UNIVERSIDADE FEEVALE

GUILHERME COIMBRA MOSTARDEIRO

DESCOBERTA DE CONHECIMENTO EM UMA BASE DE DADOS CRIADA A PARTIR DE KEYLOGGERS

Novo Hamburgo

2010

GUILHERME COIMBRA MOSTARDEIRO

DESCOBERTA DE CONHECIMENTO EM UMA BASE DE DADOS CRIADA A PARTIR DE KEYLOGGERS

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do grau de Bacharel em
Sistemas de Informação pela
Universidade Feevale

Professor orientador: Ricardo Ferreira de Oliveira

Novo Hamburgo

2010

RESUMO

As técnicas de mineração de dados têm como objetivo principal a extração de informações úteis a partir de grandes quantidades de dados. O uso deste tipo de tecnologia surgiu justamente para assumir a responsabilidade de exame e interpretação de dados onde estas tarefas não são possíveis de serem executadas manualmente, ou seja, sem o uso de computação. Portanto torna-se imprescindível o desenvolvimento de ferramentas que auxiliem o homem, de forma automática e inteligente, na tarefa de analisar, interpretar e relacionar esses dados para que se possa desenvolver e selecionar estratégias de ação em cada contexto de aplicação. Sendo assim, este trabalho propõe o desenvolvimento de uma ferramenta que irá atuar na tarefa de mineração de dados, onde estes foram capturados anteriormente pela utilização de *keyloggers*. As informações capturadas por *keyloggers* são muito valiosas em relação a tarefa de medição da produtividade dos usuários. Com elas poderemos saber que informações o usuário inseriu no computador tiveram relação com o negócio da empresa, assim como as informações que não têm nenhuma relação com o ambiente empresarial, e acabam afetando a produtividade do mesmo. Também poderão ser extraídos dados estatísticos e feitas associações como, por exemplo, que tipo de informações são inseridas para cada tipo de aplicação que o usuário está utilizando, ou até mesmo associar por título de janela do sistema operacional.

Palavras-chave: Descoberta de conhecimento. Reconhecimento de padrões. Inteligência artificial. *Keyloggers*.

ABSTRACT

The Data Mining techniques have as main objective the extraction of useful information from large amount of data. The use of such technology came justly to take the responsibility for examination and interpretation of data where these tasks are not possible to be performed manually, in other words, without the use of computer. Therefore it is essential to develop tools that assist humans, automatically and intelligently, the task to analyze, interpret and relate these data so that we can develop and select strategies for action on each application context. Therefore, this paper proposes the development of a tool that will perform the task of data mining, where they were previously captured by the use of keyloggers. The information captured by keyloggers are very valuable for the task of measuring the productivity of users. With them we know what information the User entered in the computer were related to the company's business, as well as information that have no relation to the business environment, and end up affecting the productivity of it. They can also be extracted and made statistical associations, for example, what kind of information is entered for each type of application that the User is using, or even join in the window title of the operating system.

Keywords: Knowledge discovery. Pattern recognition. Artificial intelligence. *Keyloggers*.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE GRÁFICOS

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmo Genético
DIRC	Diretoria de Combate ao Crime Organizado
DM	Data Mining
ID3	Idemized Dichotomizer 3
IE	Information Extraction
IR	Information Retrieval
KDD	Knowledge Discovery in Databases
KDT	Knowledge Discovery in Text
PLN	Processamento de Linguagem Natural
RNA	Rede Neural Artificial
RO	Registro de Ocorrência
TM	Text Mining
WEKA	Waikato Evironment for Knowledge Analysis
ASCII	American Standard Code for Information Interchange

SUMÁRIO

INTRODUÇÃO

Atualmente, o mundo está cada vez mais repleto de informações, e este fato tende a aumentar sem previsão de que esta tendência se inverta. Os computadores tornaram tão fácil a tarefa de guardar dados que as pessoas acabam guardando dados que provavelmente já teriam sido apagados não fosse a facilidade de armazenamento proporcionada pela tecnologia. A facilidade está na obtenção de mais "espaço" para guardar coisas, assim informações não precisam mais ser descartadas. Dentre as todas estas informações que poderiam ser descartadas, podem haver muitas informações importantes e potenciais "escondidas" de forma que se forem analisadas corretamente podem ser de muito valor (FRANK, WITTEN. 2009).

No início, os computadores serviam para nos livrar das tarefas de rotina: absorviam dados, faziam contas e imprimiam relatórios. Alguns desses dados precisavam ser constantemente acessados para consulta e atualização por diferentes tipos de usuários, sendo muitas vezes, compartilhados por mais de um sistema automatizado de informação, sendo assim, depois de alguns anos, passou a ser amplamente utilizada a tecnologia de banco de dados. E a partilha de dados foi se intensificando progressivamente com as ligações através de redes (GOLDSMITH, PASSOS. 2003).

Durante todo esse tempo, um dos sonhos de todo especialista em sistemas era o de ver as pessoas se tornarem mais racionais na tomada de decisões, pelo menos a partir do dia em que pudessem contar com um acesso rápido e fácil a esses vastos repositórios de dados. No entanto, examinar grandes volumes de dados brutos requer esforço, interpreta-los corretamente pode exigir talento e um nível de capacidade técnica não trivial, até mesmo para os especialistas (GOLDSMITH, PASSOS. 2003).

A mineração de dados (também conhecida pelo termo inglês *Data Mining* - DM), é o processo de explorar grandes quantidades de dados à procura de padrões consistentes, como regras de associação ou seqüências temporais, para detectar relacionamentos sistemáticos entre variáveis, transformando os dados processados em conhecimento útil.

Segundo Jerônimo, o DM vai muito além da simples consulta a um banco de dados, pois possibilita aos usuários a captura de informações úteis a partir dos dados, tentando descobrir conhecimentos escondidos.

A tecnologia de mineração de dados é motivada pela necessidade de uso de novas técnicas que auxiliem na análise, entendimento e visualização de muitas informações armazenadas em bancos de dados capturadas de aplicações de tipos variados. É impulsionada pela descoberta de padrões e hipóteses, que são automaticamente extraídos da base de dados, para posteriormente serem utilizadas em novos dados na tentativa de detectar a mesma situação ou para auxiliar em decisões (KIM, 2004).

Basicamente, o tratamento que a mineração de dados aplica sobre os dados não tem nada de novo: as pessoas buscam por padrões em dados deste o início de sua existência. Caçadores buscam por padrões no comportamento de migração de animais, políticos buscam por padrões em opiniões de eleitores, fazendeiros buscam por padrões relacionados ao aumento de safra, e a partir do momento em que padrões são encontrados, estes podem ser utilizados futuramente em novas situações, ou seja, se baseando na experiência para identificar oportunidades (FRANK, WITTEN. 2009).

A descoberta de conhecimento em uma base de dados pela mineração de dados ultrapassa os limites de percepção de determinado usuário sobre a mesma. Seria praticamente impossível a classificação/extração manual de informações dentre milhões de registros existentes, ao contrário do que se pensava antigamente, simplesmente melhorando o acesso a informação.

Diante disto, surgiu a possibilidade de automatização do processo de descoberta de conhecimento, utilizando-se de técnicas de mineração de dados, que realmente vieram em

socorro, e acabaram redirecionando a responsabilidade do exame e interpretação dos dados ao computador.

A descoberta de conhecimento em bases de dados (*Knowledge Discovery in Databases* – KDD) vem despertando grande interesse junto às comunidades científica e industrial. O termo KDD foi formalizado em 1989 em referência ao amplo conceito de procurar conhecimento a partir de bases de dados (GOLDSMITH, PASSOS. 2003).

A análise de grandes quantidades de dados pelo homem é inviável sem o auxílio de ferramentas computacionais apropriadas. Portanto torna-se imprescindível o desenvolvimento de ferramentas que auxiliem o homem, de forma automática e inteligente, na tarefa de analisar, interpretar e relacionar esses dados para que se possa desenvolver e selecionar estratégias de ação em cada contexto de aplicação (GOLDSMITH, PASSOS. 2003).

O foco do presente trabalho será a aplicação de descoberta de conhecimento em uma base de dados que foi gerada a partir do uso de *keyloggers*, tipo de programa que captura todo conteúdo digitado em determinado(s) computador(s), de determinada rede, independente de seu uso principal. Analisando o tema do trabalho proposto, pode-se perceber o quão complicado seria caso fosse necessária uma classificação dos dados e extração/descoberta de conhecimento manualmente nesta suposta base de dados. Por este motivo é que se faz necessária a utilização de técnicas de descoberta de conhecimento.

O desenvolvimento deste trabalho poderá ser aplicado na prática em ambientes organizacionais, como empresas, para verificar que tipo de informações são inseridas nos computadores por parte dos usuários, e em cima disto, poder-se-á fazer uma análise utilizando-se das técnicas de mineração de dados propostas, para que todas as informações possam ser classificadas sem que seja necessária uma interação por parte de algum usuário, haja visto que seria necessário muito tempo para a execução desta tarefa.

Diante disto, a proposta deste trabalho é, primeiramente, identificar, através de pesquisas, a melhor maneira, utilizando-se das técnicas de mineração de dados, para que seja possível o tratamento das informações adquiridas. Baseado nestas pesquisas e nos resultados obtidos, será proposta a modelagem e desenvolvimento de uma ferramenta capaz de executar

a tarefa de mineração de dados, assim como a modelagem da ferramenta responsável pela captação dos dados que serão tratados.

Sendo assim, no capítulo 1 e 2 serão apresentadas as duas formas de extração de conhecimento a partir de dados utilizadas neste trabalho: a descoberta de conhecimento em base de dados (*Knowledge Discovery in Databases* – KDD) e a descoberta de conhecimento a partir de texto (*Knowledge Discovery in Text* – KDT), sucessivamente, haja vista a necessidade de utilização de técnicas de extração de conhecimento destes dois processos. No capítulo 3 serão apresentados tipos algoritmos de classificação, justamente por seu uso ser necessário na aplicação que está sendo proposta no trabalho. No capítulo 4 são apresentadas aplicações reais que fazem uso de descoberta de conhecimento em dados e/ou descoberta de conhecimento em texto.

1 KNOWLEDGE DISCOVERY DATABASE

Diante de um novo contexto, onde a análise de grandes quantidades de dados pelo homem é inviável sem o auxílio de ferramentas computacionais apropriadas, surgiu o termo KDD, visando atuar na tarefa de analise, interpretação e relacionamento destes dados com o objetivo de desenvolver e selecionar estratégias de ação em cada contexto de aplicação.

Em 1989 o amplo conceito de procurar conhecimento a partir de bases de dados foi formalizado como *Knowledge Discovery in Databases* - KDD. Segundo Fayyad, autor de uma das definições mais populares para o termo proposta em 1996: "KDD é um processo, de várias etapas, não trivial, interativo e iterativo, para identificação de padrões compreensíveis, válidos, novos e potencialmente úteis a partir de grandes conjuntos de dados".

Segundo Goldschmidt e Passos, a participação do usuário é de grande importância na condução de processos do KDD. Cabe ao analista humano a árdua tarefa de orientar a execução do processo, assim como, a interpretação dos resultados buscando identificar qual a estratégia a ser adotada em relação aos mesmos em relação contexto da aplicação.

Segundo Braga (2005), é preciso estar consciente de que o processo de descoberta não se faz provocando uma hipótese, mas, colhendo evidências e explicações sobre a mesma podendo eventualmente levar à construção de um modelo.

Segundo HESS e WAGNED (2001), a mineração de dados vai além de uma simples consulta no banco de dados, pois não é restrita aos dados que estão relacionados. Ela utiliza algoritmos que comparam dados sem relação aparente, obtendo informações antes desconhecidas.

Existem dois fatores que influenciam diretamente o sucesso no *Data Mining*. O primeiro é a definição precisa do problema que será levado em consideração para se encontrar a solução. O segundo é a utilização dos dados certos no processo (TWO CROWS CORPORATION. 1999).

Segundo Cabena (1998), a definição clara do problema ou desafio relacionado ao negócio é um ingrediente essencial em qualquer projeto de KDD. Embora pareça simples e intuitivo, não é. Frequentemente, mesmo sem ter pensado nos objetivos de sua aplicação, pessoas pensam que simplesmente possuindo as informações é possível executar a descoberta de conhecimento a partir das mesmas. Acabam esquecendo de que a mineração irá buscar encontrar uma solução a partir das informações para um determinado problema.

Este tipo de descoberta de conhecimento é caracterizado como um processo composto por três etapas operacionais básicas: Pré-processamento, Mineração de dados e Pós-processamento. A seguir é apresentada uma imagem que demonstra o processo de KDD como um todo e o detalhamento de cada etapa do mesmo.

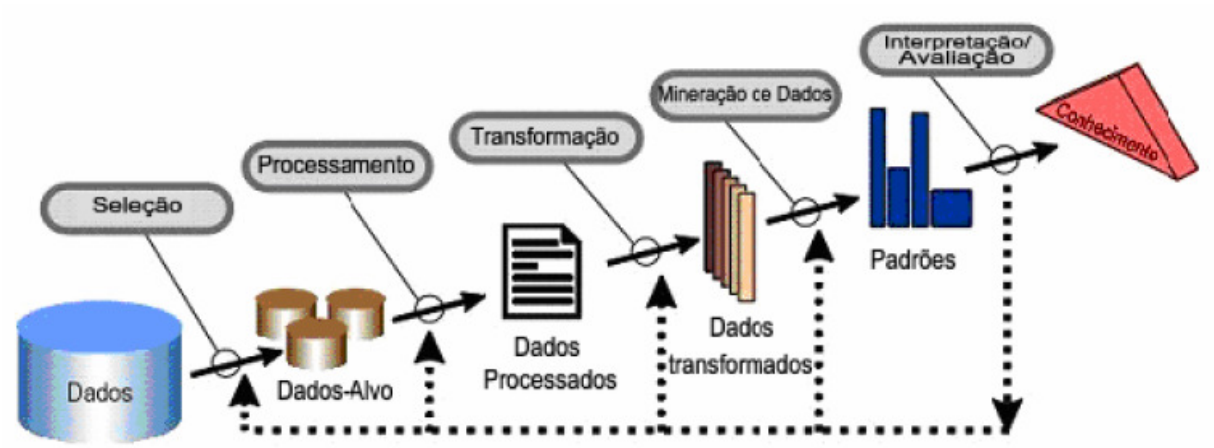


Figura 1.1 – Visão geral do processo de KDD.
Fonte: SANTOS, 2008, p. 15.

Sendo assim, as etapas de seleção, processamento e transformação são apresentadas no Tópico Pré-processamento. É importante lembrar que a etapa denominada Processamento da Figura 1.1 está, na realidade, relacionada a um processamento necessário dentro do contexto

do Pré-processamento. E a etapa denominada Interpretação/Avaliação será apresentada no Tópico Pós-processamento.

1.1 Pré-processamento

No mundo real, facilmente defronta-se com casos em que dados que desejam ser processados por técnicas de *Data Mining* possuem algumas características como: inconsistência, inexistência e incompletude. Existem muitas possíveis razões para que dados com estas características tenham sido gerados, tanto erros humanos como computacionais. Contudo, a tarefa de Pré-processamento não visa a descoberta dos motivos que levarão a geração de dados com este perfil, e sim a execução de tarefas que eliminem ou corrijam tais dados para que os mesmos possam passar para etapa seguinte (FRANK, WITTEN. 2009).

Esta etapa, que compreende as funções relacionadas à captação, à organização e ao tratamento dos dados, tem foco na preparação dos dados para que estes possam ser utilizados na etapa de Mineração de dados. Possui fundamental relevância no processo de descoberta de conhecimento. Compreende desde a correção de dados errados até o ajuste da formatação dos dados para os algoritmos de Mineração de Dados a serem utilizados (GOLDSCHMIDT, PASSOS. 2003). Segundo Cabena (1997), o pré-processamento é a etapa que mais consome recursos em seu processo, geralmente exigindo até 60% do esforço de todo o projeto.

As funções existentes nesta etapa que serão detalhadas a seguir são: seleção, limpeza e transformação dos dados.

1.1.1 Seleção de dados

Responsável por identificar quais das informações existentes devem ser consideradas durante o processo de KDD. É muito comum encontrarmos em uma base de dados informações que serão descartadas por esta função. Por exemplo, o nome do cliente é uma informação totalmente irrelevante em uma aplicação de KDD cujo objetivo seja construir um modelo que preveja o comportamento de novos clientes quanto ao pagamento de futuros créditos a eles concedidos. A seleção de dados pode ser desmembrada basicamente em dois

tipos: escolha de atributos, onde é feita somente a seleção de atributos que serão considerados no processo; e a escolha de registros, onde somente registros escolhidos passarão para próxima etapa (GOLDSCHMIDT, PASSOS. 2003).

1.1.2 Limpeza

Como o próprio nome sugere, nesta tarefa será realizado o tratamento sobre os dados selecionados visando assegurar a qualidade dos mesmos. Envolve uma verificação da consistência das informações, a correção de possíveis erros e o preenchimento ou a eliminação de valores desconhecidos e redundantes, além da eliminação de valores não pertencentes ao domínio. Em aplicações reais, é comum que os dados sobre os quais se deseja extrair algum conhecimento estejam incompletos, ruidosos ou inconsistentes. (GOLDSCHMIDT, PASSOS. 2003).

Segundo Frank e Witten (2009), os dados encontrados no mundo real tendem a ser incompletos, “embaraçados”, e inconsistentes. A função da Limpeza de dados visa preencher valores não presentes, atenuar os dados ruins e corrigir as inconsistências presentes nos dados.

1.1.3 Transformação

A transformação dos dados é a fase que antecede a fase de mineração dos dados. Após serem selecionados, limpos e pré-processados, os dados necessitam ser armazenados e formatadas adequadamente para que os algoritmos possam ser aplicados.

Objetivo desta fase é transformar ou consolidar os dados em uma forma apropriada que seja compatível com o processo de mineração dos dados (FRANK, WITTEN. 2009). Segundo Silva (2008), nesta fase os dados pré-processados são transformados para que se possa criar um modelo de dados analítico. A acurácia e validade do resultado final dependerão de como o analista de dados decidiu estruturar e apresentar as entradas que servirão de base para a escolha de um algoritmo específico de DM.

1.2 Mineração de Dados

Alguns autores se referem a esta etapa como sendo a mais importante do processo de KDD, tendo em vista que nela é realizada a busca efetiva por conhecimentos úteis dentre os dados presentes (GOLDSCHMIDT, PASSOS. 2003).

Segundo Han e Kamber (2001), muitas pessoas se referenciam ao *Data Mining* como sendo um sinônimo para o termo KDD, porém, na realidade, o processo de DM é simplesmente uma única etapa do processo do KDD. *Data Mining* é um campo interdisciplinar, com influência de uma série de disciplinas, incluindo-se Sistemas de Base de Dados, Estatística, Inteligência artificial, Visualização e Ciência da Informação, como pode ser visto da Figura 1.2. Dependendo do tipo de aplicação em que o DM for aplicado, outras técnicas poderão ser aplicadas.

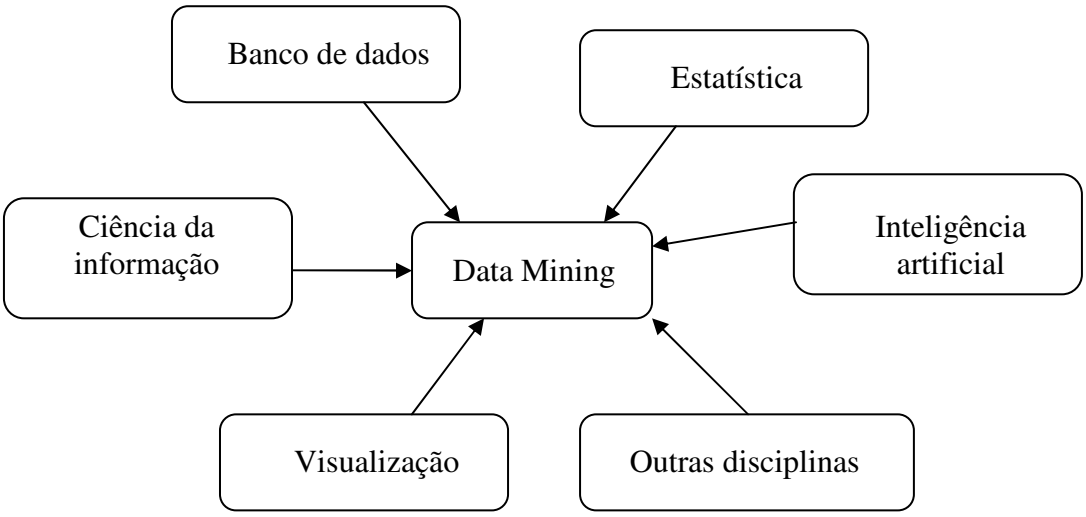


Figura 1.2 – Disciplinas que influenciam o Data Mining.
Fonte: HAN, KAMBER, 2001, p. 15.

Na realidade, esta etapa do KDD é quase inseparável da próxima etapa, denominada Pós-processamento, onde é feita a análise dos resultados. Isto porque as duas etapas estão interligadas no processo, pois durante o mesmo o analista geralmente interage com as duas

etapas ao mesmo tempo, podendo necessitar de alterações na etapa de mineração para que a mesma esteja de acordo com os objetivos da aplicação (HAN, KAMBER. 2001).

