	332	366	557	471	162	445	86	3576
Média de Acertos(%)	45,5	32,8	65,4	46	35,9	87,2	11,6	100
Desvio P.das Médias	27,51	---						

f) Amostra de 12.000 registros:

Percentual de acerto: 70,58%

Tabela 3.13: Médias de Registros classificados corretamente por Objeto de Gasto - 12000

Resultados	Alim.	Automot	Comb	Exped	Limp	Medic	Obra	Outro
Nº Registros Total	981	1499	1143	1375	606	685	996	4795
Nº de Acertos	496	584	842	695	299	652	164	4795
Média de Acertos (%)	50,6	39	73,7	50,6	49,3	95,2	16,5	100
Desvio P.das Médias	26,58	---						

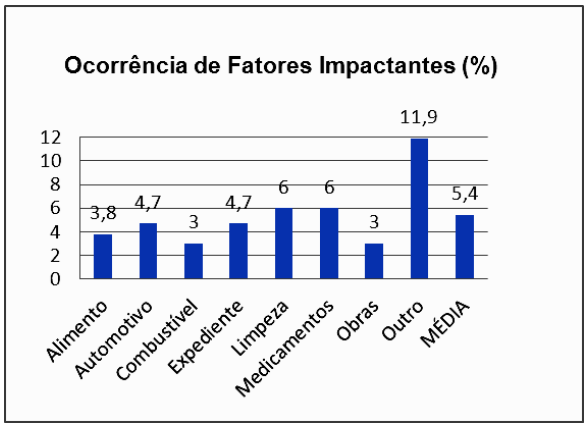
4. Avaliação

Esta seção apresenta uma avaliação geral de resultados, abordando inicialmente o processo de análise visual dos dados e classificação através funções de “full text search” do Postgresql. Em seguida se apresenta um processo de classificação e avaliação, utilizando-se ferramentas de mineração do WEKA. Ao final, é avaliada a solução do processo de mineração de textos no escopo das atividades do TCMCE.

O processo de classificação com base em análise visual, descrito no Item “3.3.3. A Construção dos Dados”, apresentou uma grande dificuldade operacional, considerando a necessidade de se analisar visualmente cada histórico de nota de empenho da coleção. Porém, este método permitiu a descoberta de termos, que foram utilizados para vincular ou desvincular documentos de despesas a objetos de gastos pré-definidos. Tal descoberta possibilita, inclusive, a utilização deste conhecimento adquirido em outras coleções de dados com características semelhantes.

Mesmo através da análise visual de cada um dos 14.072 (catorze mil e setenta e dois) registros de documentos de despesas, o erro de classificação não pôde ser descartado. Tal possibilidade decorreu da existência de fatores presentes nos dados, que dificultam o reconhecimento de padrões que vinculam documentos a objetos de gastos. Uma lista desses fatores foi apresentada no Quadro 3.3. A média percentual de erros decorrentes desses fatores é de 5,4% dos registros da coleção, conforme demonstrado no gráfico do Gráfico 4.1.

Gráfico 4.1: Percentual de registros com fatores impactantes por objetos de gasto.



O conhecimento adquirido com o uso de termos vinculantes e desvinculantes, para classificar documentos de despesas por objetos de gastos, se adequou inteiramente ao processo de classificação de documentos através das funções de “full text search” do Postgresql. Com a utilização desses termos, o Postgresql reproduziu fielmente a classificação obtida com a análise visual dos dados, inclusive a possível ocorrência de erros de classificação a uma taxa de 5,4% (Gráfico 4.1).