Esta etapa possui diversas técnicas, e a escolha da mesma dependerá do tipo de tarefa de KDD a ser realizada. Sendo assim, a seguir são apresentadas as principais tarefas da etapa de Mineração de Dados.

1.2.1 Tarefas de Mineração de Dados

1.2.1.1 Descoberta de Associação

Esta técnica baseia-se na busca por itens que freqüentemente ocorram de forma simultânea em transações do banco de dados. Um exemplo clássico desta técnica está relacionado com a área de Marketing: durante um processo de descoberta de associações em sua vasta base de dados, uma grande rede de mercados norte-americana descobriu que um número razoável de compradores de fralda também comprava cerveja na véspera de finais de semana com jogos transmitidos pela televisão. Tais compradores eram, na realidade, homens que, ao comprarem fraldas para seus filhos, compravam também cerveja para consumo enquanto cuidavam das crianças e assistiam aos jogos na televisão durante o final de semana. Este exemplo ilustra a associação entre fraldas e cervejas (GOLDSCHMIDT, PASSOS. 2003).

1.2.1.2 Classificação

Uma das tarefas mais importantes e populares é a tarefa de classificação. Consiste em descobrir uma função que mapeia um conjunto de registros em um conjunto de rótulos categóricos predefinidos, denominados classes. Um exemplo de seu uso é a classificação de clientes, baseando-se em um histórico com dados de clientes e o comportamento dos mesmos em relação ao pagamento de empréstimos contraídos previamente. Pode-se considerar dois tipos de clientes: clientes que pagaram e clientes inadimplentes. Estas são as classes do problema. A tarefa de classificação consiste em descobrir uma função que mapeie corretamente os clientes, a partir de seus dados, em uma destas classes.

Segundo Two Crows Corporation (1999), problemas de classificação visam identificar características que indiquem o grupo que cada caso pertence. Este tipo de mineração pode ser utilizado tanto para entender dados existentes quanto para prever o grupo de novos casos.

Na classificação, a forma de aprendizado baseia-se em uma série de classes pré-determinadas a partir de exemplos, assim espera-se que os algoritmos consigam aprender uma maneira de classificar novos dados, mesmo que não estejam nas classes pré-definidas de exemplo, mas tentar encontrar uma relação e conseguir associar um dado à alguma classe (WITTEN, FRANK. 2000).

Segundo HESS e WAGNER, a técnica de classificação consiste em analisar diversos registros, cada um com um conjunto de atributos, e descobrir um modelo que classifique (agrupe) estes registros em função dos seus valores para os atributos.

A seguir apresenta-se um exemplo simples da execução da tarefa de classificação. Este exemplo está relacionado com a compra de livros em alguns países levando em consideração a idade e o sexo da pessoa. E em seguida é apresentado o resultado adquirido utilizando-se do algortimo C4.5, que será apresentado no Capítulo III.

Sexo	País	Idade	Compra
M	França	25	Sim
M	Inglaterra	21	Sim
F	França	23	Sim
F	Inglaterra	34	Sim
F	França	30	Não
M	Alemanha	21	Não
M	Alemanha	20	Não
F	Alemanha	18	Não
F	França	34	Não
M	França	55	Não

Tabela 1.1 – Clientes e suas compras de livros.

Fonte: Adaptado de GOLDSMITH e PASSOS, 2003, p. 70.

Se País = Alemanha Então Compra = Não
Se País = Inglaterra Então Compra = Sim
Se País = França e Idade <= 25 Então Compra = Sim
Se País = França e Idade > 25 Então Compra = Não

Figura 1.3 – Conhecimento extraído através do algoritmo C4.5.
Fonte: Adaptado de GOLDSMITH e PASSOS, 2003, p. 71.

Provavelmente está tarefa é a mais comum dentro do escopo de análise de dados textuais. Tem como objetivo classificar, ou categorizar dados em uma série de classes ou categorias. Aplicada no gerenciamento de documentos, a tarefa é a classificação de textos, dada uma série de classes possíveis e uma coleção de documentos, deste modo o processo visa encontrar a classe correta em que cada documento se "encaixa" (FELDMAN, SANGER. 2006). Segundo Barion e Lago (2008), a classificação de dados textuais é um processo que visa a identificação de tópicos principais em um documento e a sua associação baseando-se em um algoritmo pré-definido, construído a partir de um conjunto de treinamento definido por pessoas experientes no assunto envolvido.

1.2.1.3 Clusterização

Esta tarefa visa analisar os dados e classifica-los de acordo com suas características, porém, ao contrário da tarefa de classificação, não são determinadas previamente classes em que os dados deverão ser "encaixados". Ou seja, na própria execução da tarefa os rótulos que identificam os tipos de dados serão gerados. A base de dados é dividida em *clusters*, cada um destes é formado pelos dados mais similares encontrados, e possuem dados bem diferenciados em relação aos demais *clusters* (HAN, KAMBER. 2001).

Segundo Goldsmith e Passos (2003), esta tarefa também é chamada de agrupamento, justamente pelo fato de agrupar os dados semelhantes. O objetivo desta tarefa é maximizar similaridade intracluster e minimizar a similaridade intercluster. A Clusterização é denominada indução não supervisionada, pelo fato de identificar automaticamente os rótulos ou classes dos dados.

1.2.1.4 Regressão

Esta tarefa compreende, fundamentalmente, a busca por funções, lineares ou não, que mapeiem os registros de um banco de dados em valores reais. É muito similar à tarefa de classificação, porém restrita apenas a atributos numéricos (GOLDSMITH, PASSOS. 2003).

Segundo Amaral (2001), a regressão é a função de aprendizado que mapeia os dados com pré-elaboração variável de valores reais. A regressão linear é a forma mais simples de regressão, onde a função a ser abstraída a partir dos dados é uma função linear. A seguir apresentar-se-á um exemplo de regressão linear simples, relacionada a concessão de empréstimos, que teve como objetivo identificar a função que relaciona-se o salário e os débitos de usuários.

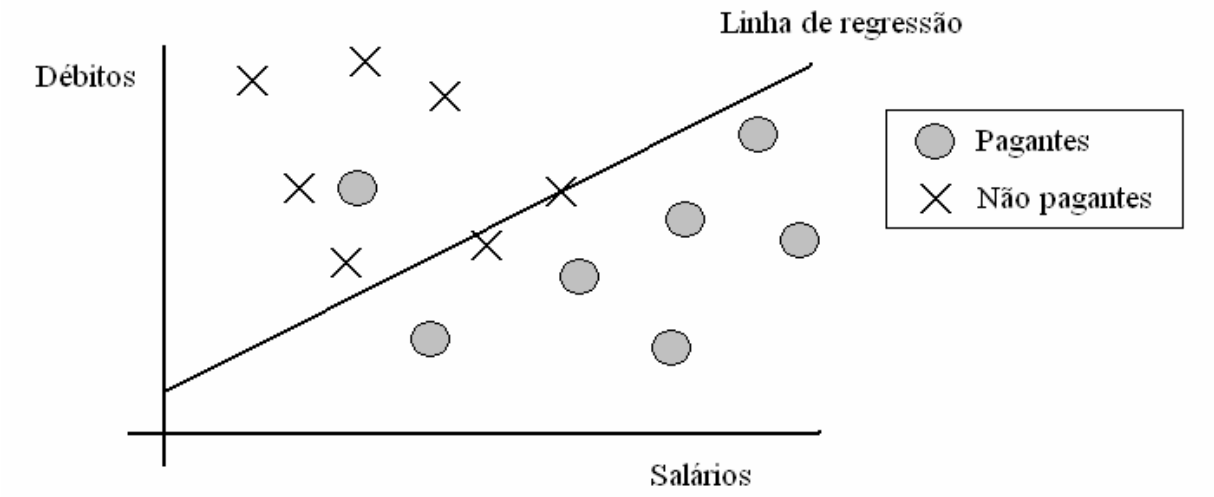


Figura 1.4 – Regressão para conjunto de dados de empréstimos.
Fonte: Adaptado de AMARAL, 2001, p. 25.

Pode-se perceber que a função não conseguiu atingir o resultado totalmente esperado, isto ocorre pois existe uma correlação fraca entre débitos e salários. Ou seja, nem sempre as pessoas que ganham um salário acima da média pagam seus débitos, assim como, nem sempre as pessoas abaixo da média salarial deixam de pagar seus débitos.

Segundo Two Crows Corporation (1999), a regressão utiliza dados existentes para prever o que novos dados representam. Infelizmente muitos problemas do mundo real não são simples projeções lineares de valores prévios, assim como o exemplo que foi apresentado anteriormente.

1.2.1.5 Sumarização

A tarefa de sumarização consiste em identificar e apresentar, de forma concisa e compreensível, as principais características dos dados contidos em um conjunto de dados. Não é uma simples enumeração dos dados. A sumarização dos dados busca gerar descrições para caracterização resumida dos dados e possivelmente comparação (discriminação) entre eles. É também conhecida por descrição de conceitos, visto que um conceito normalmente se refere a uma coleção de dados com pelo menos uma característica em comum (GOLDSCHMIDT, PASSOS. 2003).

No contexto de mineração de texto, a sumarização tem por objetivo produzir uma lista de sentenças mais importantes do documento origem resumindo o conteúdo do mesmo, reduzindo seu volume, porém mantendo a mesma informação inicial. É uma técnica bastante utilizada em mineração de textos com o intuito de identificar palavras ou frases mais importantes dos documentos (BARION, LAGO. 2005). A sumarização compreende dois processos: a seleção do conteúdo relevante de uma mensagem e sua organização coerente. Em ambos, três características principais devem ser observadas: a satisfação do objetivo comunicativo do discurso, a veiculação de sua proposição central e o inter-relacionamento coerente das unidades de conteúdo necessárias para satisfazer as restrições anteriores (ARANHA, PASSOS. 2006).

1.2.1.6 Descoberta de Seqüências

Baseia-se na descoberta de conhecimento em relação ao aspecto temporal entre os registros no banco de dados. Segundo Goldschmidt, os padrões são denominados intertransação, pois diversas transações devem ser analisadas em ordem cronológica de ocorrência.

Muitas vezes referenciada como análise de evolução, justamente por levar em consideração a evolução dos dados durante o tempo, esta técnica visa descobrir regularidades ou tendências para conjuntos de dados que apresentaram mudanças em seu comportamento em relação ao tempo (HAN, KAMBER. 2001).

1.3 Pós-processamento

Essa fase tem como objetivo utilizar o conhecimento gerado pela etapa de Mineração de Dados para desenvolvimento da visualização, análise e a interpretação do mesmo. Segundo Goldschmidt, é nesta etapa que o especialista em KDD e o especialista no domínio da aplicação avaliam os resultados obtidos e definem novas alternativas de investigação dos dados.

Segundo Cabena (1998), a análise de resultados da mineração executada previamente é uma das etapas mais importantes do processo como um todo. Porém antes do conhecimento adquirido ser destinado ao desenvolvimento de sua visualização, citada anteriormente, isto só será feito após o conhecimento ser devidamente verificado pelo analista de dados trabalhando juntamente com o analista de negócios. Ocasionalmente, o executivo responsável poderá ser convocado para confirmação das conclusões obtidas em relação aos objetivos do negócio.

Nesta fase existem duas operações muito importantes, a Simplificação do Modelo de Conhecimento, que vista remover detalhes do modelo de conhecimento gerado de forma a torná-lo menos complexo, porém sem perda de informação relevante; e a Transformação de Modelo de Conhecimento, que abrange alguns métodos utilizados com o objetivo de facilitar a análise de modelos de conhecimento, estes métodos consistem basicamente na conversão da forma de representação do mesmo modelo (GOLDSCHMIDT, PASSOS. 2003).

A especificação das atividades nesta etapa pode variar dependendo do tipo de aplicação que está sendo desenvolvida. Todavia, uma pergunta que permanece no pensamento de todas as pessoas envolvidas no processo de KDD é a mesma: “Encontramos alguma coisa que seja interessante, válida, de algum proveito?” Geralmente a resposta é não, neste caso o processo de *Data Mining* deve ser repetido (CABENA. 1998).

A seguir é apresentado o Gráfico 1.6, que ilustra o esforço requerido para cada etapa dentro do processo do KDD. Segundo Cabena (1998), nem todas as etapas do processo têm pesos iguais em termos de tempo de execução e esforço utilizado. Como pode ser visualizado no gráfico, 50% do tempo é direcionado para a preparação dos dados para a mineração, ressaltando assim importância desta etapa para o sucesso processo do KDD. A etapa de mineração dos dados utiliza geralmente cerca de 10% do esforço total.

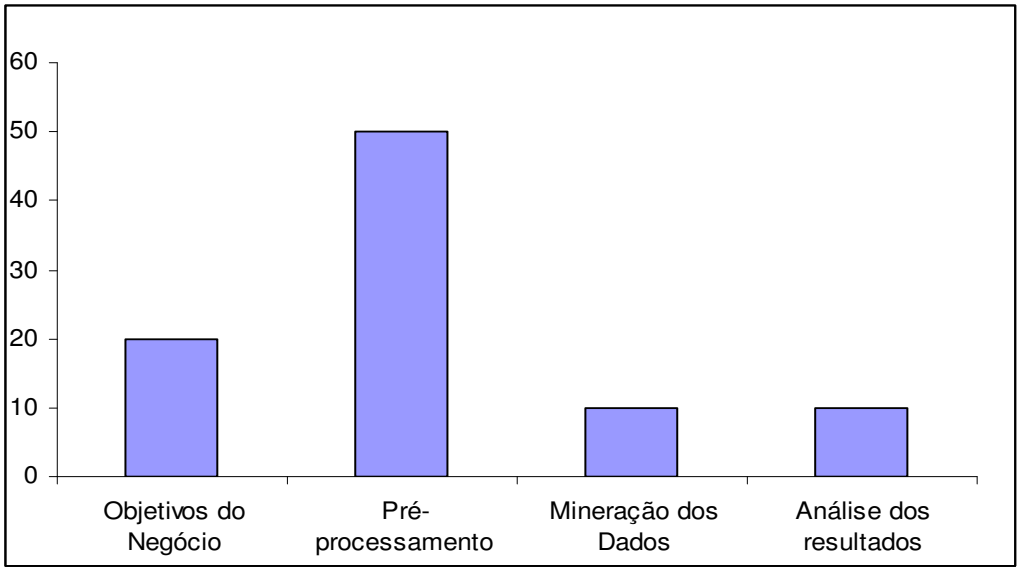


Gráfico 1.1 – Esforço requerido para cada etapa do processo do KDD.
Fonte: Adaptado de CABENA, 1998, p. 42.

2 KNOWLEDGE DISCOVERY IN TEXT

Data Mining está relacionada com a procura de padrões em dados. Da mesma forma, o processo do KDT, também conhecido como *Text Mining,* está relacionado com a procura de padrões em texto. Comparada com a mineração de dados, esta técnica apresenta algumas dificuldades como: dados não estruturados e amorfos. A descoberta de conhecimento a partir de textos é possível porque não é necessário entender o sentido de um texto para que se possa extrair informações úteis do mesmo (WITTEN, FRANK. 2000).

Segundo Silva (2002), a KDT pode ser considerada um processo de KDD para dados não estruturados como, por exemplo, aqueles encontrados na Internet ou ainda em organizações que ultimamente, pela facilidade e barateamento de custos, vêm armazenando quantidades crescentes de textos em meios magnéticos.

As aplicações típicas de *Data Mining* utilizam informações estruturadas que provém de um preparo feito cuidadosamente. Os dados devem ser transformados pela etapa de Pré-processamento, e além disso, deve haver um cuidado muito grande na seleção de informações destinadas a etapa de Mineração. Assim como o *Data Mining,* o *Text Mining* também necessita de um tratamento muito especial antes do inicio de qualquer atividade relacionada com a descoberta de conhecimento, haja visto que os textos geralmente são uma coleção de dados não estruturados e sem tratamento para composição de documentos (WEISS. 2004).

O *Text Mining,* assim como o *Data Mining*, tem como objetivo principal a extração de informações úteis a partir de base de dados através da identificação e exploração de padrões interessantes para a aplicação. Claro que no caso do *Text Mining* a base de dados é, na realidade, uma coleção de documentos, ou uma coleção de textos, e padrões interessantes são

encontrados não entre registros mas em textos não estruturados (FELDMAN, SANGER. 2006).

O *Text Mining* deriva muito de suas características de pesquisas relacionadas com o *Data Mining*. Portanto, não é uma surpresa descobrir que estas duas técnicas apresentam funcionalidades similares. Por exemplo, os dois utilizam rotinas de pré-processamento, algoritmos de descoberta de padrões e o pós-processamento responsável pela visualização dos resultados (FELDMAN, SANGER. 2006).

A KDT combina técnicas de extração, recuperação de informação, processamento da linguagem natural e sumarização de documentos com os métodos de DM. Por lidar com dados não-estruturados, a mesma também é considerada mais complexa que o KDD (SILVA. 2002).

Text Mining pode ser definido como a aplicação de recursos e técnicas computacionais sobre dados textuais com o objetivo de encontrar informações intrínsecas e relevantes que anteriormente estavam escondidas no conjunto de dados e que por este motivo eram desconhecidas por seus usuários (PRADO, OLIVEIRA, FERNEDA, WIVES, SILVA, LOH. 2005).

Segundo Beppler e Fernandes (2005), cerca de 80% de informações contidas nas organizações estão representadas na forma de texto. Porém, o *Text Mining* é um processo muito mais complexo pois envolve procedimentos com dados textuais que estão em linguagem natural, de forma não estruturada.

Basicamente, o KDT segue a mesma linha de modelo de um sistema KDD, ou seja, tarefas de pré-processamento, mineração de dados, representação de resultados e reformulação de do processo como um todo caso necessários (para casos em que o resultado não foi satisfatório). A seguir apresentar-se-á um detalhamento sobre as etapas do KDT.

2.1 Pré-processamento

Esta etapa está relacionada com a limpeza dos dados para facilitar as análises na etapa seguinte, que é a Extração de conhecimento em si. Pode ser dividida em cinco grandes fases, que são: correção ortográfica, remoção de *Stopwords*, processo de *Stemming*, e seleção de termos relevantes. Sendo aplicadas nesta ordem, porém sem obrigatoriedade de execução. Em alguns casos de aplicação ainda existe uma fase denominada identificação de termos (GOMES, MONTEIRO, OLIVEIRA. 2006).

Na etapa de pré-processamento do *Text Mining,* as operações estão focadas na identificação e extração de representações de linguagem natural em documentos. Esta operação é responsável por transformar informações não estruturadas em coleções de documentos que estejam da forma mais estruturada possível, o que é uma preocupação não relevante para a maioria de sistemas de *Data Mining*. A etapa de Pré-processamento é "obrigatória" dentro do processo de *Text Mining*, pois geralmente os algoritmos de descoberta de conhecimento não são utilizados em coleções de textos que estejam despreparadas (FELDMAN, SANGER. 2006).

Após a aplicação dos processos que visam transformar os dados textuais, que são não estruturados ou semi-estruturados, em dados estruturados, as técnicas de mineração de dados, apresentadas anteriormente podem ser utilizadas sobre o texto processado (BARION, LAGO. 2008). Ou seja, algumas aplicações de *Text Mining* utilizam somente a etapa de Pré-processamento, assim, os dados textuais já estarão estruturados e técnicas de *Data Mining* poderão ser utilizadas.

2.1.1 Correção ortográfica

Como o próprio nome sugere, esta fase tem como objetivo a verificação de palavras em conformidade com a sua ortografia. Geralmente um verificador ortográfico utiliza um dicionário e compara cada palavra do texto com os termos do dicionário. Caso a palavra do texto em questão também existir no dicionário a mesma é considerada como ortograficamente correta e é aceita no texto (GOMES, MONTEIRO, OLIVEIRA. 2006).

Segundo Toniazzo (2005), os termos obtidos durante o processo de extração podem conter erros. Os erros podem ser causados pela má digitação, sendo assim, uma correção deve ocorrer para permitir que o processo de tratamento das informações pela aplicação funcione corretamente.

2.1.2 Remoção de *Stopwords*

Consiste na retirada de palavras que se repetem inúmeras vezes no decorrer do texto ou palavras com pouco significado, tais como artigos, preposições, e algumas conjunções, ou seja, palavras que não têm nenhuma relevância aparente para o entendimento do texto, estas são consideradas *Stopwords*. Geralmente estas palavras estão definidas em um arquivo, que contém a lista de todas as palavras consideradas *Stopwords* de acordo com o tipo de aplicação (GOMES, MONTEIRO, OLIVEIRA. 2006).

Segundo Silva (2002), pelo fato das *Stopwords* serem bastante freqüentes em qualquer texto, a eliminação das mesmas leva a uma redução entre 40% a 50% dos textos a serem analisados. A seguir é apresentada a Figura 2.1 que possui uma lista palavras que são geralmente consideradas *stopwords* em aplicações de *Text Mining*.

coisa	deste	ela	eu	lo	nestas	pelos	poucos	seu	toda
coisas	destes	elas	fazendo	mas	ninguém	pequena	primeiro	seus	todas
com	deve	ele	fazer	me	no	pequenas	primeiros	si	todavia
como	devem	eles	feita	mesma	nos	pequeno	própria	sido	todo
contra	devendo	em	feitas	mesmas	nós	pequenos	próprias	só	todos
contudo	dever	enquanto	feito	mesmo	nossa	per	próprio	sob	tu
da	deverá	entre	feitos	mesmos	nossas	perante	próprios	sobre	tua
daquele	deverão	era	foi	meu	nosso	pode	quais	sua	tuas
daqueles	deveria	essa	for	meus	nossos	pôde	qual	suas	tudo
das	deveriam	essas	foram	minha	num	podendo	quando	talvez	última
de	devia	esse	fosse	minhas	numa	poder	quanto	também	últimas
dela	deviam	esses	fossem	muita	nunca	poderia	quantos	tampouco	último
delas	disse	esta	grande	muitas	o	poderiam	que	te	últimos
dele	disso	está	grandes	muito	os	podia	quem	tem	um
deles	disto	estamos	há	muitos	ou	podiam	são	tendo	uma
depois	dito	estão	isso	na	outra	pois	se	tenha	umas
dessa	diz	estas	isto	não	outras	por	seja	ter	uns
dessas	dizem	estava	já	nas	outro	porém	sejam	teu	vendo
desse	do	estavam	la	nem	outros	porque	sem	teus	ver
desses	dos	estávamos	la	nenhum	para	posso	sempre	ti	vez
desta	e	este	lá	nessa	pela	pouca	sendo	tido	vindo
destas	é	estes	lhe	nessas	pelas	poucas	será	tinha	vir
deste	e'	estou	lhes	nesta	pelo	pouco	serão	tinham	vos

Figura 2.1 – Lista de stopwords freqüentes em aplicações de Text Mining.
Fonte: Adaptado de LOH, 2008.

2.1.3 Processo de Stemming

Existem muitas palavras dentro de um texto que podem ter variações morfológicas, como por exemplo: plural, formas de gerúndio e sufixos temporais. A identificação é efetuada através do radical de uma palavra. O processo de *Stemming* consiste na remoção destas variações, sendo o resultado obtido chamado de Stem (raiz) (GOMES, MONTEIRO, OLIVEIRA. 2006).

Segundo Silva (2002), a vantagem é que, em uma busca, o usuário não precisa se preocupar com a classe da palavra, podendo ela aparecer no texto como um substantivo, um verbo ou um adjetivo. No entanto, isto implica diminuição na precisão da pesquisa.

2.1.4 Seleção de termos relevantes

Da mesma forma que as *Stopwords* são consideradas as palavras que não tem um significado importante e pouca relevância para o texto, existem palavras que são o oposto. Estas têm grande importância no texto, como por exemplo: palavras encontradas em tópicos, como títulos e substantivos.

Segundo Silva (2002), deve-se considerar que em um texto as palavras têm níveis de destaque distintos. Aquelas mais freqüentes, excluindo-se as *Stopwords*, são mais importantes que outras que aparecem com menos freqüência.

Segundo Toniazzo (2005), existem três formas de verificar a importância de uma palavra em relação a sua freqüência no texto. Essas freqüências, na verdade, servem para identificar o peso do termo ou a força do termo. Estas formas estão apresentadas a seguir.

- **Freqüência Absoluta**: quantidade total de ocorrências de uma determinada palavra no documento ou conjunto analisado.
- **Freqüência Relativa**: freqüência de uma palavra em relação a todas as outras palavras do documento.
- **Freqüência Inversa de Documentos**: medida que leva em consideração tanto Freqüência Absoluta quando a Freqüência Relativa.

2.1.5 Identificação de Termos

Consiste na identificação de palavras importantes do texto, ignorando símbolos e caracteres de controle de arquivo ou formatação, pois existem seqüências de caracteres que muitas vezes podem ser substituídos por outras palavras ou termos compostos não podem ser tratados individualmente. Como exemplo, são apresentados os termos compostos: processo judicial, processo computacional. Neste caso esta fase se faz necessária para que não haja perda de significado (SILVA, 2002).

2.2 Mineração de texto

Assim como no KDD, esta etapa consiste na aplicação de um algoritmo de análise sobre os dados com o objetivo de identificar e extrair informações desconhecidas que possam ser úteis. Dentre os métodos existentes para a realização desta etapa, é importante se preocupar com uma característica fundamental: a relevância da informação (TONIAZZO, 2005).

Segundo Gomes, Monteiro e Oliveira (2006), os algoritmos e técnicas desenvolvidos para esta etapa podem ser divididos em duas categorias: a de geração de conhecimento, que utiliza técnicas para gerar conhecimento a partir de informações contidas em um determinado texto, e a de extração de conhecimento, que utiliza técnicas para retirar o conhecimento que esteja explícito no texto. A seguir apresentar-se-á alguns dos principais tipos de mineração em textos.

2.2.1 Recuperação da informação

Segundo Silva (2002), este tipo de mineração de texto (também conhecida pelo termo inglês *Information Retrieval* - IR), tem como objetivo localizar documentos que contenham informações relevantes para atender às necessidades definidas pelo usuário em uma consulta. Deste modo, o usuário necessita examinar os documentos resultantes dessa busca para encontrar a informação, o que é uma tarefa demorada.

Para diminuir o esforço de busca por informações utiliza-se a indexação, que provê uma busca mais rápida e eficiente. Esta indexação é considerada um tipo de filtro capaz de selecionar e identificar as características de um documento, extraindo os termos mais significativos e desconsiderando aqueles que não são importantes.

2.1.1.1 Indexação

Segundo Barion e Lago (2008), a indexação é o processo pelo qual as palavras contidas no texto são armazenadas em uma estrutura de índices para viabilizar a pesquisa de documentos através das palavras que o mesmo contém.

A indexação serve para se fazer uma rápida busca de documentos através de palavras-chave. Uma estrutura de dados de armazenamento inteligente proporciona aumento drástico de performance. Além de recuperar dados textuais, ela pode fazer cálculos com múltiplas palavras-chave de busca realizando uma ordenação segundo a avaliação de cada documento (ARANHA, PASSOS. 2006).

Para realização da indexação são realizados alguns processos para diminuição de termos no texto, visando o melhor desempenho na utilização dos índices gerados. Muitos destes processos correspondem à parte de pré-processamento, tais como, remoção de *stopwords*, correção ortográfica e identificação de termos mais importantes.

2.2.2 Extração de informação

Esta tarefa de mineração de texto (também conhecida pelo termo inglês *Information Extraction* - IE), tem como objetivo analisar dados não estruturados ou semi-estruturados, que seriam os textos propriamente ditos, e extrair informações a partir dos mesmos visando armazená-los em um banco de dados.

Segundo Silva (2002), a IE estuda metodologias, técnicas e ferramentas que possam encontrar dados específicos dentro de um texto, extraindo automaticamente valores de atributos tais como campos de banco de dados. Em geral, as aplicações nessas áreas são dependentes do domínio, isto é, só apresentam bom desempenho com certas classes de documentos.

A extração de informação é fruto do Processamento de Linguagem Natural (PLN), que estuda os problemas da geração e compreensão automáticas de línguas humanas naturais. Porém os processos envolvidos na Extração de Informação são bem mais simples do que o PLN, e necessitam de definição de que informações devem ser extraídas do texto e que regras devem ser seguidas para que a extração possa ser realizada (TONIAZZO, 2005). O processo de extração de informação identifica as palavras dentro de conceitos específicos e ainda contém um processo de transformação que modifica a informação extraída em um formato compatível com um banco de dados (BARION, LAGO. 2008).

Este processo de extração deve ser feito sobre um tipo de domínio com as informações pré-definidas do que se deseja encontrar no texto. Este domínio é chamado de *Slots*. Fazendo-se uma analogia, os *slots* podem ser comparados a atributo-valor nos bancos de dados tradicionais, devendo conter as informações que se deseja extrair do texto. Por exemplo, em textos sobre transmissões de doença, os *slots* poderiam ser preenchidos com as informações: nome da doença, meios de transmissão, origem da doença, etc. Estas lacunas a serem preenchidas com as informações extraídas do texto são denominadas *Templates* (BARION, LAGO. 2008).

Para exemplificar este tipo de mineração de textos, apresentar-se-á a seguir um exemplo de sua utilização em um texto referente à requisitos para uma vaga de trabalho, adaptado de Mooney e Banescu (1995).

Job Title: Senior DBMS Consultant
Location: Dallas,TX
Responsibilities:
DBMS Applications consultant works with project teams to define DBMS based solutions that support the enterprise deployment of Electronic Commerce, Sales Force Automation, and Customer Service applications.
Desired Requirements:
3-5 years exp. developing Oracle or SQL Server apps using Visual Basic, C/C++, Powerbuilder, Progress, or similar.
Recent experience related to installing and configuring Oracle or SQL Server in both dev. and deployment environments.
Desired Skills:
Understanding of UNIX or NT, scripting language. Know principles of structured software engineering and project management

Figura 2.2 – Texto descrevendo requisitos para uma vaga de emprego.
Fonte: Adaptado de MOONEY, BANESCU, 1995, p. 4.

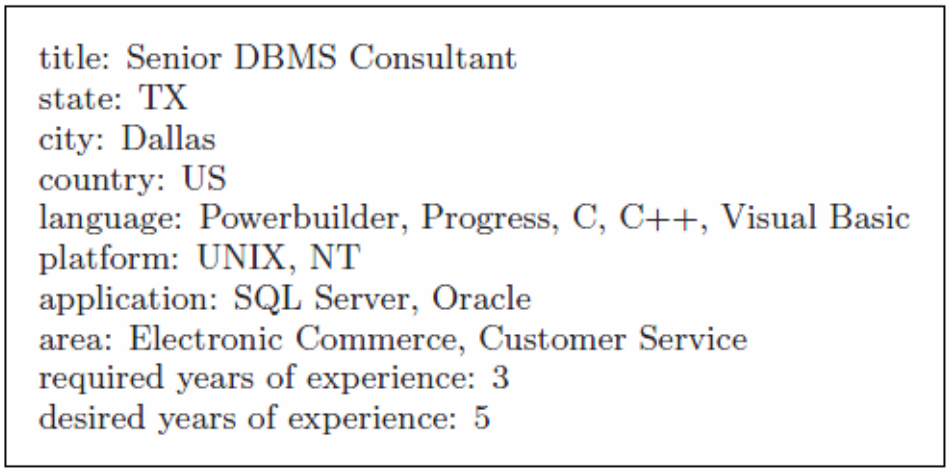


Figura 2.3 – Resultado da aplicação de Extração de Informação sobre a Figura 2.2.
Fonte: Adaptado de MOONEY, BANESCU, 1995, p. 4.

3 ALGORITMOS DE CLASSIFICAÇÃO

Neste capítulo busca-se apresentar os principais algoritmos utilizados em tarefas de classificação de dados. É importante destacar que não existe um modelo ou algoritmo que possa ser usado exclusivamente. Para qualquer problema, a natureza dos dados irá influenciar na escolha do algoritmo escolhido. Não existe um modelo melhor que outro. Enfim, será necessário uma variedade de ferramentas e tecnologias para verificação do melhor modelo a ser utilizado (TWO CROWS CORPORATION. 1999).

3.1 Redes Neurais

Redes neurais são modelos matemáticos inspirados nos princípios de funcionamento dos neurônios biológicos e na estrutura do cérebro. Esses modelos têm capacidade de adquirir, armazenar e utilizar conhecimento experimental e buscam simular computacionalmente habilidades humana tais como aprendizado, generalização, associação e abstração (GOLDSMITH, PASSOS. 2003).

Segundo Jerônimo (2001), essa tecnologia é a que oferece o mais profundo poder de mineração, mas é também a mais difícil de entender, pois tentam construir representações internas de modelos ou padrões achados nos dados, mas estas representações não são apresentadas ao usuário. Elas são utilizadas pelo processo de descoberta de padrões em uma espécie de “caixa-preta”.

A figura 3.1 representa graficamente a estrutura de uma Rede Neural simples. Os círculos representam os neurônios e as linhas representam os pesos das conexões. Pode-se

perceber que uma RNA é dividida basicamente em 3 camadas: a camada de entrada, que é responsável por receber os dados externos; a camada de saída, que mostra os resultados obtidos; e a camada interna ou escondida, que é a camada onde ocorre o processamento interno da rede. Importante destacar que uma RNA pode ter várias camadas escondidas, tudo depende do nível de complexidade do problema.

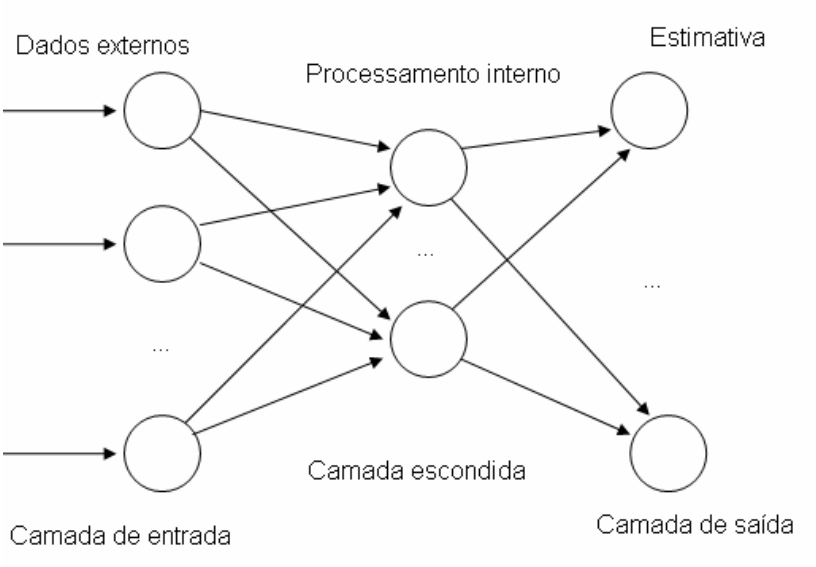


Figura 3.1 – Arquitetura simples de uma Rede Neural.
Fonte: Adaptado de GOLDSMITH, PASSOS, 2003 p. 176.

Segundo Kranz (2004), existem duas técnicas fundamentais na tecnologia das redes neurais. A primeira, chamada de aprendizagem supervisionada, envolve o treinamento da rede, tendo por base um conjunto de dados. Nesta técnica, as várias entradas na rede são pesadas de maneira diferente a partir da experiência da aprendizagem. Esse procedimento é repetido até se obter um ponto ótimo global. A segundo técnica é a aprendizagem não supervisionada. Nesta, os padrões são detectados enquanto os dados são passados através do programa. Como não há pesagem prévia, os padrões são detectados como um subproduto do processo.

Segundo Santos (2008), a desvantagem da utilização de uma RNA está relacionada com o resultado encontrado, no sentido de não ser possível saber como o mesmo foi encontrado, justamente porque a camada onde ocorre o processamento da rede estar escondida. KRANZ (2004) complementa afirmando que existe outra desvantagem da RNAs,

que está relacionada a alimentação dos dados, pois representações de dados diferentes podem produzir resultados diversos.

Uma das principais vantagens da utilização de RNAs está ligada ao fato de a mesma apresentar resultados satisfatórios mesmo tratando-se de áreas complexas, com entradas incompletas ou imprecisas (SILVA. 2008). Outra vantagem da RNA está relacionada com a sua estrutura, que permite à mesma um potencial de ser tolerante a falhas, devido ao armazenamento distribuído das informações na rede. (STAHNKE. 2008).

3.1.1 Back Propagation

Também conhecido como algoritmo de retropropagação do erro, é um algoritmo de aprendizado supervisionado, e tem como objetivo minimizar a função de erro entre a saída gerada pela rede neural e a saída real desejada. (GOLDSMITH, PASSOS. 2003).

Segundo Silva (2008), o treinamento através deste algoritmo tem duas fases de processamento: processamento para frente, ou *forward*, e processamento para trás, ou *backward*. No processamento para frente, um padrão é apresentado à camada de entrada da rede. A atividade resultante percorre a rede, camada por camada, até que uma resposta seja produzida pela camada de saída. No processamento para trás, é feita a comparação da saída obtida com a saída desejada para este padrão particular. Se o resultado não estiver correto, o erro é calculado, sendo o erro propagado a partir da camada de saída até a camada de entrada, modificando os pesos das conexões das unidades das camadas internas, de acordo com a retro propagação do erro.

Este processo é repetido para cada linha no conjunto de treinamento. Cada passagem através de todas as linhas no conjunto de treinamento é chamado de *epoch*. O conjunto de treinamento será usado repetidamente, até que a taxa de erro não sofra mais nenhuma diminuição. Neste ponto, a rede neural é considerada como um treinamento para encontrar padrão no conjunto de teste (TWO CROWS CORPORATION. 1999).

Uma característica importante a ser comentada, é que as Redes Neurais necessitam de uma extensa quantidade de treinamento, que inclui muitos dados e muito tempo, ao menos que

o problema seja muito pequeno. Entretanto, depois de treinada, uma RNA pode chegar a resultados muito rapidamente. (TWO CROWS CORPORATION. 1999).

A seguir é apresentado um gráfico que ilustra a diminuição da taxa de erro com a utilização de algoritmos de treinamento. Pode-se perceber através do gráfico que a taxa de erro tem seu valor decrementado a cada execução do algoritmo em conjuntos de treinamento e que quanto mais uma RNA for treinada, com um conjunto extenso de dados e bastante tempo para sua execução, mais preparada fica para a execução com os dados realmente válidos. Entretanto, chega um ponto em que a Rede Neural alcança seu auge em relação a diminuição de erros. Neste momento a mesma está pronta para utilização.

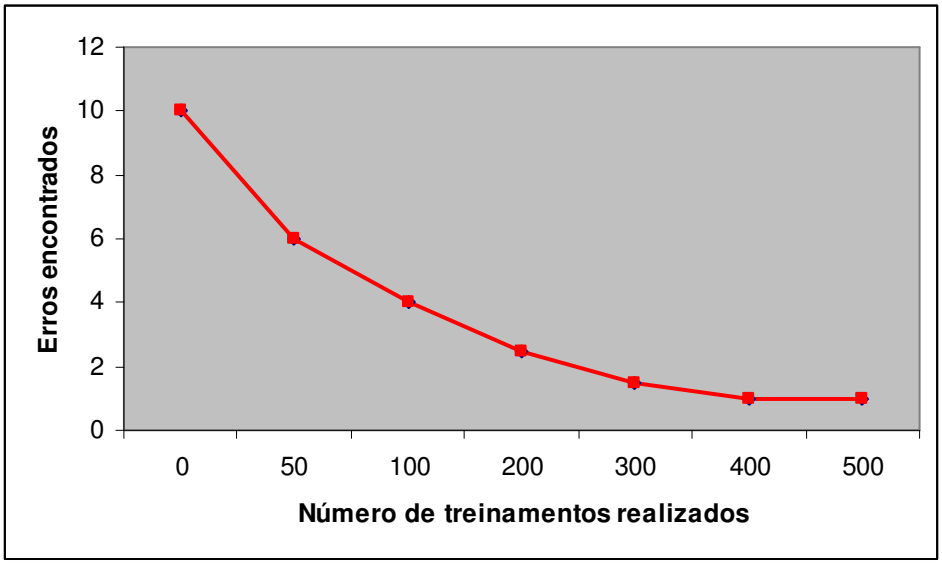


Gráfico 3.1 – Diminuição da taxa de erro de RNAs utilizando algoritmos de treinamento.
Fonte: Adaptado de TWO CROWS CORPORATION, 1999 p. 13.

3.2 Árvores de decisão

As árvores de decisão são meios de representar resultados de tarefas de mineração de dados em uma estrutura semelhante a de uma árvore, daí o motivo de seu nome, e baseiam-se em algoritmos de classificação. Uma árvore de decisão é formada por diversos nós internos, cada um destes representa uma decisão sobre um atributo que determina como os dados estão particionados pelos seus nós filhos.

Muitos estudiosos de Mineração de dados consideram Ross Quinlan, da Universidade de Sidney, Austrália, o "pai das árvores de decisão". Sua contribuição foi um novo algoritmo chamado ID3, desenvolvido em 1983. O ID3, assim como suas evoluções (ID4, ID6, C4.5, See 5) são muito bem adaptadas para usar em conjunto com árvores de decisão, na medida que estes produzem regras ordenadas pela importância (JERÔNIMO. 2001).

Segundo Two Crows Corporation (1999), árvores de decisão são uma forma de representação de uma série de regras que lidam com uma classe ou um valor. Por exemplo, empresas que necessitam classificar clientes como bom pagadores ou não.

Segundo Santos (2008), uma árvore de decisão é composta por um conjunto de nós que são conectados através de ramificações, estes nós se dividem em três grupos:

- **Nodo raiz**: nodo que inicia a árvore;
- **Nodos comuns**: representam uma decisão, dividem um atributo e gera novas ramificações;
- **Nodos folha**: possui informações de classificação do algoritmo.

A interpretação de uma árvore de decisão é feita da seguinte maneira: cada nó não folha representa uma decisão dentro da árvore, envolvendo um atributo e um conjunto de valores possíveis. Os nós folha correspondem à atribuição de um valor ou conjunto de valores a um atributo do problema. Cada caminho da árvore que parte do nó raiz e termina em um nó folha corresponde a uma regra da forma SE <condições> ENTÃO <conclusão> (GOLDSMITH, PASSOS. 2001).

Muitos dos métodos de classificação apresentam um inconveniente: são dificilmente entendidos por humanos, como a técnica de RNA. As Árvores de decisão, por outro lado, são um dos algoritmos de classificação mais facilmente entendidos, haja vista a clareza de como é possível chegar a determinado resultado, utilizando as regras criadas pelo modelo (FELDMAN e SANGER. 2006).