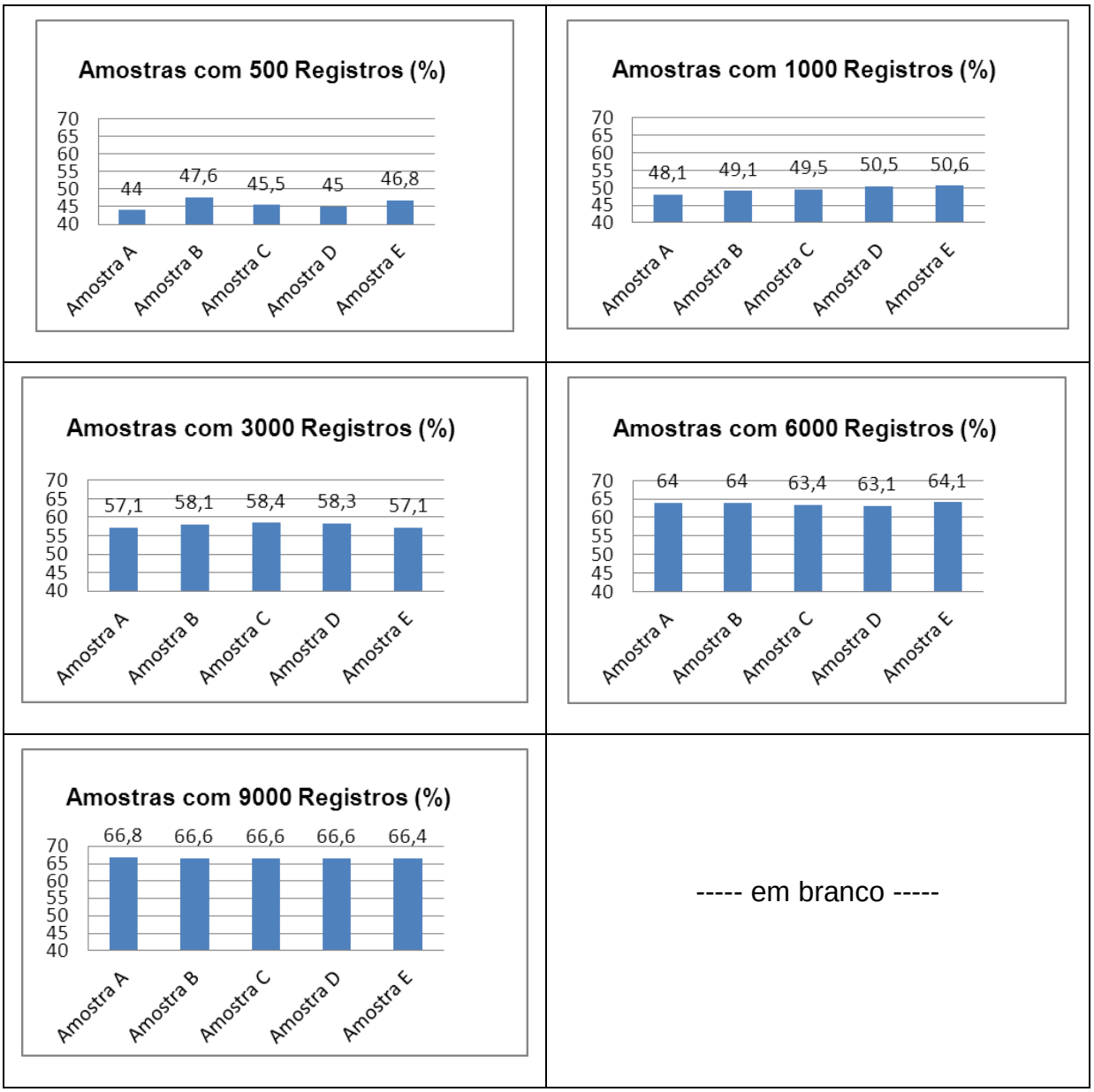
Com a coleção de 14.072 (catorze mil e setenta e dois) registros, devidamente classificados, foi possível implementar um outro processo de classificação, agora baseado em aprendizado supervisionado. Este processo foi desenvolvido através do programa WEKA, com o algoritmo J48, para efetuar a classificação dos dados e o método Cross Validation para avaliar os resultados.

Considerando-se a garantia de que aproximadamente 94,6% dos registros da coleção foram classificados corretamente, o processo foi conduzido no sentido de se conhecer o comportamento do algoritmo J48 na classificação de dados textuais, bem como a avaliação dos resultados pelo método Cross Validation.