Segundo Goldsmith e Passos (2001), a maioria dos algoritmos baseados na abstração de árvores de decisão consistem em duas fases: Construção da Árvore de Decisão e

Simplificação da Árvore de Decisão. Na fase de construção, uma árvore é gerada pelo particionamento recursivo dos dados de treinamento. O conjunto de treinamento é separado em duas ou mais partições usando restrições sobre os conjuntos de valores de cada atributo. O processo é repetido recursivamente até que todos ou a maioria dos exemplos em cada partição pertençam a uma classe. A fase de Simplificação visa tornar árvores de decisão mais complexas em modelos mais simples. Esta fase pode ser realizada enquanto a árvore está sendo gerada, decidindo pela não divisão de um conjunto de casos de treinamento ou removendo estruturas durante o processo. Ainda existem algoritmos como o C4.5 que executam a fase de simplificação da árvore somente depois de sua construção.

Basicamente, a forma de execução para a criação de árvores de decisão é a seguinte: dado um conjunto de dados cabe ao usuário escolher uma das variáveis como objeto de saída. A partir daí, o algoritmo encontra o fator mais importante em relação a variável de saída e coloca-o como raiz da árvore. Os demais fatores subseqüentes são classificados como nós até que se chegue ao último nível, a folha.

A seguir é apresentado um exemplo simples de resultados obtidos a partir de uma base de dados utilizando-se árvore de decisão e, logo após, as regras criadas a partir da mesma.

NOME	**ESCOLARIDADE**	**IDADE**	**RICO** (***atributo classe***)
Alva	Mestrado	>30	Sim
Amanda	Doutorado	<=30	Sim
Ana	Mestrado	<=30	Não
Eduardo	Doutorado	>30	Sim
Inês	Graduação	<=30	Não
Joaquim	Graduação	>30	Não
Maria	Mestrado	>30	Sim
Raphael	Mestrado	<=30	Não

Tabela 3.1 – Base de dados antes da aplicação do algoritmo de Árvore de Decisão.
Fonte: GONÇALVES.

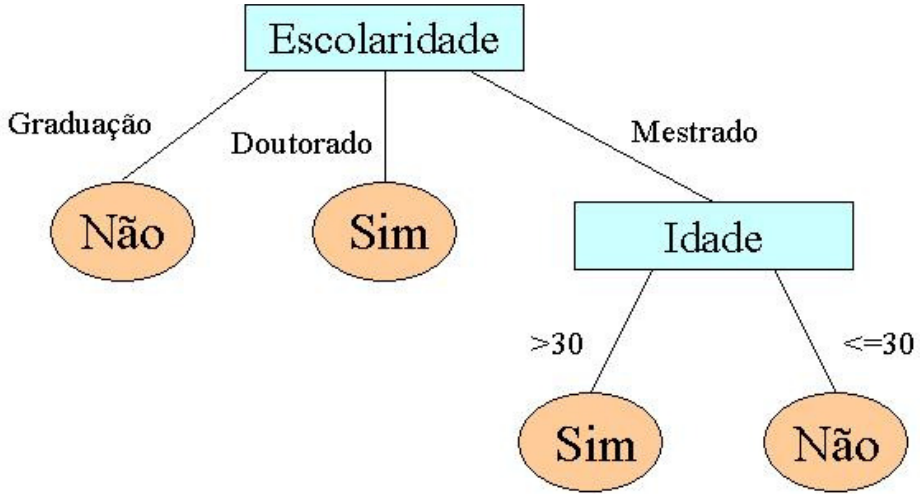


Figura 3.2 – Árvore de decisão construída a partir do conjunto de dados da Tabela 3.1.
Fonte: GONÇALVES.

Se (Escolaridade = Graduação) então Rico = Não
Se (Escolaridade = Doutorado) então Rico = Sim
Se (Escolaridade = Mestrado) e (Idade >30) então Rico = Sim
Se (Escolaridade = Mestrado) e (Idade <=30) então Rico = Não

Figura 3.3 – Regras que compõem Árvore de decisão da Figura 3.2.
Fonte: GONÇALVES.

Segundo Silva (2008), as desvantagens de utilização de árvores de decisão estão relacionadas a necessidade de haver uma quantidade considerável de dados para que seja possível descobrir estruturas complexas e na possibilidade de haver erros na classificação, no caso de existirem muitas classes. Segundo Jerônimo (2001), as principais vantagens das árvores de decisão estão relacionadas ao fato de que elas fazem decisões considerando o grau de relevância, e ao fato de serem perfeitamente entendidas pela maioria das pessoas.

Visando um maior entendimento dos algoritmos de Árvores de Decisão, apresenta-se a seguir um dos mais tradicionais destes: o C4.5.

3.2.1 Algoritmo C4.5

O C4.5 é um dos algoritmos mais utilizados na tarefa de Classificação. Este método visa abstrair árvores de decisão a partir de uma abordagem recursiva de particionamento das bases de dados e é baseado em seu antecessor, o ID3. Ambos foram desenvolvidos por John Ross Quinlan (GOLDSMITH, PASSOS. 2005).

Segundo Stahnke (2008), o objetivo deste algoritmo é abstrair árvores de decisão baseado em um método recursivo de particionamento das bases de dados. A maioria destes algoritmos consiste em duas fases: a construção da árvore de decisão e sua simplificação, conforme comentado anteriormente em Árvores de Decisão.

Segundo Silva (2008), este algoritmo transforma a árvore de decisão em um conjunto de regras ordenadas pela sua importância, para facilitar a identificação de fatores mais importantes da estrutura.

A regra diagnosticada como sendo de mais relevância para o contexto é apresentada na árvore como o primeiro nó, e as regras menos relevantes são posicionadas nos nós mais abaixo, ou seja, quando mais alto for o nó em questão, maior é sua relevância dentro das decisões encontradas pela árvore de decisão (JERÔNIMO. 2001).

3.3 Algoritmos Genéticos

São algoritmos que simulam o processo de seleção natural proposto por Charles Darwin, em 1859. Segundo Darwin, a seleção natural é um processo que privilegia os organismos que melhor se adaptam ao meio ambiente, isto é, quanto mais adaptado o organismo está ao seu ambiente, maior a chance de sobrevivência e mais características ele irá transmitir para seus sucessores por meio de seus cromossomos. Os Algoritmos Genéticos (AG) usam esta mesma propriedade para desenvolver seus modelos. Vários modelos são estudados, mas apenas aqueles que se mostram mais hábeis para o encontro da solução desejada são desenvolvidos (AMARAL. 2001).

Segundo Two Crows Corporation, são chamados de Algoritmos Genéticos pelo fato de seguirem padrões de evolução biológica, onde os membros de uma geração competem para transmitir suar características para a próxima geração, até que a geração mais próxima da perfeição seja encontrada. A informação transmitida às novas gerações está contida em "cromossomos", que contém os parâmetros para criação do modelo.

Segundo Goldsmith e Passos (2005), Algoritmos Genéticos são técnicas que procuram obter boas soluções para problemas complexos por meio da evolução de populações de soluções codificadas em cromossomos artificiais. Empregam um processo adaptativo e paralelo de busca de soluções em problemas complexos. O processo é adaptativo, pois as soluções existentes a cada instante influenciam a busca por futuras soluções. O paralelismo do processo é decorrência natural do fato de que diversas soluções são consideradas a cada momento pelos Algoritmos Genéticos.

De modo geral, o aprendizado genético inicia-se com a criação de uma população constituída de regras geradas aleatoriamente. Cada regra pode ser representada por uma cadeia de *bits*. Por exemplo, supondo-se que as amostras em um conjunto de treinamento sejam descritas por duas variáveis booleanas, A1 e A2, e que existam duas classes, C1 e C2. A regra "Se A1 e não A2 então C2 pode ser codificada como a seqüência de *bits* "100", onde os dois caracteres a esquerda representam os atributos booleanos A1 e A2 e o último representa a classe. Do mesmo modo, a regra "Se não A1 e não A2 então C1", é codificada como "001". Se um atributo tiver *k* valores, então *k bits* devem ser usados para codificar os valores do mesmo. (HAN, KAMBER. 2001).

Cada um dos indivíduos da população criada representa uma possível solução para o problema, pois cada um destes é formado por cromossomos, que são, na verdade, uma cadeia de *bits* que apresentam uma solução. Esta cadeia de *bits* é formada através da codificação das regras citadas anteriormente. Deste modo o processo de solução adotado nos Algoritmos Genéticos consiste em gerar, através de regras específicas, um grande número de indivíduos, denominada população, de forma a promover uma varredura tão extensa quanto necessária para todas as possíveis de soluções do problema, e caso não for encontrada uma solução, uma nova população será gerada pelo processo de reprodução, que se estabelece de três modos: mutação, reprodução e cruzamento.

A seguir é apresentada a estrutura básica do Algoritmo Genético. Onde cada iteração do algoritmo corresponde à aplicação de um conjunto de operações básicas como: seleção do indivíduo, avaliação de seu cromossomo, verificação se solução do problema proposto foi encontrada, e se necessário a criação de uma nova população.

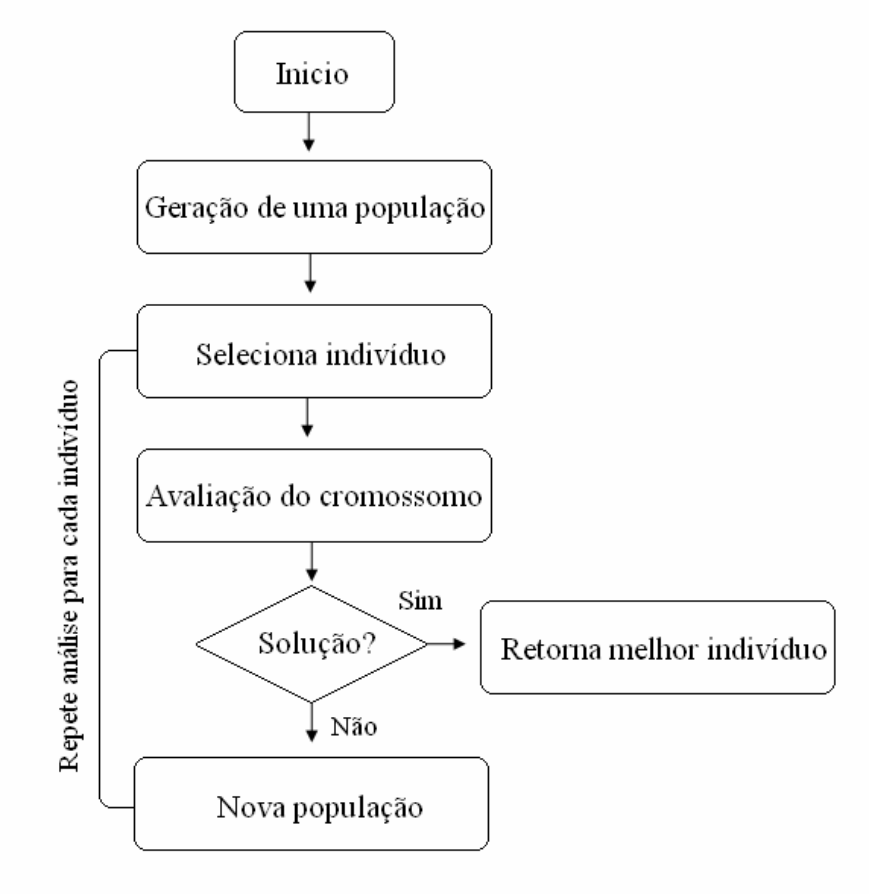


Figura 3.4 – Estrutura básica de um Algoritmo Genético.
Fonte: Adaptado de COX, 2009, p. 244.

4 APLICAÇÕES DE DATA MINING E TEXT MINING

Neste capítulo são apresentadas algumas aplicações que foram implementadas e que utilizam técnicas de *Data Mining* e *Text Mining*. Sendo assim, a primeira aplicação está relacionada com o uso de descoberta de conhecimento em banco de dados, fazendo uso de algoritmos de classificação e associação. As duas aplicações seguintes estão relacionadas com a descoberta de conhecimento em texto, porém uma faz uso de algoritmos de associação e outra utiliza algoritmos de classificação.

4.1 Uma aplicação de Mineração de Dados à Previsão de *No-Show* no Agendamento de Serviços médicos (ADEODATO, SANTOS, VASCONCELOS, SANTOS. 2005)

O assunto desta aplicação de mineração de dados está relacionado com um problema que ocorre frequentemente em relação à prestação de serviços. Trata-se do problema de não comparecimento a compromissos agendados por parte dos clientes, muito comum em serviços de assistência médica e de transporte de passageiros. Este problema é chamado de *no-show* que, em sua tradução literal, significa "não comparecimento", justamente pelo fato de que os clientes não comparecerem aos compromissos e não se comunicarem previamente sobre a impossibilidade de comparecimento.

O grande prejudicado nestes casos é o prestador de serviços, que tem prejuízos pela alocação desnecessária de recursos. É necessário um dimensionamento por parte do mesmo para a alocação de recursos de forma a minimizar as perdas, muitas vezes sendo necessário o balanceamento entre a disponibilização de recursos não utilizados e a possibilidade de faltarem recursos para o atendimento.

Diante deste problema, este artigo apresenta um estudo realizado sobre o problema do *no-show* em consultas médicas a partir do agendamento via c*all-center*, para uma empresa prestadora de serviços médicos na cidade de Recife. Sendo assim, foram destinados esforços para o desenvolvimento de um sistema de apoio à decisão na estimativa do risco de *no-show* com base em uma amostra das consultas agendadas pelos pacientes.

O problema do *no-show* ocorre quando o paciente não comparece à consulta ou se atrasa em mais de uma hora em relação ao horário marcado. Sendo assim, o problema em questão irá considerar estes dois casos para a busca de conhecimento. Essa caracterização transforma o a previsão do *no-show* em um problema de classificação binária.

Para a busca de solução do problema foi utilizada uma base de dados com cerca de 50.000 consultas agendadas para uma empresa de assistência médica na cidade de Recife-PE em 2003. Foi realizada uma alteração em relação aos campos existentes do atendimento, sendo criado um novo campo denominado *Status*. Este novo campo possui a informação a *posteriori,* categorizada em uma variável dicotômica, que irá indicar se houve ou não o comparecimento do paciente à consulta agendada.

No desenvolvimento do artigo foram utilizadas técnicas de Redes Neurais Artificiais, para estimar o risco que um paciente oferece em relação ao comparecimento ou não à consulta agendada, e também o algoritmo Apriori, para tentar encontrar relações entre as consultas antigas, buscando identificar as regras de associação.

Primeiramente, foi feita a busca por relações entre as consultas, utilizando-se do algoritmo Apriori. O mesmo foi realizado em 3 etapas:

1) Aplicação do algoritmo para induzir as regras que satisfazem a uma cobertura mínima e que têm uma complexidade limitada;
2) Seleção das regras com níveis de cobertura mais amplos que os induzidos originalmente e níveis de confiança de interesse (*lift)* para cada classe;
3) Seleção das regras que apresentam mais interesse sob os pontos de vista do especialista e da eliminação de redundância nas regras.

O *lift*, referenciado na 2ª etapa, mede a qualidade das regras, verificando o aumento da confiança da regra em relação à confiança média da amostra analisada. A seguir é apresentada uma imagem ilustrando as regras extraídas utilizando-se o algoritmo Apriori.

Percentual de *no-show* da massa inteira: 48%							
Regras de No-show (Ruins)				**Regras de Comparecimentos (Boas)**			
Condição	**Suporte**	**Confiança**	**Lift**	**Condição**	**Suporte**	**Confiança**	**Lift**
Especialidade = Cardiologia	4.48%	70.99%	1.48	Especialidade = Fonoaudiologia	3.08%	74.12%	1.43
Tele_operadora = Salete	5.02%	68.53%	1.43	Tele_operadora = "Cris"	6.05%	89.76%	1.73
Especialidade = Ginecologia e Tempo_Consulta = 6 a 11 dias	10.46%	64.36%	1.34	Especialidade = "Pediatria" e Tele_operadora = "Cris"	3.48%	96.40%	1.85

Figura 4.1 – Regras extraídas com o algoritmo Apriori.
Fonte: ADEODATO, SANTOS, VASCONCELOS, SANTOS, 2005, p. 3.

Para execução da tarefa de estimativa do risco que um paciente oferece em relação ao comparecimento ou não à consulta agendada, utilizou-se a técnica redes neural *Multilayer Perceptron* (MLP), treinada com o algoritmo *back propagation*, modelo utilizado com sucesso na solução de problemas de classificação de padrões. Dentre as características mais atrativas deste tipo de rede neural os autores do artigo destacam a sua excelente capacidade de generalização, a simplicidade de operação e o fato de a mesma ser um aproximador universal de funções e produzir uma resposta contínua que permite uma decisão baseada em limiar sobre uma grandeza escalar (o escore) para separar as duas classes. Esse escore possibilita o monitoramento mais refinado do desempenho do sistema de apoio à decisão.

Os resultados apresentados foram realizados sobre um conjunto de teste com 3.389 ocorrências, estatisticamente independentes dos dados de modelagem. Há uma grande separação entre as distribuições dos escores do *no-show* e do comparecimento, indicando que o sistema é capaz de prever o *no-show* com uma boa margem de acerto. Na Figura 4.2 pode-se perceber que o comparecimento às consultas está concentrado nos altos escores, enquanto os casos de *no-show* estão nos baixos escores.

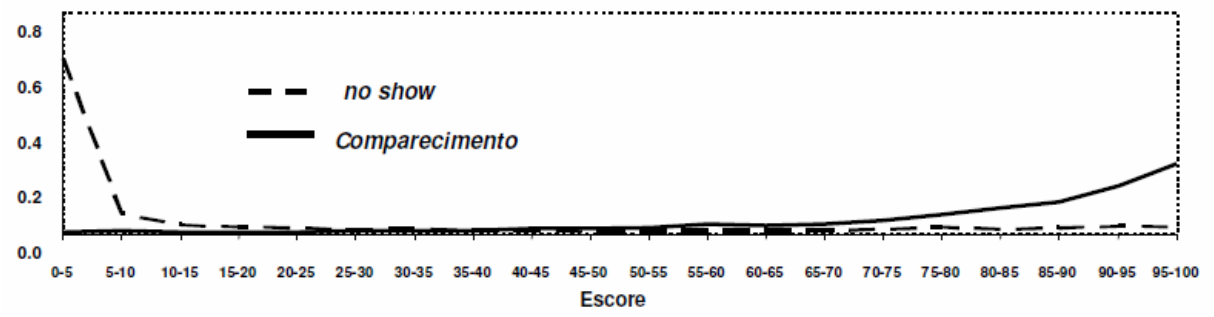


Figura 4.2 – Distribuição das classes em função do escore.
Fonte: ADEODATO, SANTOS, VASCONCELOS, SANTOS, 2005, p. 3.

A aplicação de mineração de dados ao problema do *no-show* obteve resultados estatisticamente relevantes na estimação do nível de comparecimento dos pacientes, segundo um experiente especialista da empresa de assistência médica. A qualidade das regras permitiu melhor entendimento das causas do *no-show*. Os autores propõem a implantação deste sistema com o objetivo de tentar classificar os pacientes no momento do agendamento da consulta, assim, as operadoras do *call-center* digitariam os dados e teria, *online*, o risco do *no-show* que o paciente representa, e daí então poderiam tentar convencer o paciente a fazer um re-agendamento para a redução desse risco.

4.2 Aplicação de Text Mining para a Extração de Conhecimento Jurisprudencial (BEPPER, FERNANDES. 2005)

O principal objetivo desta aplicação de *Text Mining* estava relacionado com a extração de conhecimentos implícitos de textos no domínio do direito, extraindo automaticamente tais conhecimentos através de uma análise requisitada pelo usuário. Atualmente, os Tribunais de Justiça possuem uma grande quantidade de informações na forma textual. Esses tribunais produzem textos jurídicos descrevendo decisões sobre os recursos interpostos em primeira instância.

Dentre os textos jurídicos se encontra a Jurisprudência, que representa o conjunto uniforme e constante das decisões judiciais sobre casos semelhantes. Sendo assim, surgiu a necessidade de fazer um estudo de técnicas de *text mining* e por meio destas, desenvolver uma ferramenta que terá como objetivo extrair conhecimento em relação à jurisprudência.

Como exemplo do funcionamento do sistema, pode-se considerar o seguinte modelo: determinado usuário pretende fazer uma consulta para extrair alguma informação sobre furto de banco. Após aplicadas as técnicas de *text mining* o sistema retorna que, de todos os julgamentos, que correspondem a furto de banco, 80% está relacionado com o termo reincidência, ou seja, após o cumprimento da sentença a pessoa praticou um crime da mesma natureza. Deste modo este conhecimento poderá ser utilizado, por exemplo, por Advogados em favor da defesa de seu cliente, alegando que as providências que vêm sendo tomadas por juízes e tribunais não vêm surtindo efeito esperado. Da mesma forma, o conhecimento poderá ser utilizado também por Promotores de Justiça, mas com o objetivo de melhorar a eficácia na defesa dos interesses da sociedade.

A arquitetura da ferramenta desenvolvida é composta por quatro módulos, que são: Módulo de Entrada, Módulo de Saída, Pré-Processamento dos Dados e Aplicação da Técnica. Em seguida é apresentado o detalhamento de cada um destes.

- **Módulo de Entrada:** responsável por adquirir as informações para que o processo de análise possa ser iniciado. Baseia-se no preenchimento, por parte do usuário, de quatro campos, que são: seleção dos documentos, que identifica quais e quantos documentos farão parte do processo; numero de regras a serem descobertas: especifica a quantidade de regras que se deseja que retorne após a análise; fator de suporte: valor porcentual que indica o grau mínimo de suporte; e fator confiança: valor porcentual que indica o grau mínimo de confiança.
- **Módulo de Saída:** representa as informações que foram extraídas e apresentam as mesmas ao usuário. Estas informações resultam do conhecimento implícito extraído dos documentos textuais, mais especificamente, são as regras de associações aplicadas aos textos retornando padrões úteis a serem analisados.
- **Pré-Processamento dos Dados:** tem como objetivo remover ruídos e preparar os dados para que a técnica de extração de conhecimento possa ser aplicada. Dentro deste módulo existem algumas tarefas como: limpeza de caracteres indesejados (que visa ignorar todos os caracteres especiais, pois não contribuirão para a aplicação), remoção de *stopwords* e o processo de *stemmin*g.

- **Aplicação da Técnica:** considerada a fase mais importante, pois é nela que se extrai o conhecimento implícito. Tem como objetivo encontrar associações, padrões ou correlações em conjuntos de itens na linguagem natural. O sistema irá aplicar a técnica analisando todos os textos-fonte que passaram pela fase de Pré-processamento. Irá retornar a quantidade de regras solicitadas pelo usuário que estejam acima do porcentual dos fatores de confiança e suporte mínimo definidos pelo mesmo.

Neste trabalho utilizou-se o algoritmo Apriori do *software* Weka (*Waikato Evironment for Knowledge Analysis*) para execução da tarefa de criação de regras de associação. Antes da utilização do Weka para análise dos documentos, houve a necessidade de um tratamento sobre termos que seriam analisados, ou seja, foi realizada uma ordenação dos termos a serem analisados. Todos os documentos foram pré-processados com o objetivo de encontrar os termos mais freqüentes e realizar a ordenação de forma decrescente. Foi criada uma lista para armazenamento dos termos freqüentes, sendo que os termos mais freqüentes localizam-se no topo da mesma. Esses termos do topo da lista foram considerados como os melhores termos a serem usados na obtenção das regras de associação.

A figura a seguir ilustra todas as etapas necessárias antes da possibilidade do uso do software Weka. Os dados prontos resultantes deste modelo estão relacionados aos dados que podem ser redirecionados pelo uso do Weka. A etapa "Preparação para Weka" está relacionada à adequação dos dados para que os mesmos possam ser utilizados pelo software, pois este possui um formato próprio.

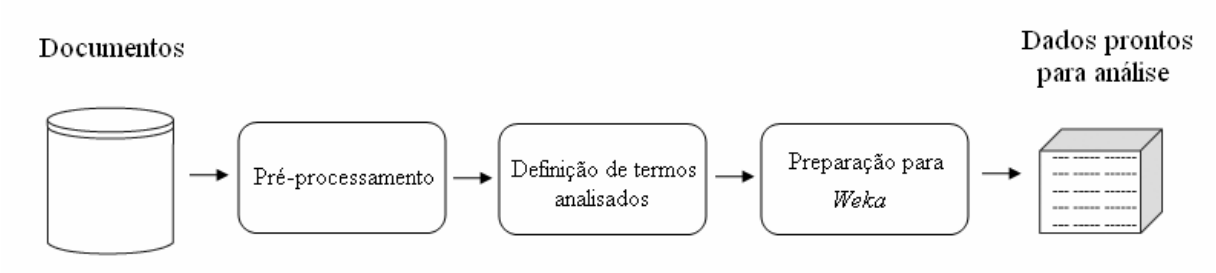


Figura 4.3 – Etapas necessárias antes da mineração de dados com uso do Weka.
Fonte: Autor.

A ferramenta desenvolvida foi aplicada às jurisprudências do Tribunal de Justiça do Estado de Santa Catarina (TJSC) e como resultado da mesma obteve-se a descoberta de regras de associação com alto fator de confiança. Um dos pontos positivos da aplicação foi a utilização do módulo de pré-processamento, que foi muito eficaz e reduziu consideravelmente o tempo computacional. Um dos pontos negativos da aplicação foi relacionado a algumas regras de associação que não apresentavam nenhum conhecimento interessante para o usuário, por se tratarem de relacionamento óbvio entre palavras no antecedente e no conseqüente da regra.

4.3 Aplicação de Text Mining na Validação de Registros de Ocorrência Policiais na Região da Grande Florianópolis (SILVA. 2005)

Este estudo foi realizado com o objetivo de minerar dados oriundos de Registros de Ocorrência (RO) da Polícia Militar de Santa Catarina. O motivo da realização do trabalho está relacionado com a necessidade de um processo automatizado na validação, contabilização, levantamentos estatísticos dos Registros de Ocorrência, para que seja feito o apontamento de diretrizes para o combate ao crime, esta tarefa é de responsabilidade da Secretaria de Estado da Segurança Pública e da Cidadania de Santa Catarina. Sendo assim, a mineração de textos auxilia neste processo, pois classificando e tratando os documentos, facilita na procura e contagem de ocorrências policiais pela sua natureza de operação.

Inicialmente, foram utilizados dados de 2.684 ROs policiais do ano de 2003 da Região Metropolitana da Grande Florianópolis, e os esforços foram direcionados na criação de um processo que verificasse a confiabilidade e validasse a classificação já atribuída ao RO, apontando erros encontrados, que seriam direcionados à revisão do analista da Diretoria de Combate ao Crime Organizado (DIRC). Durante o estudo houve a necessidade do desenvolvimento de algumas ferramentas que auxiliassem na execução do processo, como o *software* denominado ABC Clean, que tinha como objetivo a correção ortográfica dos termos, e o *software* denominado ABC Mining, que era responsável pela mineração dos dados. Ambos desenvolvidos pela autor do trabalho. Utilizou-se também o *software* Weka para a geração das regras de decisão do tipo Se...Então, utilizando como técnica a Árvore de Decisão.

Com a geração das árvores de decisão, pretendeu-se validar se o registro de ocorrência estava classificado na natureza de operação correta e sugerir a DIRC que implemente em seu sistema de cadastro de RO essas regras, evitando-se assim que novos ROs fossem cadastrados com a natureza de operação distorcida em relação a descrição.

Um registro de ocorrência é um documento legal elaborado pela Polícia que representa a primeira notificação oficial de uma queixa-crime, para a maior parte dos casos, que são encaminhados a uma unidade de polícia judiciária. Ao cadastrar-se um RO é atribuído ao mesmo sua natureza de operação, que determina qual o delito foi cometido. Neste estudo foram utilizadas 17 categorias de natureza de operação de um total de 732 tipos. Com citado anteriormente, parte dos ROs encontram-se classificados em natureza de operação incorreta. Para solução deste problema é feita uma comparação entre a descrição do RO, que é um breve texto descrevendo o ocorrido, com a natureza de operação já atribuída. O autor cita o seguinte exemplo: RO número 1030271, onde o policial informou a natureza de informação C903 – comunicação falsa, e a descrição da RO tem como palavras-chave arma, projéteis, óbito, que informa claramente que o RO pertence na verdade à natureza de operação D309 – óbito no local, ou seja, ocorreu um erro no cadastro da RO.

Quando uma ocorrência policial está cadastrada na natureza de operação incorreta, produz dados estatísticos distorcidos, que por sua vez, leva a polícia a elaborar políticas de combate ao crime nem sempre eficazes. Sendo assim, foi em relação a este aspecto que o trabalhou foi realizado, na criação de um processo para classificação da RO em relação a sua descrição.

Para o pré-processamento dos dados, foi desenvolvida a ferramenta desenvolvida ABC Clean, pelo fato de não existir nenhuma ferramenta que fizesse a limpeza para a língua portuguesa falada no Brasil, que ficou responsável pela correção ortográfica dos termos e espaçamento incorreto, como por exemplo “armadefogo”. A seguir é apresentada a Figura 4.4 que ilustra a interface do ABC Clean relacionada à etapa de correção ortográfica.

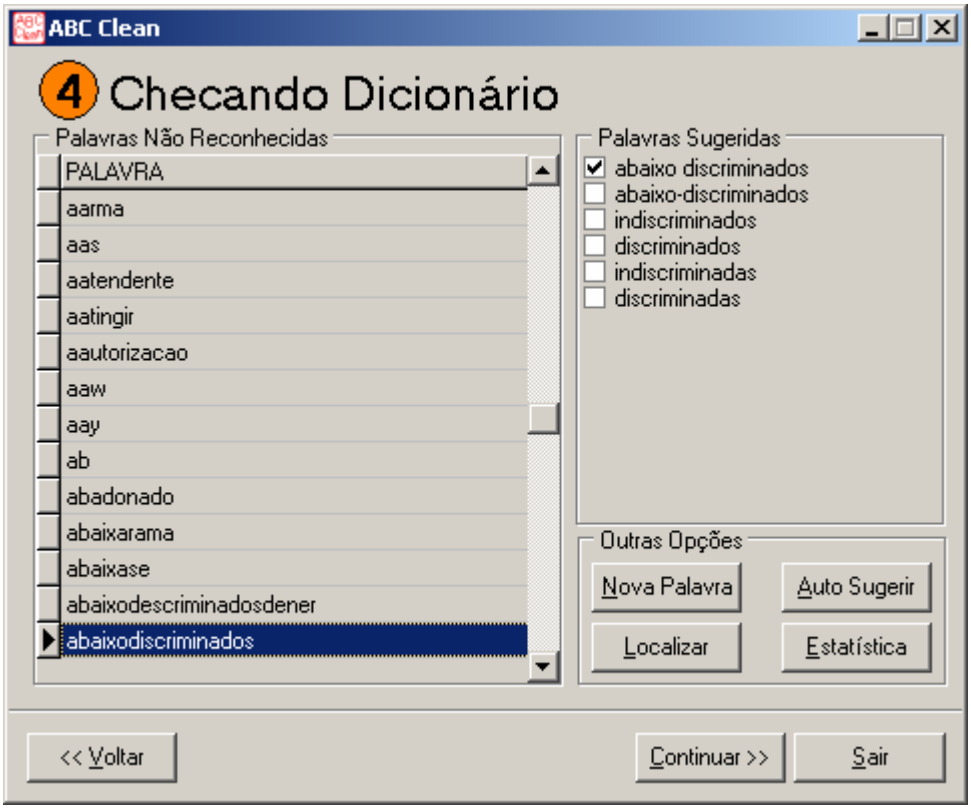


Figura 4.4 – Correção ortográfica utilizando ABC Clean.
Fonte: SILVA, 2005, p. 74.

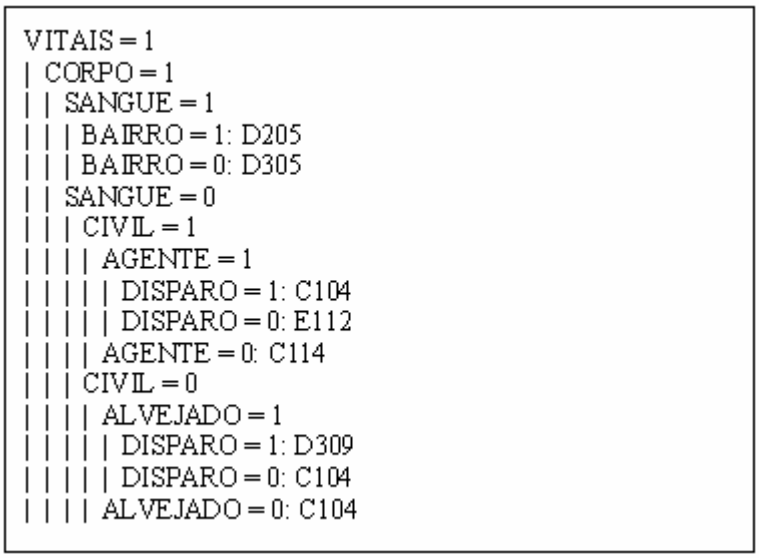
Após a correção dos dados, foi iniciada a fase de mineração utilizando-se da ferramenta desenvolvida ABC Mining. As principais operações da ferramenta são as seguintes:

- **Remoção de *stopwords***: remoção das palavras que não apresentam nenhum conhecimento em relação a aplicação;
- **Definição de *keywords***: criação de um conjunto de palavras-chave para cada categoria, ou seja, as palavras mais freqüentes encontradas em documentos de uma determinada categoria que expressem algum sentido. Uma parte das *keywords* encontradas pode ser visualizada na Figura 4.5;
- **Exportação para Weka**: é realizada a criação de um arquivo no formato suportado pelo Weka para que fosse possível descobrir quais palavras aparecem ou não em cada registro de ocorrência.

natureza	palavra1	palavra2	palavra3
E123	AGREDIDO		
E112	MASCULINO	VITIMA	INTERIOR
E111	ARMA	AGENTE	HOSPITAL
D316	POSTO	ALVEJADO	POPULARES
D309	ARMA	FOGO	VITAIS
D305	MOTO	SANGUE	MORRO
D205	CORPO	RESIDENCIA	ARMA
C610	FOGO	ARMA	DISPARO
C222	ASSALTO	ESTABELECIMENTO	GUARNICAO
C221	VEICULO	ASSALTO	TOMADO
C215	VEICULO	FEMININA	BAIRRO
C207	VEICULO	SEQUESTRO	RELAMPAGO
C114	VITIMA	MASCULINO	FOGO
C112	VEICULO	POSTO	SEQUESTRO
C104	FOGO	AGENTE	ARMA
A203	HOTEL	CORPO	CIVIL

Figura 4.5 – *Keywords* encontradas pelo ABC Mining para cada natureza de operação.
Fonte: SILVA, 2005, p. 83.

A partir do arquivo gerado pelo ABC Mining pode-se gerar a árvore de decisão com o auxílio do WEKA. Utilizou-se para isso o algoritmo ID3 disponível no *software*. Da árvore de decisão gerada pelo WEKA pode visualizar-se um trecho na Figura 4.6.



```
VITAIS = 1
| CORPO = 1
| | SANGUE = 1
| | | BAIRRO = 1: D205
| | | BAIRRO = 0: D305
| | SANGUE = 0
| | | CIVIL = 1
| | | | AGENTE = 1
| | | | | DISPARO = 1: C104
| | | | | DISPARO = 0: E112
| | | | AGENTE = 0: C114
| | | CIVIL = 0
| | | | ALVEJADO = 1
| | | | | DISPARO = 1: D309
| | | | | DISPARO = 0: C104
| | | | ALVEJADO = 0: C104
```

Figura 4.6 – Parte da árvore de decisão gerada pelo Weka.
Fonte: SILVA, 2005, p. 85.

Como resultado da aplicação, obteve-se que dos 2.684 registros de ocorrência minerados, 146 registros encontravam-se na natureza de operação incorreta, ou seja, 5.4396% do total de registros estavam classificados incorretamente. Esse percentual tende a aumentar, com a utilização de uma base de dados maior, e sem a seleção prévia dos possíveis registros corretos. Utilizou-se aproximadamente 50% dos ROs atendidos em 1 dia de atendimento do serviço 190. O modelo proposto mostrou-se eficaz para esses ROs e está apto a ser utilizado com outras bases de dados.

Sendo assim, na próxima etapa deste trabalho, utilizar-se-á o conhecimento adquirido para o desenvolvimento de um processo que fará uso das técnicas estudadas. Esta aplicação que será realizada tem relação com a descoberta de conhecimento em dados oriundos de programas *Keyloggers*. Como os dados gerados por este tipo de programa estarão na forma textual, será necessária a utilização de tarefas existentes nos dois tipos de descoberta de conhecimento estudados.

O objetivo da aplicação que será desenvolvida é a classificação dos dados adquiridos em relação ao contexto de sua aplicação. Por exemplo, caso seja aplicado em um ambiente empresarial, iremos buscar criar um processo de classificação dos dados de acordo com o foco da empresa, em relação à medição de que palavras capturadas se enquadram no contexto da empresa, podendo assim realizar um relatório de medição de produtividade do usuário. Outra possibilidade seria a aplicação do trabalho em um ambiente de ensino, onde seria realizada a medição de utilização do computador por cada aluno, sendo assim possível a classificação do conteúdo capturado em relação à tarefa proposta pelo professor.

Primeiramente será realizado o desenvolvimento da ferramenta responsável pela captação dos dados e, em seguida, sua aplicação em algum tipo de ambiente organizacional, para que tenha-se conhecimento dos dados que serão utilizados e iniciarmos o tratamento em busca de sua normalização para utilização na mineração propriamente dita, com o objetivo de descoberta de conhecimento.

5 DESCOBERTA DE CONHECIMENTO A PARTIR DE KEYLOGGERS

Um dos fatores mais importantes para a realização deste trabalho, está relacionado diretamente aos dados utilizados. Desta forma o trabalho contempla a captura dos dados para análise posterior. Os dados foram capturados com a utilização de uma ferramenta, desenvolvida no decorrer deste trabalho, que desempenha o papel de *keylogger* quando executado em um computador, e que será detalhada mais adiante neste capítulo.

Através de uma autorização de um professor da Universidade Feevale e com consentimento dos alunos a ferramenta *keylogger* foi submetida a utilização em sala de aula, visando capturar informações inseridas pelos mesmos durante a aula para criação de uma base de dados. De maneira geral, pretende-se capturar dados com o uso da ferramenta *keylogger* para que estes sejam submetidos ao processo de mineração.

O principal objetivo da mineração dos dados está relacionado a comparação dos dados obtidos com os que supostamente deveriam ser encontrados com a utilização da ferramenta *keylogger*. Sendo assim, será realizada a classificação dos dados capturados em relação ao conteúdo da disciplina. A ferramenta foi utilizada em um dia de aula em que uma atividade foi proposta pelo professor, sobre determinado assunto coerente com a disciplina. Deste modo, tendo conhecimento do assunto da atividade proposta, é possível que seja feita a classificação dos dados capturados em relação ao conteúdo pertinente à atividade.

Durante a aplicação deste trabalho, buscou-se não comprometer nenhuma informação que pudesse ser sigilosa ou a divulgação de que aluno tais dados foram capturados. Deste modo decidiu-se fazer a distinção de informações baseando-se somente no computador utilizado, ou seja, levando-se em consideração o nome do mesmo.

Para que o processo de captura de dados fosse mais automatizado, ou seja, não houvesse a necessidade de aquisição manual do arquivo de dados capturados gerado pelo *keylogger*, em cada computador, uma segunda ferramenta foi desenvolvida com o intuito de centralizar os dados capturados de todos os *keyloggers* (1). Esta ferramenta é responsável pela criação da base de dados que será utilizada no processo de mineração (2), conforme demonstra a Figura 5.1.

Com a base de dados criada é possível a realização da etapa de pré-processamento (3), que visa a remoção de dados inválidos e desnecessários dentro do processo, criando assim uma nova base de dados (4). Esta nova base de dados deve ser encaminhada a etapa de treinamento (5), que é responsável pelo aprendizado supervisionado em relação a classificação das palavras encontradas. A etapa de treinamento tem como objetivo a criação de um arquivo de treinamento (6), o qual contém informações sobre a classificação das palavras. Com a utilização do arquivo de treinamento criado, é possível a geração de resultados estatísticos (7), em relação a classificação das palavras de acordo com cada computador em que a captura de dados foi realizada. Estes resultados são apresentados ao analista responsável pelo processo (8) e o mesmo deve verificar se existe um resultado satisfatório e, caso não exista, o processo de treinamento deve ser realizado novamente (9). Caso o resultado seja satisfatório o processo é dado como encerrado, pois o conhecimento desejado foi encontrado (10).

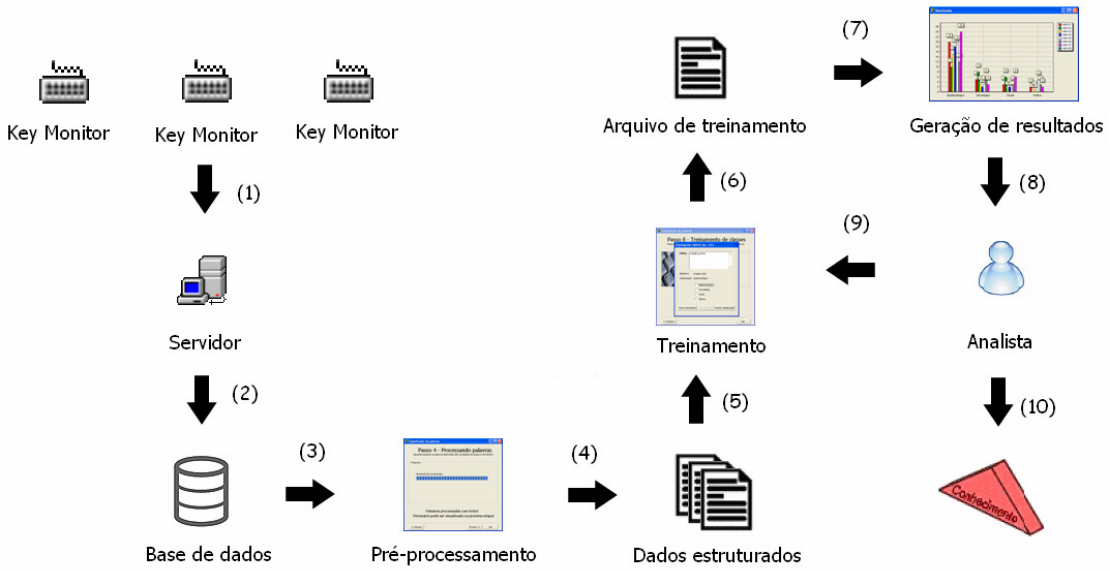


Figura 5.1 – Modelo geral do processo utilizado para realização do trabalho.
Fonte: Autor.