Para se analisar o classificador J48 e o método Cross Validation, foram selecionadas aleatoriamente 5(cinco) amostras de 500(quinhentos), 1000(mil), 3000(três mil), 6000(seis mil) e 9000(nove mil) registros e 1(uma) amostra de 12.000(doze mil) registros da coleção total, mantendo-se as frequências relativas por objetos de gastos da coleção completa, conforme apresentado na Tabela 3.2. Todas as amostras foram classificadas através do algoritmo J48 e os resultados foram avaliados pelo método Cross Validation. Os parágrafos e quadros seguintes apresentam as constatações obtidas:

Constatou-se a regularidade na quantidade de classificações corretas, quando se observou as amostras com mesmo número de registros. Constatou-se a regularidade, também, na quantidade de classificações corretas com o aumento do número de registros nas amostras. Vide Gráficos 4.2.

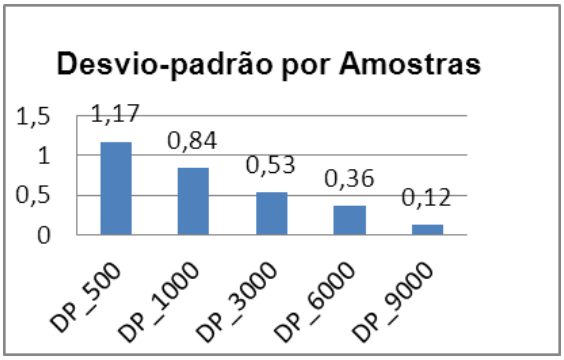
Gráficos 4.2 : Percentuais de classificações corretas por amostras.



----- em branco -----

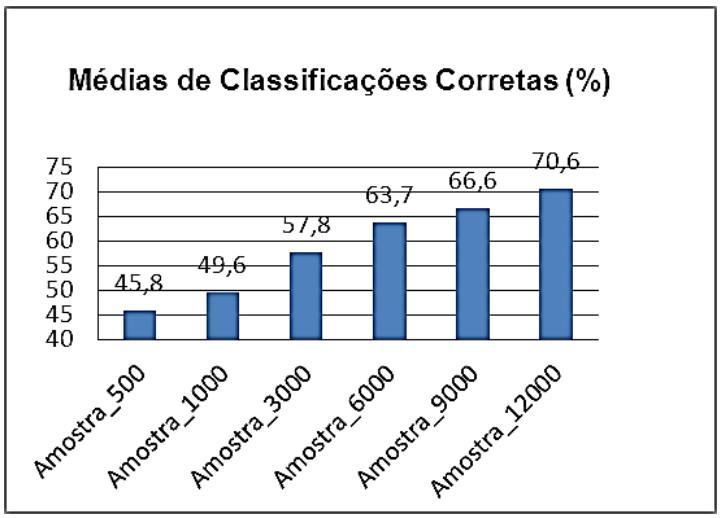
Constatou-se que os valores dos desvios-padrão, das taxas de acertos de amostras com mesmo número de registros, diminuíram, à proporção que o número de registros por amostra aumentou. Vide Gráfico 4.3:

Gráfico 4.3: Desvio-padrão da taxa de acertos por número de registros das amostras.