Todas as etapas apresentadas na Figura 5.1 serão abordadas detalhadamente nas sessões que se seguem. Porém, antes de prosseguirmos para o detalhamento das etapas do processo de mineração, será apresentada uma seção que visa apresentar esclarecimentos sobre aplicativos *keyloggers*.

5.1 Keyloggers

O termo *keylogger* vem do inglês e significa "registrador de teclas", seu principal objetivo é monitorar tudo o que for digitado em um computador. Atualmente, sua principal utilização está relacionada à coleta de informações sigilosas, sejam estas senhas, números de cartão de crédito e afins, ou seja, atuando como um vírus. No mundo dos golpes virtuais e vírus, o *keylogger* é considerado uma das mais eficientes e atuantes pragas virtuais.

Segundo Lindgren e Sullivan (2006), um *keylogger*, também conhecido pelo termo keystroke *logging*, consiste, basicamente, na gravação da atividade relacionada a utilização do teclado em um computador. Esta tecnologia é utilizada como uma ferramenta de observação que oferece a funcionalidade de capturar detalhadamente a atividade de escrita e é utilizada para variados estudos, não apenas para fins da lingüística, textual e estudo cognitivo da escrita, mas também em aplicações sobre desenvolvimento de aprendizagem de línguas, alfabetização e didática de línguas. A vantagem na utilização de *keyloggers* está relacionada a possibilidade de captura de informações inseridas de forma discreta.

Segundo Ciampa (2009), um *keylogger* pode ser um dispositivo físico ou um programa que monitora cada tecla que um usuário pressiona no teclado de um computador. Na medida em que as teclas são pressionadas, estas são gravadas em um arquivo de texto. Após isto, estas informações podem ser recuperadas pelo agressor ou então transmitidas para uma estação remota. Com as informações recebidas, o agressor pode então procurar alguma informação útil no texto capturado, tais como, senhas, números de cartão de crédito, ou informações pessoais. Na forma de dispositivos físicos, um *keylogger* é um pequeno aparelho que é instalado entre o conector do teclado e a porta de teclado do computador, como mostrado na Figura 5.2. Neste caso, após os dados terem sido capturados pelo dispositivo, é necessário a remoção do mesmo para visualização do conteúdo capturado. Como este dispositivo trabalha

interceptando informações entre dois dispositivos físicos, não existe um meio de sua presença ser detectada virtualmente.

Figura 5.2 – Dispositivo *keylogger* instalado entre o conector do teclado e a respectiva porta localizada no computador.
Fonte: http://securitysearchblog.blogspot.com.

Existem ainda os *keylogger* do tipo "kernel", e como o próprio nome sugere, buscam penetrar no código do sistema operacional. Utilizam suas próprias rotinas para substituir trechos do sistema operacional, o que os torna praticamente imunes à detecção de programas antivírus e similares, pois depois de instalados passam a fazer parte do próprio sistema operacional.

No presente trabalho optou-se pelo desenvolvimento de um *keylogger* do tipo software devido a seu nível moderado de implementação e devido ao fato de que, em comparação com os outros tipos de *keylogger*, apresenta certas vantagens. A utilização de *keyloggers* do tipo físico foi considerada inviável para aplicação neste trabalho, pois seria necessária a utilização de diversos dispositivos, um para cada computador em que fosse desejada a captura dos dados, e também pelo esforço necessário para a centralização de todos os dados capturados dos computadores, que teria de ser manual, ou seja, seria necessário o recolhimento de todos os dispositivos instalados após a coleta. O tipo de *keylogger* "kernel" para também foi desconsiderado aplicação neste trabalho, pois apresenta um alto nível de dificuldade para sua implementação, por se tratar de aplicativos que substituem o próprio código do sistema operacional.

A ferramenta *keylogger* desenvolvida não tem intenção alguma de capturar dados sigilosos e, por este motivo, não foi desenvolvida para capturar caracteres numéricos, tendo

em vista que a maioria das senhas utilizadas atualmente necessita de caracteres numéricos, como é o caso de senhas bancárias. Sendo assim, um programa *keylogger* pode ser utilizado tanto para fins legais quanto para ilícitos.

5.1.1 Fundamentos jurídicos

No Brasil, existe uma carência em relação a leis apropriadas para lidarem com crimes eletrônicos. A falta destas leis faz como que o mesmo se torne um verdadeiro paraíso para todo tipo de invasão e manipulação ilícita de dados. As punições utilizadas no país são baseadas em leis que se aproximam da situação do crime eletrônico. Grande parte dos casos resolvidos pelas autoridades nacionais é relativa a casos de pirataria e pedofilia, e não invasão e "hackeamento" de sistemas. (ULBRICH, VALLE. 2003).

Diversos projetos de lei já foram formulados visando a criação de leis adequadas aos crimes eletrônicos, porém nenhuma passou pelo processo de aprovação. Segundo Ulbrich e Valle (2003) isto se deve ao fato do processo de aprovação ser muito lento, e á total ignorância dos parlamentares em relação ao assunto.

O projeto de lei mais conhecido para crimes de informática foi proposto em 2005, pelo senador Eduardo Azeredo, que visava tipificar e criminalizar diferentes tipos de ação criminosa em redes privadas ou públicas de computadores. O projeto estabelecia punição para crimes eletrônicos como criação e a propagação de vírus, *phishing*[1], invasões de redes, acesso e a divulgação indevida de dados e pedofilia. Propunha também que os provedores deveriam armazenar dados de acesso de seus usuários por 3 anos e estabelecia a criação de equipes de combate ao cibercrime (COMPUTERWORLD, 2008). Porém, em setembro de 2008 o projeto foi dado como arquivado e descontinuado.

Deste modo, atualmente, a única forma de julgamento para crimes virtuais é a localização de alguma lei que seja aplicável ao crime eletrônico. Analisando-se um crime relacionado a violação ou interceptação de *emails*, por exemplo, o mesmo pode ser aplicável

[1] Forma de fraude eletrônica, caracterizada por tentativas de adquirir informações sigilosas, tais como senhas e números de cartão de crédito, ao se fazer passar como uma pessoa confiável ou uma empresa enviando uma comunicação eletrônica oficial, como um correio ou uma mensagem instantânea.

ao artigo 151 do código penal, que é o crime de violação de correspondência. Podemos equiparar a correspondência eletrônica à correspondência tradicional, tendo em vista que a própria conversação telefônica entre pessoas também são tuteladas pelo artigo 151, e que a Internet, neste aspecto, é apenas uma evolução dos meios de comunicação (PINHEIRO. 2006).

Levando-se em consideração o fato de que a violação ou interceptação de *emails* podem ser consideradas um crime, mais especificadamente, um crime contra a inviolabilidade de correspondência, podemos considerar o uso de aplicativos *keylogger* também um crime do mesmo gênero, pois o mesmo também acessa indevidamente conteúdo de correspondência fechada, ou dirigida a outrem. Porém, segundo Pinheiro (2006), existem exceções em que empresas, por exemplo, são liberadas a rastrear *email* corporativo de empregados para obter provas de demissão justa causa. Neste ponto de vista, pode-se afirmar que, caso uma empresa desejasse utilizar *keyloggers* para monitorar o que seus funcionários digitam no computador, em relação ao ramo da empresa, é muito provável que a empresa conseguiria a liberação da justiça para implantação da ferramenta.

Segundo Miranda (2010), a legislação pátria é omissa quanto à possibilidade ou não do monitoramento de *e-mail* e *internet* pelo empregador. Diante desta discussão, conflitos surgem, de um lado à intimidade do empregado e de outro a propriedade do empregador, sendo assim, o direito á intimidade do empregado não poderá ser aplicado em sua plenitude, pois prevalecerá o direito à propriedade do empregador. Atualmente, o monitoramento de *email* corporativo e *internet* vêm sendo admitidos pelos tribunais do país, já que visam tutelar também a imagem da empresa e resguarda-la da responsabilidade de danos que possam ser causados por seu empregado à terceiros.

A monitoração do empregado em relação ao uso da tecnologia disponibilizada pela organização pode ser feita, desde que esteja previsto e que seja de conhecimento do mesmo. Pesquisas recentes apontam para a realidade e demonstram o mau uso desses meios. Há uma grande quantidade de colaboradores que acessam desde sites pornográficos até praticam pedofilia pela *internet*. O fato é que a atitude de monitoração das informações do empregado por parte do empregador está totalmente correta, pois é ele quem fornece os instrumentos e

todo o ambiente laboral para facilitação do trabalho. Assim, a má utilização dos recursos poderá servir como motivo para dispensa por justa causa (MIRANDA, 2010).

Levando-se em consideração que a monitoração de *emails* e conteúdo acessado através da *internet* é possível perante a lei pelo empregador, podemos afirmar que possivelmente uma ferramenta *keylogger* também seria identificada da mesma maneira. É importante que o empregador deixe bem claro para o empregado que todas as informações estão sendo monitoradas, em muitos casos é recomendado que o empregador formule um documento formal que simboliza o consentimento do empregado, para evitar futuras incomodações.

5.2 Aquisição de dados

Este capítulo, como um todo, requer que a base de dados já tenha sido criada, a partir de dados capturados por *keyloggers,* para que possa ser feita a análise proposta relacionada aos mesmos. Sendo assim a primeira providência a ser tomada foi o desenvolvimento de uma aplicação *keylogger*, responsável por capturar e manter um histórico de todas as teclas digitadas em um computador. Deste modo, esta aplicação pôde ser utilizada para criação da base de dados necessária. A aplicação *keylogger* foi desenvolvida em linguagem de programação Delphi, para execução em sistemas operacionais da plataforma Windows. Neste trabalho a aplicação foi denominada como Key Monitor.

Houve a necessidade de desenvolvimento de uma segundo ferramenta, denominada Key Monitor Server, com o objetivo de aquisição remota dos arquivos nos computadores onde o Key Monitor foi instalado. As duas ferramentas serão brevemente descritas a seguir a respeito de seu funcionamento.

5.2.1 Ferramenta Key Monitor

Como dito anteriormente, o objetivo principal da ferramenta Key Monitor é capturar e armazenar temporariamente quais teclas foram pressionadas no computador. No presente trabalho não existe a necessidade de captura de todas as teclas do teclado, sendo assim, serão capturadas somente teclas que referenciem caracteres alfabéticos, ou seja, números e símbolos serão desconsiderados. A aplicação funciona sem interação com a interface do usuário, como

um processo oculto, para que o mesmo não identifique sua presença e que não interfira na utilização por parte do usuário. Conforme citado, o aplicativo deverá manter um histórico das teclas pressionadas. Para isso o mesmo faz uso de arquivo de texto simples. O arquivo gerado ainda mantém outros dados além das teclas pressionadas, tais como: título da janela, data/hora, executável relacionado; todos estes referentes às teclas pressionadas. Estas informações podem ser utilizadas, caso haja necessidade, para uma análise mais aprofundada dos dados, como opções de filtragem, por exemplo. O diagrama de classes referente à esta ferramenta por ser visualizado na Figura 5.3.

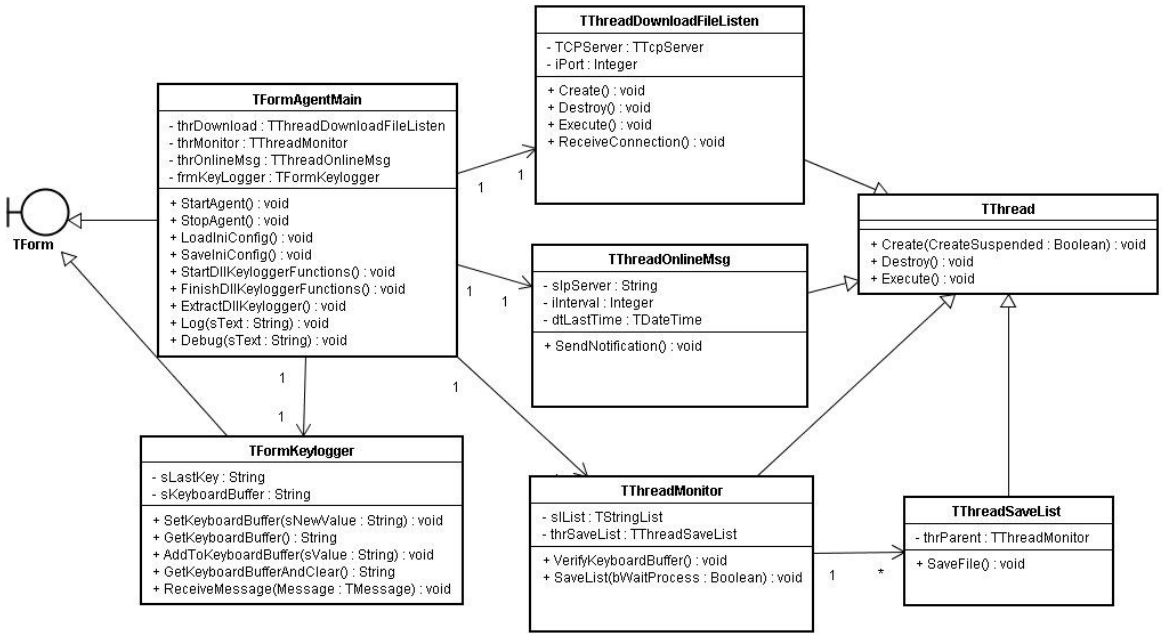


Figura 5.3 – Diagrama de classes referente à ferramenta Key Monitor.
Fonte: Autor.

A verificação de execução do Key Monitor somente pode ser feita através do Gerenciador de tarefas do Windows, como ilustra a Figura 5.4.

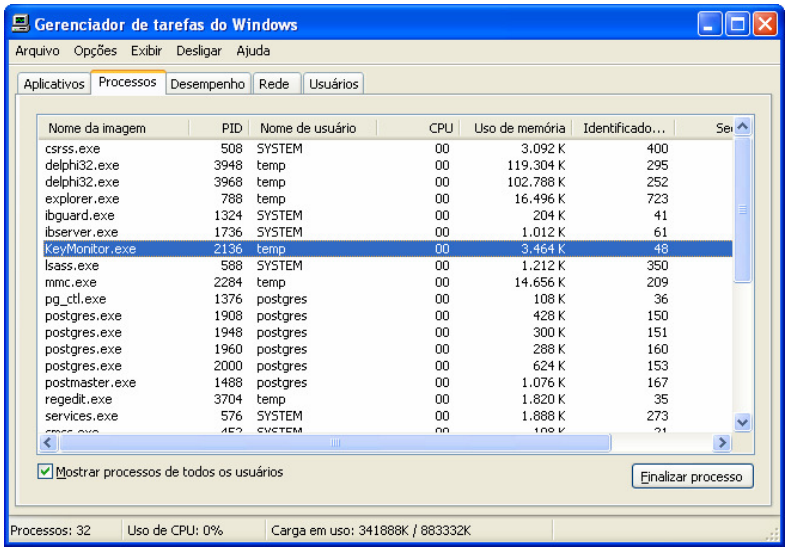


Figura 5.4 – Key Monitor em execução sem interação com a interface do usuário.
Fonte: Autor.

Para que o *Key Monitor* esteja devidamente instalado no computador, é necessário, em sua primeira utilização, que o mesmo seja instalado. Para isso o programa deve ser executado com o parâmetro "-install". Com isso, a pasta padrão do programa será criada automaticamente e o programa será registrado para que seja executado juntamente com a inicialização do sistema operacional. Após a instalação o programa já é executado automaticamente. O diretório padrão do Key Monitor encontra-se em "C:\Key Monitor\", onde é possível verificar a existência de outros arquivos, conforme ilustra a Figura 5.5.

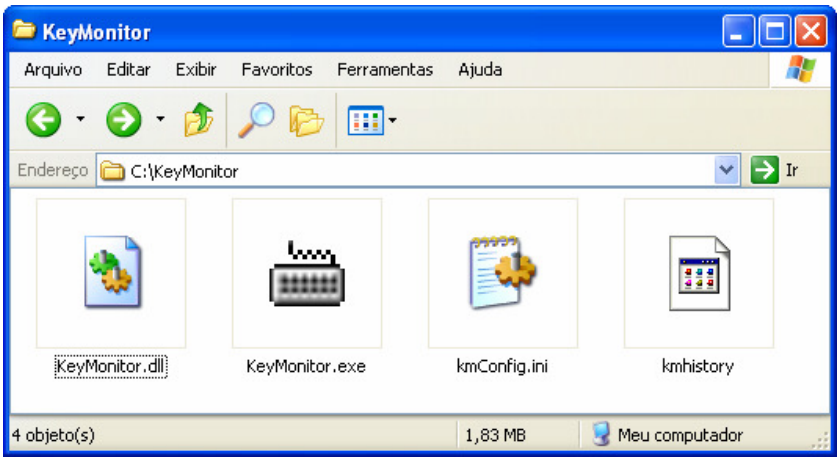


Figura 5.5 – Diretório de instalação do Key Monitor.
Fonte: Autor.

Como pode-se visualizar na Figura 5.5, quatro arquivos são criados no diretório com a execução do aplicativo, sendo estes: KeyMonitor.dll, que é a DLL responsável pelo monitoramento de teclas e consequentemente, o envio de mensagens ao aplicativo principal informando qual tecla foi pressionada; kmConfig.ini, que é o arquivo responsável pela configuração de endereço do servidor do Key Monitor, ou seja, o computador para o qual o aplicativo irá se reportar e se identificar; o arquivo kmHistory, que é o arquivo que guarda todo o histórico de teclas pressionadas no computador, este arquivo é mantido sob um nível de criptografia quando armazenado em disco, não podendo ser visualizado seu conteúdo; e o arquivo KeyMonitor.exe, que é a aplicação principal, responsável por receber mensagens da DLL citada anteriormente referentes a teclas pressionadas e pela manutenção do arquivo kmHistory.

A única configuração necessária para que o Key Monitor esteja em pleno funcionamento está relacionada ao endereço do computador em que a ferramenta Key Monitor Server está sendo executada. Esta configuração pode ser acessada através do arquivo chamado kmConfig.ini. Ao iniciar, o aplicativo Key Monitor verifica a existência deste arquivo, na pasta corrente, para que possa configurar o endereço da aplicação servidora. A Figura 5.6 ilustra a sintaxe do arquivo de configuração. O valor do campo “IP_ADDR” indica o IP do computador que se encontra a aplicação servidora. O campo “INTERVAL_NOTIFY” está relacionado ao intervalo de tempo, em minutos, no qual o computador irá se reportar como ativo para a aplicação servidora, porém quando não definido, é definido automaticamente para 1 minuto.

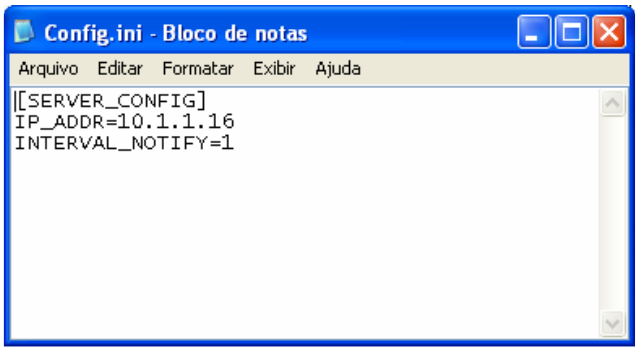


Figura 5.6 – Arquivo de configuração do Key Monitor.
Fonte: Autor.

Tendo em vista que o aplicativo Key Monitor estará em constante funcionamento em diversos computadores da rede, dependendo do ambiente computacional em que o mesmo estiver inserido, torna-se inviável a aquisição manual do arquivo gerado pelo mesmo em cada computador. Sendo assim, foi necessário o desenvolvimento de outra aplicação, esta responsável pela aquisição dos arquivos gerados pelo Key Monitor, denominada Key Monitor Server.

5.2.2 Ferramenta Key Monitor Server

Esta segunda aplicação tem como funcionalidades principais: a aquisição dos arquivos com histórico de teclas pressionadas de cada computador, utilizando-se de protocolos de rede; e o armazenamento de uma lista, esta atualizada constantemente, que indica quais computadores da rede estão com o Key Monitor devidamente instalado e com comunicação ativa com o aplicativo servidor. A Figura 5.7 ilustra o diagrama de classes referente a esta ferramenta.