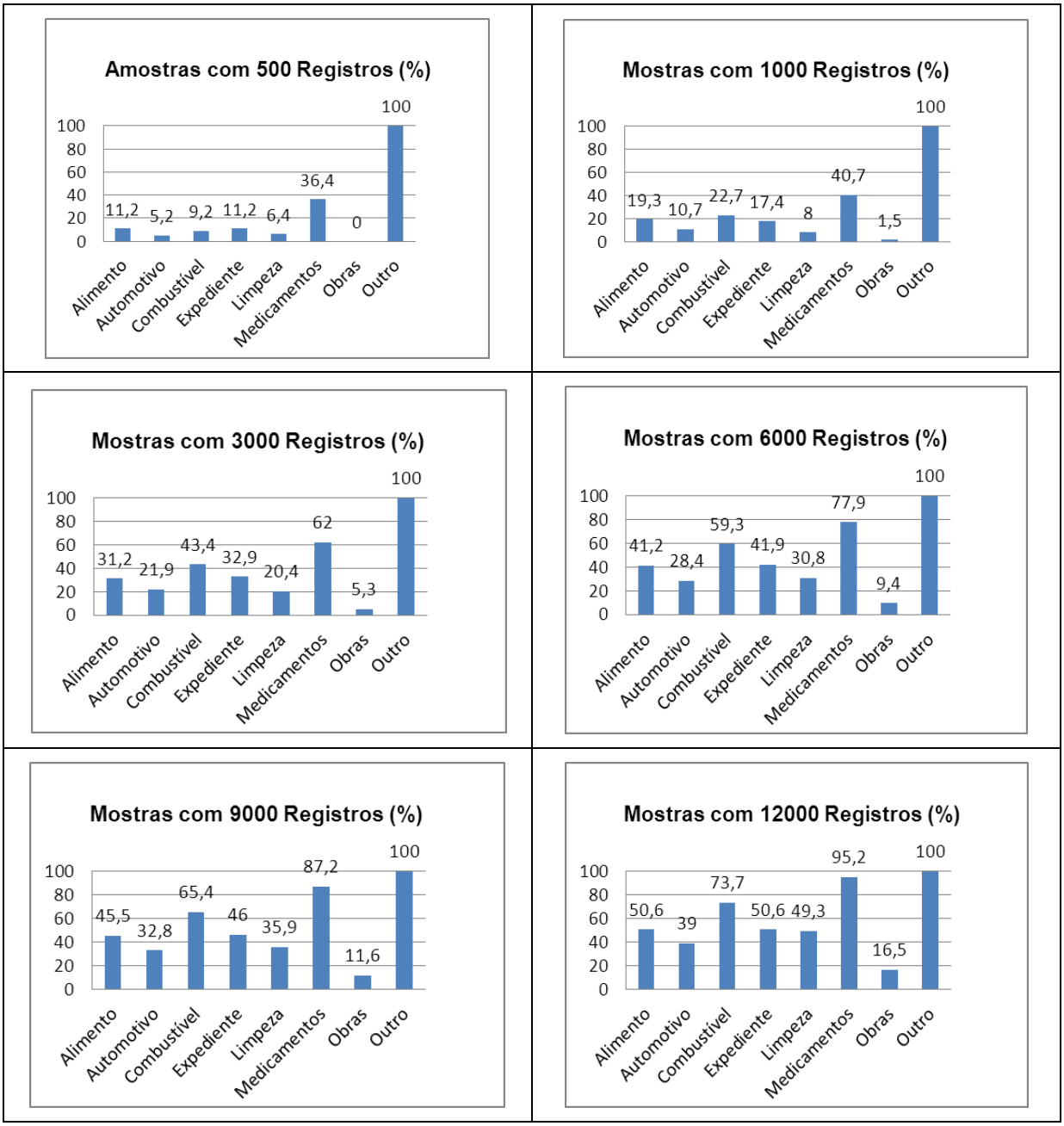
Constatou-se que as médias das classificações corretas aumentaram regularmente, com o aumento do número de registros das amostras. Vide Gráfico 4.4:

Gráfico 4.4: Médias de classificações corretas por amostras.



Constatou-se que as médias percentuais, de classificações corretas por objetos de gastos, aumentaram regularmente, com o aumento do número de registros das amostras. Vide Gráficos 4.5:

Gráficos 4.5: Médias percentuais de classificações corretas das amostras por objetos de gastos.



Os meios utilizados nas etapas definidas para a classificação de despesas, através da Mineração de Texto, mostram-se perfeitamente compatíveis com as disponibilidades técnicas do TCMCE. Pois, além de requerer uma infraestrutura mínima, em termos de equipamento, os programas utilizados no processo de mineração de texto pertencem à plataforma de software livre (Postgresql e Weka).