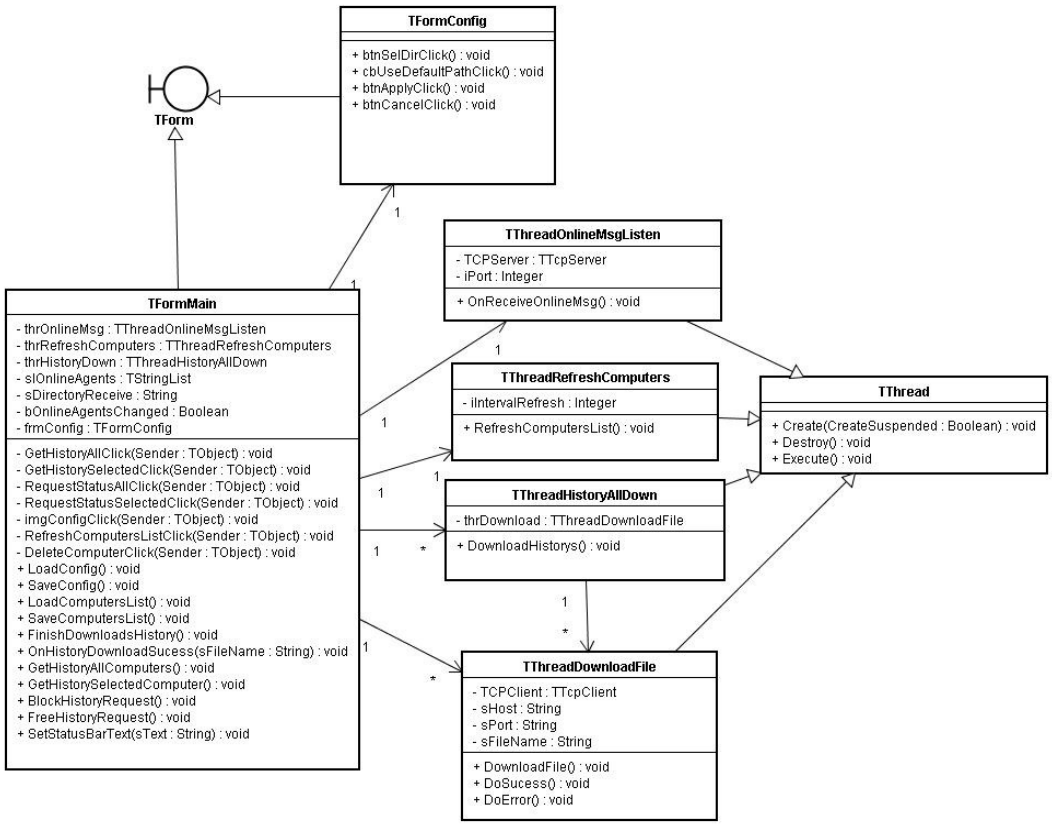


Figura 5.7 – Diagrama de classes referente à ferramenta do Key Monitor Server.
Fonte: Autor.

A interface do aplicativo Key Monitor Server é apresentada na Figura 5.8.

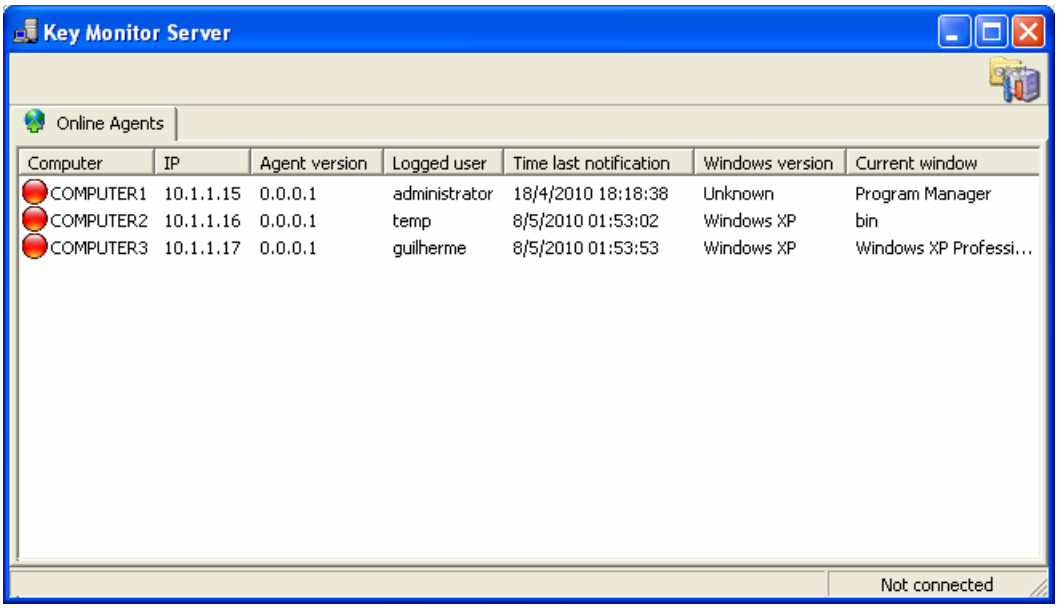


Figura 5.8 – Interface do aplicativo Key Monitor Server, responsável pela aquisição de arquivos nos computadores da rede.
Fonte: Autor.

A lista que podemos visualizar na Figura 5.8 representa os computadores que estão cadastrados na aplicação servidora. Para que um computador esteja cadastrado é necessário que o Key Monitor esteja devidamente configurado em relação ao endereço da rede do computador onde a aplicação servidora está sendo executada.

A partir do momento em que todos os computadores estiverem cadastrados na lista da ferramenta Key Monitor Server é possível realizar o procedimento responsável por capturar os arquivos referentes ao histórico de teclas pressionadas, criado pelo Key Monitor em cada computador. Ao ativar o procedimento a aplicação irá percorrer todos os computadores cadastrados, e para cada um destes realizará a conexão no computador remoto e fará um *download* do arquivo existente no mesmo. Ao término do *download*, em caso de sucesso, o arquivo é apagado no computador remoto para que não haja duplicidade de informações. A transferência do arquivo é assegurada por criptografia, e somente quando o arquivo é recebido pela aplicação servidora, é realizada descriptografia do mesmo, permitindo assim sua visualização.

Nas opções da aplicação servidora é possível definir a pasta em que serão gravados os arquivos recebidos pelo Key Monitor. Os arquivos são gravados em relação ao nome do computador onde o arquivo foi gerado. Automaticamente, caso já exista algum arquivo de determinado computador e um novo arquivo for recebido, com dados novos, o conteúdo do novo arquivo simplesmente será adicionado ao arquivo já existente, ou seja, sempre existirá somente um arquivo para cada computador. A Figura 5.9 ilustra a tela de configuração de diretório, que se encontra na aplicação servidora.

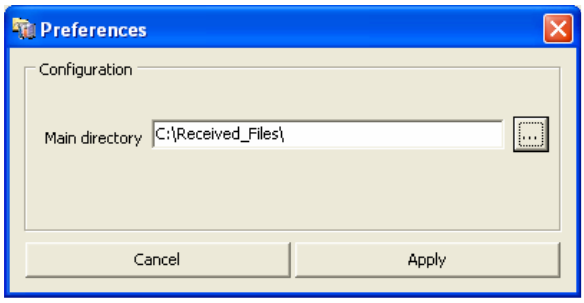


Figura 5.9 – Tela de configuração de pasta da ferramenta Key Monitor Server.
Fonte: Autor.

Importante ressaltar que, caso um computador esteja na lista de computadores cadastrados, porém não estiver com *status online*, o mesmo será desconsiderado no procedimento responsável por capturar os arquivos nos computadores da rede que estiverem executando o Key Monitor. A Figura 5.10 ilustra a solicitação de execução do processo de *download* dos arquivos de cada computador.

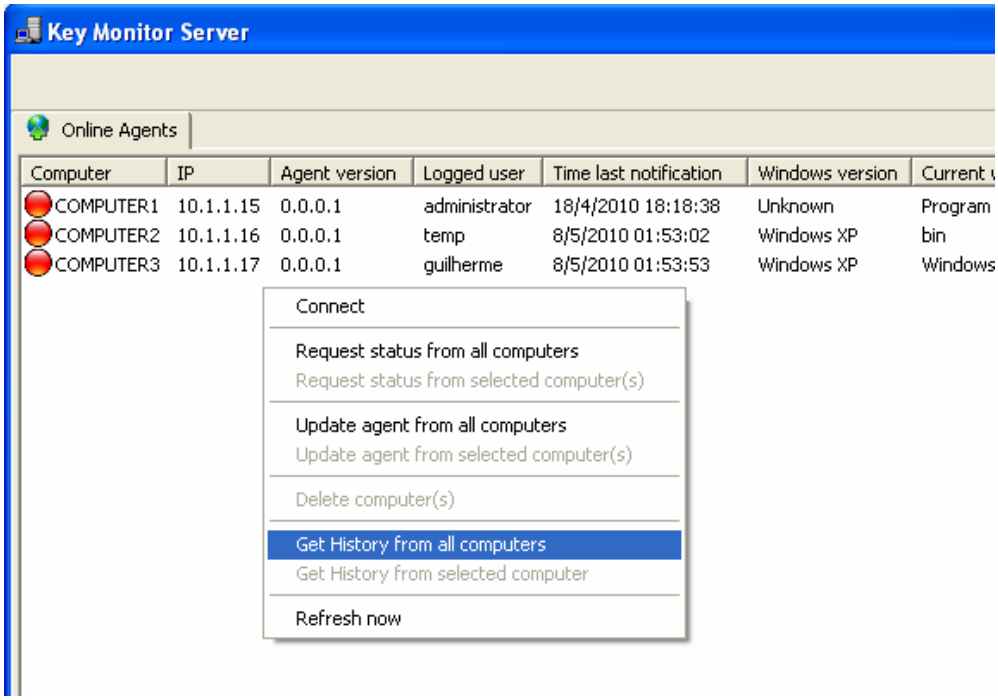


Figura 5.10 – Solicitação de arquivos com historio de teclas pressionadas
Fonte: Autor.

Após a solicitação dos arquivos de histórico de teclas pressionadas, inicia-se um processo de solicitação individual para cada computador que estiver cadastrado na lista do Key Monitor Server. Esta solicitação baseia-se, na realidade, em uma conexão direta com cada computador remoto, para que o arquivo em questão seja transferido. Conforme o procedimento transcorre, a interface do aplicativo indica em qual computador da lista a transferência de arquivo está sendo realizada. Enquanto isso é possível visualizar os arquivos que estão sendo recebidos visualizando-se o diretório, que foi previamente configurado, onde o usuário definiu para que os arquivos fossem salvos. Quando todas as solicitações e transferências tiverem sido concluídas será indicado que o processo foi finalizado em um texto situado na parte inferior da interface da ferramenta.

Após o recebimento dos arquivos referentes ao histórico de teclas pressionadas, é possível visualizar seu conteúdo, pois agora o mesmo está livre de qualquer tipo de criptografia, isto a aplicação Key Monitor Server se encarrega de fazer assim que a transferência do mesmo é concluída. Sendo assim, a Figura 5.11 mostra um exemplo de como se encontra a estrutura do arquivo gerado pelo Key Monitor. Analisando esta figura, podemos

perceber que, para cada registro, existem quatro propriedades separadas por um delimitador, sendo: título da janela, nome do programa que estava sendo executado, data e hora e o conteúdo que foi digitado enquanto a janela permanecia a mesma. O delimitador neste arquivo de histórico é indicado pelos três caracteres "@®@", que separam os campos. Importante destacar que a cada troca de janela uma nova linha começa a ser registrada em relação ao monitoramento de teclas, assim, o conteúdo fica mais organizado e permite ainda o uso destas informações adicionais para novas formas de análise. Na seção 4, referente ao pré-processamento, serão detalhadas as rotinas que fazem uso deste arquivo.

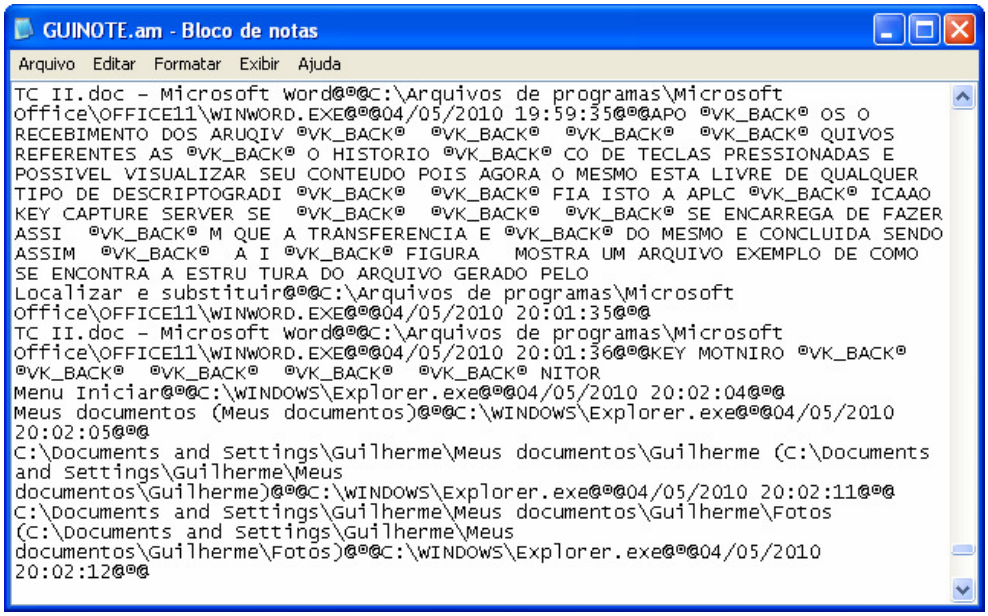


Figura 5.11 – Exemplo de estrutura de arquivo de histórico.
Fonte: Autor.

A seguir é apresentada, na Figura 5.12, um modelo que ilustra o funcionamento e relação entre as ferramentas desenvolvidas, Key Monitor e Key Monitor Server. A ilustração demonstra uma transferência de arquivo de histórico de teclas pressionas em execução, e outros computados da rede, que estão executando o Key Monitor, estão aguardando para atender a requisições do Key Monitor Server.

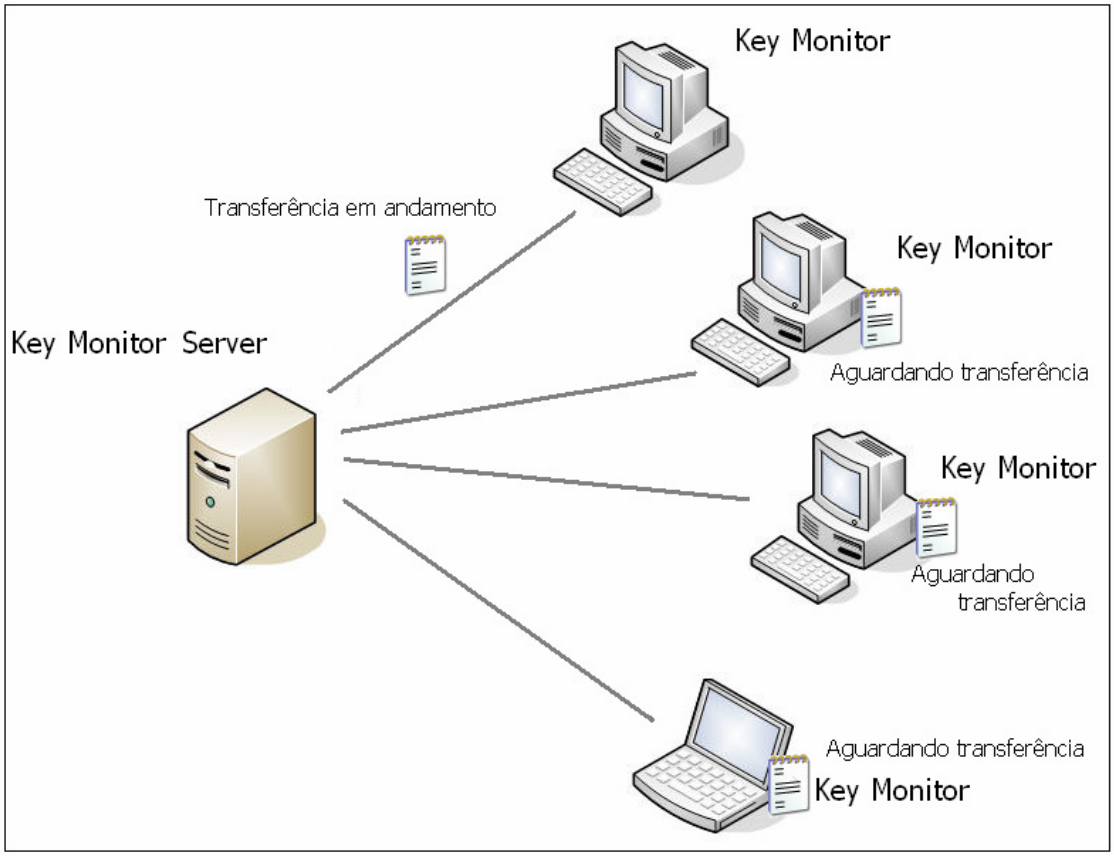


Figura 5.12 – Funcionamento das ferramentas desenvolvidas.
Fonte: Autor.

Com a finalização do processo de transferência dos arquivos de histórico de teclas pressionadas de todos os computadores, tem-se a base de dados necessária construída. Assim, a próxima etapa do trabalho pode ser realizada, a mesma refere-se ao pré-processamento.

5.3 Pré-processamento

Conforme citado no capítulo 1, facilmente encontram-se características como inconsistência, inexistência e incompletude nos dados. E uma das razões para que dados com estas características tenham sido gerados, está relacionada a erros humanos. No presente trabalho está é, de fato, uma característica encontrada nos dados, levando-se em consideração o método de aquisição dos mesmos, ou seja, através de *keyloggers*. As fases relacionadas à etapa de Pré-processamento são: tratamento especial de *backspaces*, verificação de ortografia, remoção de *stopwords* e tratamentos especiais.

Para realização desta etapa foi necessário o desenvolvimento de uma nova ferramenta, denominada Data Analiser. Esta ferramenta engloba, além de rotinas referentes ao pré-processamento, as opções de treinamento de classes e visualização de resultados, que serão abordados nas próximas seções. A Figura 5.13 apresenta o diagrama de classes elaborado referente à ferramenta Data Analiser.

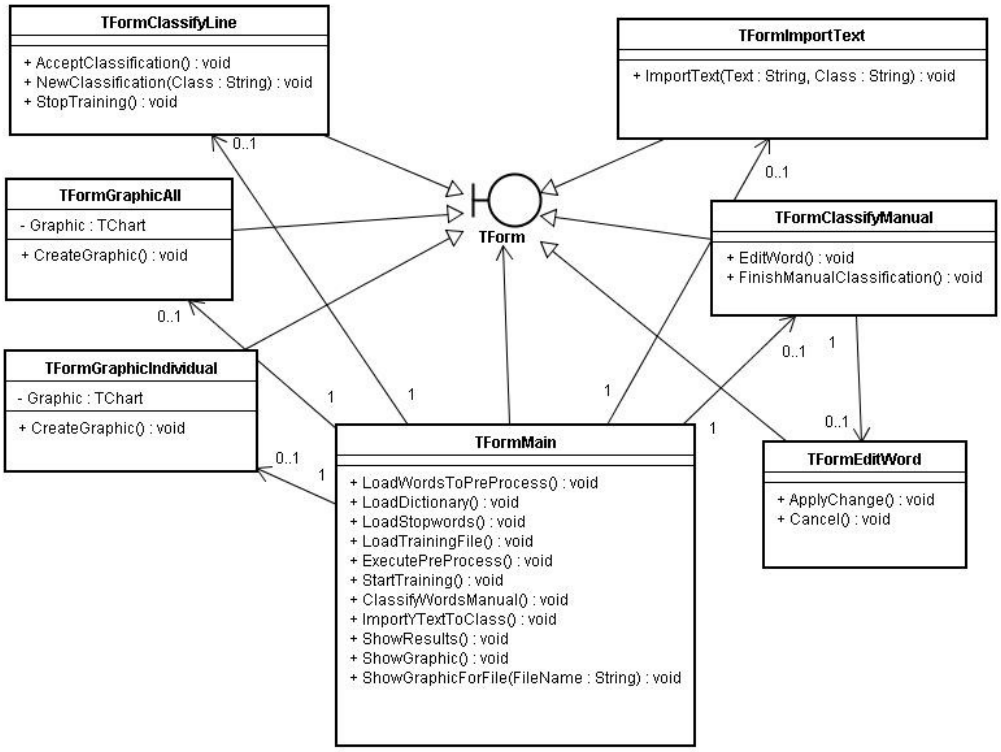


Figura 5.13 – Diagrama de classes da ferramenta Data Analiser.
Fonte: Autor.

A figura 5.14 ilustra a interface da ferramenta, durante a execução do processo de pré-processamento.

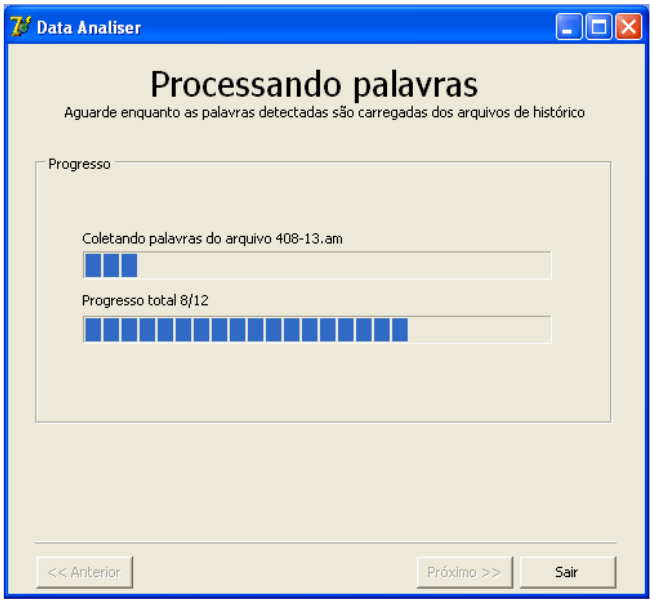


Figura 5.14 – Pré-processamento em execução na ferramenta Data Analiser.
Fonte: Autor.