Para as atividades de fiscalização do TCMCE, a classificação de documentos visa agrupar despesas por objetos de gastos e, consequentemente, identificar tipos de fornecedores de produtos específicos. Isto é, para que um fornecedor seja vinculado a um objeto de gasto, basta que este seja identificado em um único documento de despesa. Portanto, a taxa de acertos de 94,6% das classificações é mais que suficiente para a referida identificação.

Outro requisito considerável é que o objetivo das auditorias é contemplar “amostras representativas” das despesas de uma gestão municipal, e não a totalidade das despesas. Portanto, trabalhar com uma coleção de dados, em que se garante a classificação correta de aproximadamente 94,6% dos registros de despesas, é mais que o necessário para se atingir os objetivos almejados, considerando-se, portanto, mínimo o “custo” do erro projetado.

É importante para as atividades de aditorias, porem, a existência de meios que demonstrem a qualidade das classificações em dados desconhecidos de uma forma prática, facilitada e, de preferência, através de procedimentos automatizados. Neste caso, é necessário que se conheça previamente o comportamento das ferramentas de classificação e avaliação de resultados para o tipo de dado utilizado.

O processo de mineração de texto em apreço utilizou dados textuais, sem qualquer regularidade quantitativa e qualitativa nos termos constante em cada documento a ser minerado. Isto é, a estrutura dos dados utilizados fugiu totalmente aos padrões convencionais. Portanto, inicialmente foi necessário conhecer como o algoritmo J48 e o método “Cross Validation” se comportariam, durante o processamento das amostras de dados, sabendo-se que aproximadamente 94,6% dos registros estão classificados corretamente.

Os gráficos 4.2, 4.3, 4.4 e 4.5 desta seção, resumiram os resultados obtidos com a utilização do algoritmo J48 e do método de avaliação “Cross Validation”, no processo de classificação de documentos de despesas por objetos de gastos, com as seguinte constatações:

- Os resultados das classificações das amostras apresentaram percentuais de acertos abaixo de 94,6% (Tabelas: 3.3, 3.5, 3.7, 3.9, 3.11), portanto, abaixo da taxa de acertos encontrada com a classificação através da análise visual dos dados e das funções do

Postgresql;

- Ocorreu regularidade na quantidade de classificações corretas, quando se observou amostras com mesmo número de registros e com o aumento do número de registros nas amostras (Gráfico 4.2);
- Os valores dos desvios padrão, das taxas de acertos de amostras com mesmo número de registros, diminuem, à proporção que o número de registros por amostra aumenta (Gráfico 4.3);
- As médias das classificações corretas aumentam regularmente, com o aumento dos números de registros das amostras (Gráfico 4.4);
- As médias percentuais, de classificações corretas por objetos de gastos, aumentam regularmente, com o aumento do número de registros das amostras (Gráficos 4.5).

5. Conclusões

As técnicas utilizadas nesta dissertação atingiram os objetivos gerais de classificar despesas públicas, com a análise do conteúdo dos campos de históricos das notas de empenhos através da mineração de texto. Utilizando-se o modelo de projetos CRISP-DM, foram desenvolvidas as etapas do processo de Mineração de Texto, envolvendo Pré-processamento de Dados Textuais, Processamento de Linguagem Natural – PLN e Mineração de Dados, atingindo-se, ao final, o objetivo de se atribuir classes às notas de empenhos, com base no conteúdo textual de campos de históricos de despesas.

A análise visual de 14.072 registros de despesas resultou na descoberta dos termos a serem utilizados, para vincular e desvincular cada documento aos objetos de gastos pré-definidos (Tabela 3.2). Tal procedimento convergiu com o objetivo de se identificar fornecedores por objetos de gastos, uma vez que, classificado o documento, se chega ao fornecedor do serviço ou bem adquirido. O conhecimento obtido nesta fase de análise foi totalmente transferido para as funções de “full text search” do Postgresql, possibilitando o pré-processamento, a análise e a classificação automática dos documentos de despesa, a uma taxa de acerto de aproximadamente 94,6% dos registros.

Com os dados previamente classificados, utilizando-se o algoritmo J48, foi implementada a etapa de aprendizado supervisionado e, em seguida, a criação de modelos de classificadores, que executaram, novamente, a classificação automatizada dos dados. Este último processo possibilitou a utilização do método “Cross Validation”, para a avaliação de resultados de forma automatizada.

Nessa oportunidade, foram constatadas divergências entre as taxas de acertos nas classificações de dados obtidas pelo Postgresql e pelo algoritmo J48, mesmo utilizando-se dados semelhantes em ambos os processos. Porem, ao se analisar o assunto à luz dos quadros demonstrativos da Seção 4. “Avaliação”, constatou-se que tais divergências aconteceram de forma muito regular, com a variação do número de registros das amostras avaliadas. O que evidenciou, como principal causa das divergências, o comportamento do algoritmo J48 e do método “Cross Validation” quando processam dados textuais com as características utilizadas.

Concluiu-se, então, que as taxas de acertos nas classificações de documentos de despesas municipais por objetos de gastos, obtidas através do algoritmo J48 e do método “Cross Validation”, podem ser utilizadas como parâmetros comparativos para avaliar bases de dados com características semelhantes, desde que sejam acrescentados fatores compensatórios relacionados com o tamanho das amostras utilizadas.

Finalmente, considerando-se aspectos operacionais, as técnicas de mineração de texto apresentadas nesta dissertação mostram-se aplicáveis às atividades de fiscalização do TCMCE. Portanto, a demanda de classificar despesas por objetos de gastos e a identificação de fornecedores, com ferramentas informatizadas, pode ser atendida de forma satisfatória a um baixo custo operacional.

6. Trabalhos Futuros

Desenvolver uma metodologia otimizada, para descobrir termos constantes em empenhos de despesas públicas, que vinculem tais documentos a classes temáticas pré-definidas.

7. Referências Bibliográficas

ALPAYDIN; Ethem. **Introduction to Machine Learn**. MIT, Cambridge, 2004. 423 p.

ARANHA, Christian Nunes. **Uma Abordagem de Pré-Processamento Automático para Mineração de Textos em Português**: Sob o Enfoque da Inteligência Computacional. 2007, 144 p. Tese de Doutorado - Programa de Pós-graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica – Pontífice Universidade Católica do Rio de Janeiro, Rio e janeiro – RJ.

BERRY, Michael W. . University of Tennessee. **Survey of Text Mining, Clustering, Classification e Retrieval**. USA. 2004. 241 p.

BERRY, Michael W.; Kogan, Jacob. **Text Mining Application and Theory**. Reino Unido. 2010. 205 p.

BOUCKAERT, Remco R.; FRANK, Eibe; HALL, Mark; KIRKBY, Richard; Peter; REUTEMANN, SEEWALD, Alex; SCUSE, David. **WEKA Manual for Version 3-7-0**. University of Waikato, Hamilton, New Zealand. June 4, 2009, 214 p.

BRASIL. **Portaria Interministerial no 163**, de 04 de maio de 2001. Dispõe sobre normas gerais de consolidação das Contas Públicas no âmbito da União, Estados, Distrito Federal e Municípios, e dá outras providências.

BRASIL. **Portaria Nº 448, de 13 de setembro de 2002**. Divulga o detalhamento das naturezas de despesas 339030, 339036, 339039 e 449052. DOU de 17.9.2002.

BRASIL. **Portaria Conjunta STN/SOF nº 3**, de 15 de outubro de 2008. Aprova os Manuais de Receita Nacional e de Despesa Nacional e dá outras providências. D.O.U. de 16 de outubro de 2008.

CRISP 1.0 - Process and User Guide. **CROSS INDUSTRY STANDARD PROCESS FOR DATA MINING**. (Documentação do sítio oficial). Disponível em <http://www.crisp-dm.org/Process/index.htm >. Acesso em 18/11/2009.

DÖRRE, J., et al. **Text Mining**: Finding Nuggets in Mountains of Textual Data In Fifth International Conference on Knowledge Discovery and Data Mining (KDD-99), pp. 398-401.

FAYYAD, U; et.al. **Advances and Knowledge Discovery and Data Mining**. Cambridge: MIT. 1996. 611 p.

FELDMAN, Ronen; SANGER, James.**The Text Mining Handbook** - Advanced Approaches in Analyzing Unstructured Data. Cambridge University, USA, 2007. 410 p.

GEAN, C. C. e Kaestner, C. A. A. . "**Classificação Automática de Documentos usando Subespaços Aleatórios e Conjuntos de Classificadores**". In: TIL 2004 - 2º WORKSHOP EM TECNOLOGIA DA INFORMAÇÃO E DA LINGUAGEM HUMANA, Salvador.Anais do SBC 2004 v.1, p.1-8.

HILLOL Cargupta; JIAWEY Han; PHILIP S. Yu; RAJEEV Motwani; VIPIN Kumar. **Next Generation of Data Mining**. Chapman and Hall/CRC, 1ª. Edição. USA. 2008, 601 p.

IAN H. Witten; FRANK, Eibe. **Data Mining - Practical Machine Learning Tools and Techniques**. 2nd. ed. USA: Elservier, 2005, 558 p.

JACKSON, PETER; MOULINIER, ISABELLE. **Natural Language Processing for Online Aplication, Text Retrieval, Extraction and Categorization**. 2007. USA. 241 p.

KAO, A.; POTEET, S. . **Natural Language Processing and Text Mining**. USA. 2007. 265 p. .

Manuais Online do SQL Server 2008 (Documentação do sítio oficial publicado em julho de 2009). **Pesquisa de texto completo (SQL Server)**. Disponível em http://msdn.microsoft.com/pt-br/library/ms142571.aspx. Acesso em 28/04/2010.

MITCHEL, Tom M. **Machine Learning**. USA : McGraw-Hill 1997, 432 p.

POSTGRESQL. **Postgresql**. (Documentação do sítio oficial). Disponível em http://www.postgresql.org/ . Acesso em 05/07/2009.

RAJMAN, M.; BESANÇON, R. Text Mining: **Natural Language techniques and Text Mining applications.** Chapman & Hall, 1997.

SALTON, G., Wong, A., e Yang, C. S.. "**A Vector Space Model for Automatic Indexing**". in Readings in Information Retrieval, K.Sparck Jones and P.Willet, eds.,Morgan Kaufmann Publishers, Inc. 1997. San Francisco.

SCHIESSL, José Marcelo. **Descoberta de conhecimento em texto aplicada a um sistema de atendimento ao consumidor**. 2007, 106 p. Dissertação (Mestrado em Ciência da Informação) - Faculdade de Economia, Administração, Contabilidade e Ciência da Informação e Documentação (FACE), Universidade de Brasília, Brasília – DF.

TAN, A.-H. **Text Mining: The state of the art and the challenges**. In Proceedings of the Pacific Asia Conf on Knowledge Discovery and Data Mining PAKDD'99 workshop on Knowledge Discovery from Advanced Databases, p. 65–70, 1999.

WIVES, L. K.; LOH, S. **Tecnologias de descoberta de conhecimento em informações textuais** (ênfase em agrupamento de informações). In: OFICINA DE INTELIGÊNCIA ARTIFICIAL (OIA), III, 1999, Tutorial, Pelotas, RS. Proceedings... Pelotas: EDUCAT, 1999. p. 28-48.

WEISS, Sholom; INDURKHYA, Nitin; ZHAN ,Tong; DAMERAU , Frd J.. **Text Mining: Predictive Methods for Analizing Unstructured Information**. USA. 2005.

ZADROZNY, Bianca. **Aprendizado Automático**. Disponível em http://www.ic.uff.br/~bianca/topicos/. Acesso em 31/08/2009 .