A seguir são detalhadas as rotinas referentes a etapa de pré-processamento que foram implementadas na ferramenta Data Analiser.

5.3.1 Tratamento especial de *backspaces*

Para que houvesse um melhor aproveitamento de todas as teclas pressionadas, e visando um entendimento maior sobre os caracteres agrupados, ou seja, as palavras em si, foi implementado na ferramenta *keylogger* a utilização do indicador "®VK_BACK®" que tem como objetivo caracterizar o uso da tecla *backspace*. Fez-se necessário a utilização deste identificador para que fosse possível a criação de uma rotina específica para recuperação de palavras que possam ter sofrido alterações devido a utilização do *backspace*. É muito comum, ao utilizar um editor de texto, por exemplo, que a tecla *backspace* seja pressionada para correção de uma palavra que está sendo digitada. Sendo assim, esta etapa para tratamento de *backspaces* foi inserida no pré-processamento dos dados com a intenção de aumentar o número de palavras válidas.

Na Figura 5.11 da seção anterior, que ilustra um exemplo de arquivo gerado pelo Key Monitor, pode-se visualizar um número razoável de palavras ortograficamente corretas, porém

existe também uma quantidade muito grande de identificadores de *backspace*. Isto é justificado levando-se em consideração que erros na digitação são muito comuns no dia-dia. Assim é possível que haja um controle posterior à captura de teclas para uma diminuição dos erros de digitação. Por exemplo, o indicador de tecla *backspace* pode ser usado da seguinte maneira: ao encontrar o indicador em uma linha de texto, é necessário simplesmente remover um caractér anterior, e remover, obviamente, o indicador. Assim o texto da linha em questão irá manter o comportamento padrão que a tecla *backspace* oferece em um editor de texto. Infelizmente, muitas vezes pode ocorrer de não existir nenhuma relação entre a tecla anterior e a tecla *backspace*.

A seguir são apresentadas duas figuras: a Figura 5.15, que apresenta o texto capturado em um registro gravado pelo *Key* Monitor, e a Figura 5.16, que demonstra o mesmo texto da Figura 5.15 após ser submetido ao processo de remoção de *backspaces* do texto.

APO ®VK_BACK® OS O RECEBIMENTO DOS
ARUQIV ®VK_BACK® ®VK_BACK® ®VK_BACK®
®VK_BACK® QUIVOS REFERENTES AS ®VK_BACK®
O HISTORIO ®VK_BACK® CO DE TECLAS
PRESSIONADAS E POSSIVEL VISUALIZAR SEU
CONTEUDO POIS AGORA O MESMO ESTA LIVRE DE
QUALQUER TIPO DE DESCRIPTOGRADI
®VK_BACK® ®VK_BACK® FIA

Figura 5.15 – Texto capturado sem tratamento relacionado a backspaces.
Fonte: Autor.

APOS O RECEBIMENTO DOS ARQUIVOS REFERENTES
AO HISTORICO DE TECLAS PRESSIONADAS E
POSSIVEL VISUALIZAR SEU CONTEUDO POIS AGORA
O MESMO ESTA LIVRE DE QUALQUER TIPO DE
DESCRIPTOGRAFIA

Figura 5.16 – Texto da Figura 5.15 após tratamento relacionado a backspaces.
Fonte: Autor.

5.3.2 Verificação de ortografia

Para que esta fase da etapa de pré-processamento pudesse ser realizada foi necessária a utilização de um dicionário em português, para que tivéssemos um meio de comparação entre as palavras detectadas e as palavras realmente existentes. O dicionário escolhido foi o Ispell, em se tratar de um dicionário *freeware*. O mesmo funciona em sistemas operacionais UNIX, mas na realidade o que se utilizou deste dicionário foram somente o arquivos que contém a listagem dos termos, não sendo necessária a utilização do aplicativo em si.

Houve a necessidade de uma conversão em relação aos arquivos do Ispell para que os dados fossem adequados a realidade dos *keyloggers*. Os dados advindos dos *keyloggers* não possuem tratamento em relação a acentuação e são apresentados sempre em letras minúsculas. Sendo assim uma ferramenta bastante simples foi desenvolvida, na linguagem de programação Delphi, para realizar este tipo de conversão e manter um formato compatível. A Figura 5.17 ilustra a interface da ferramenta em questão.

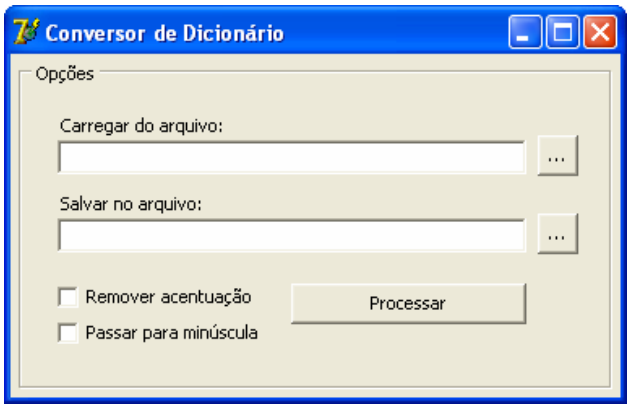


Figura 5.17 – Ferramenta utilizada para conversão de dicionários do Ispell.
Fonte: do Autor.

Nesta fase é necessária a validação de todas as palavras detectadas pelo aplicativo *keylogger* em relação as palavras cadastradas no dicionário. As palavras que não forem encontradas no dicionário serão descartadas. Tendo em vista que a língua portuguesa tem uma grande quantidade de palavras e variações, e que muitas das palavras capturadas que poderiam ser importantes na análise poderiam acabar sendo simplesmente descartadas, optou-se pela

utilização de um recurso que pudesse adicionar, de forma rápida e segura, novas palavras ao dicionário.

Esta opção foi incluída na ferramenta responsável pela mineração de textos. A mesma inclui rotinas de pré-processamento, treinamento de palavras e resultados obtidos, porém a será detalhada mais adiante. A Figura 5.18 ilustra esta etapa no aplicativo dentro do processo de mineração de textos.

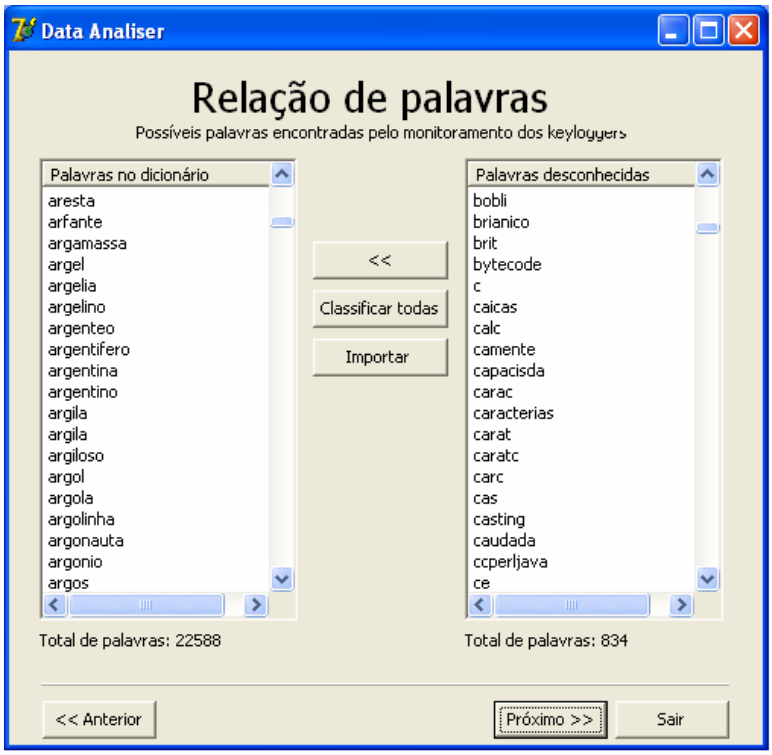


Figura 5.18 – Etapa da ferramenta de pré-processamento para validação de palavras.
Fonte: do Autor.

Como pode-se observar esta etapa apresenta duas listas: uma traz uma relação de todas as palavras existentes no dicionário, e a outra mostra uma lista de palavras, detectadas pelo Key Monitor, que não foram encontradas no dicionário. Sendo assim é possível, facilmente, adicionar uma palavra ao dicionário para que a mesma não seja descartada. Isto pode ser feito simplesmente selecionando-se as palavras da lista “Palavras desconhecidas” e pressionando o botão “<<”. Assim estas palavras serão cadastradas no dicionário fazendo com que em próximas verificações as mesmas também sejam consideradas como válidas.

Existe ainda uma outra opção para que palavras desconhecidas sejam adicionadas ao dicionário. Na etapa apresentada na Figura 5.18, é possível perceber a existência de um botão denominado “Importar”. Ao utilizar este recurso é dada a possibilidade de adição de palavras pelo usuário através de um simples trecho de texto. Ao informar o texto para adição, a ferramenta executa uma rotina para listagem de todas as palavras existentes no mesmo, e, após separar as palavras, estas são adicionadas ao dicionário. Porém, é necessária uma atenção maior na utilização deste recurso para que palavras com algum erro gramatical não sejam adicionadas. Caso alguma das palavras de texto já exista no dicionário, a ferramenta irá desconsiderá-la automaticamente.

5.3.3 Remoção de stopwords

Dentro da etapa de pré-processamento esta fase se torna imperceptível, pois seu objetivo é simplesmente retirar palavras pré-estabelecidas da lista de palavras válidas detectadas pelo *keylogger*. Estas palavras se encontram listadas em um arquivo de texto. O arquivo que contém a listagem de *stopwords*, utilizado neste trabalho, foi disponibilizado por Stanley Loh em seu blog sobre mineração de texto. Este arquivo é carregado durante a etapa de pré-processamento pela ferramenta responsável pelo pré-processamento dos dados. O arquivo contendo a lista de stopwords também teve que ser submetido à ferramenta de conversão, citada anteriormente, para que fossem removidos os acentos de todas as palavras, e assim, fosse possível a comparação com palavras detectadas. A remoção de stopwords é muito importante para que palavras sem significado sejam retiradas, pois estas poderiam até mesmo acabar influenciando no resultado final da mineração. A Figura 5.19 e a Figura 5.20 exemplificam a importância de remoção de stopwords adquirida nos dados gerados a partir da implantação realizada.

Figura 5.19 – Texto antes de ser destinado ao processo de remoção de *stopwords*.
Fonte: do Autor.

PYTHON LINGUAGEM INTERPRETADA
COMPILADA INTERPRETADOR EXECUTADA
SISTEMA OPERACIONAL

Figura 5.20 – Texto da figura 5.19 após a remoção de *stopwords*.
Fonte: do Autor.

Em muitos casos, pode-se afirmar que a maioria das palavras em um texto são na realidade, *stopwords*. As figuras apresentadas, por exemplo, demonstram a transformação ocorrida em uma frase através do processo de remoção destas palavras.

5.3.4 Tratamentos especiais

Após o processo de implantação das ferramentas no ambiente de ensino e, consequentemente, o sucesso na aquisição dos dados a serem utilizados pelo trabalho, tudo indicava que os dados estavam de acordo com o esperado. Porém após uma visualização mais detalhada dos dados obtidos, percebeu-se que todas as palavras que continham o caractér “ç” apresentavam a ausência do mesmo. Este problema ocorreu devido a um erro da preparação da ferramenta *keylogger*. Na programação de um *keylogger*, em determinado momento, é possível a detecção de qual o código ASCII foi pressionado, para que seja a partir dele seja encontrada a letra referente. Porém em alguns teclados existe uma configuração que não está relacionada somente ao código ASCII, e como o ambiente em que a ferramenta *keylogger* foi desenvolvida tem uma configuração diferente do ambiente em que houve a implantação originou-se este problema.

Deste modo houve a necessidade de uma nova tarefa dentro da etapa de pré-processamento para que todas as palavras adquiridas fossem verificadas e que, assim, o problema diagnosticado fosse resolvido. Optou-se pela remediação do problema para que não fosse necessária uma nova implantação das ferramentas, tendo em vista que o problema encontrado poderia ser resolvido com um nível baixo de dificuldade. Esta tarefa auxiliar de pré-processamento consistiu-se, basicamente, na busca por palavras que geralmente utilizam-

se do caracter “ç”, sendo estas, na maioria dos casos, terminadas em “ão”, “ões”. Foram criadas regras relacionadas as alterações necessárias, que foram utilizadas em uma rotina criada especialmente para contornar este problema.

5.4 Mineração de dados

A classificação de textos tem como objetivo principal a associação de textos a classes previamente estabelecidas. Geralmente utilizada para classificação de documentos de texto, estes são classificados a partir de características como termos ou palavras presentes nos documentos. No presente trabalho a grande dificuldade é que não está se lidando com documentos, e sim com dados capturados através de *keyloggers*. Sendo assim, não podemos classificar todo o conteúdo capturado como se fosse um documento de texto, e sim é necessário que cada situação que está inserida nos dados seja identificada e seja classificada individualmente. Tendo em vista que seria necessário que estas “situações” fossem classificadas individualmente, a ferramenta de *keylogger* foi desenvolvida de maneira que cumprisse esta condição. Na seção anterior, foi brevemente apresentada a estrutura de um arquivo gerado pela ferramenta Key Monitor, na Figura 5.11, na qual podemos perceber que existem, além das informações de teclas detectadas, informações como titulo da janela e programa. Ou seja, a informação que aponta as teclas detectadas está relacionada ao período em que o título da janela corrente permaneceu o mesmo. O mesmo vale para a informação referente ao programa que estava sendo executado. Deste modo, a classificação será realizada em relação a cada troca de janela ou programa, ficando a critério do usuário.

Segundo Silva e Galho (2006) apud Rizzi (2000), “a categorização ou classificação de textos é uma técnica utilizada para classificar um conjunto de documentos em uma ou mais categorias existentes”. Ela é geralmente utilizada para classificar mensagens, notícias, resumos e publicações. Um sistema de categorização automática de textos geralmente é composto por duas fases: definição de categorias ou classes e enquadramento de novos documentos. A fase de definição das categorias, normalmente, é realizada em três etapas: preparação de textos, seleção de categorias predominantes e criação das classes. (SILVA, GALHO). A Figura 5.21 ilustra a fase de definição de categorias.

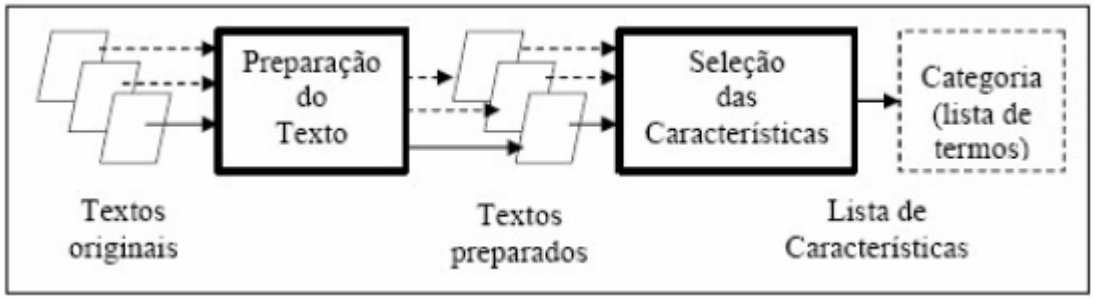


Figura 5.21 – Fase de definição das categorias.
Fonte: GALHO; SILVA, 2006, p. 2.

Basicamente, na fase de definição das categorias, documentos conhecidos e pertencentes a uma mesma classe são aplicados para a preparação do texto. Os documentos passam por uma fase de pré-processamento para que sejam removidas informações irrelevantes. Feito isso os documentos são analisados para que todas as palavras que melhor expressem as características do texto sejam encontradas, ou seja, as palavras conceitos que podem definir a classe de um documento. Em seguida é gerada uma lista de termos comum a todos os documentos. Essa lista de termos compõe o índice que representa a categoria. A Figura 5.22 ilustra a fase de categorização de novos documentos.

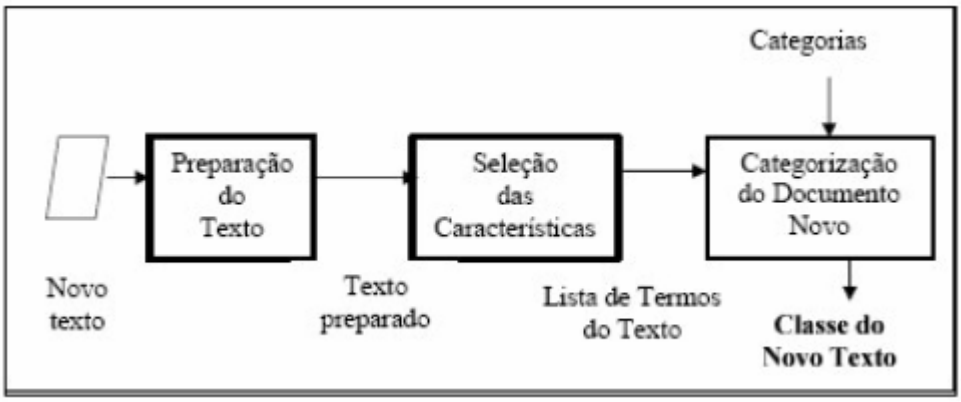


Figura 5.22 – Categorização de novos documentos.
Fonte: GALHO; SILVA, 2006, p. 3.

Sendo assim, do mesmo modo que Silva e Galho utilizaram esta metodologia de categorização para classificar os documentos em questão, neste trabalho também serão utilizadas estas 2 fases de processamento. Inicialmente será utilizado um conjunto de dados como modelo de teste, para que seja feito o treinamento de classes. Uma grande dificuldade

neste ponto está no fato de que as informações que serão adquiridas através do uso do *keylogger* não estarão sob uma organização, ou seja, todos os dados estarão misturados, não podendo ser classificado como uma classe apenas. Haverá necessidade de uma classificação em relação a cada "linha" capturada pela ferramenta. A intenção é que seja possível a classificação dos dados capturados através de *keylogger* filtrando-se os mesmos pela data, ou seja, será feita uma classificação para cada dia. Esta classificação será feita de modo probabilístico, indicando assim qual a porcentagem de cada classe baseando-se em cada dia de funcionamento do *keylogger*.

Como o propósito deste trabalho é a classificação dos dados capturados a partir de *keyloggers* em classes pré-estabelecidas, é necessário, obviamente, definir as classes que serão utilizadas. Tendo em vista que classificar os dados baseando-se somente no ambiente em que os mesmos foram gerados, ou seja, classificando os dados entre "se enquadram como produtivos de acordo com o ambiente" e "não se enquadram", seria uma divisão um tanto quanto limitada, buscou-se uma nova divisão de classes para a aplicação no trabalho.

As classes que mais se enquadraram para uso no presente trabalho, para análise de conteúdo dos dados, foram as classes utilizadas por Bassani (2009) apud Behar (2005), no desenvolvimento de um trabalho que objetivou um estudo sobre análise de trocas de mensagens em espaços de educação à distância. No mesmo foram apontados quatro eixos conceituais para a análise das interações em um ambiente virtual de aprendizagem, sendo estas:

a) epistemológico: envolve tudo o que faz referência e/ou caracteriza o processo de construção do pensamento sobre o objeto de estudo, neste caso, o conteúdo/matéria do curso;
b) tecnológico: envolve tudo o que faz referência ao gerenciamento dos aspectos tecnológicos, em relação a questões essencialmente relacionadas à tecnologia, como funcionamento/regras/lógica do sistema computacional e demais softwares de apoio, e o conhecimento necessário para a comunicação/interação e pertinência nestes ambientes (aspectos operacionais e funcionais);
c) social: tudo o que envolve o processo de construção numa coletividade, seja essa através de relações individuais ou interindividuais;

d) afetivo: caracteriza-se pela expressão de emoções, como desejos, emoções e sentimentos.

Sendo assim, pode-se perceber que a primeira classe descrita foi criada em relação direta ao trabalho desenvolvido por Bassani (2009) e que, evidentemente, não se enquadra como uma classe aceitável deste trabalho. Deste modo, esta classe foi desconsiderada e outra foi adicionada em seu lugar. Esta nova classe visa englobar todos os assuntos pertinentes ao ambiente onde os dados foram adquiridos, ou seja, dados que tenham relação direta com atividades principais que se enquadram em determinado espaço.

Com a utilização destas quatro classes definidas é provável que em alguns casos não seja possível classificar os dados em relação a uma classe somente. Ou seja, haverão casos em que os dados serão classificados em mais de uma classe. Pode ocorrer que determinados dados sejam classificados como, por exemplo, epistemológico-social ou social-afetivo.

Para que a medição de probabilidade de cada histórico, em relação às classes existentes, possa ser realizada, será necessária uma classificação prévia das palavras de forma individual. Ou seja, em determinado momento será necessária a classificação manual de todas, ou em parte, das palavras encontradas de acordo com as classes. Assim cada palavra terá seu respectivo valor em relação a cada classe existente. O Gráfico 5.1 demonstra como exemplo, a classificação de algumas palavras em relação às classes definidas.

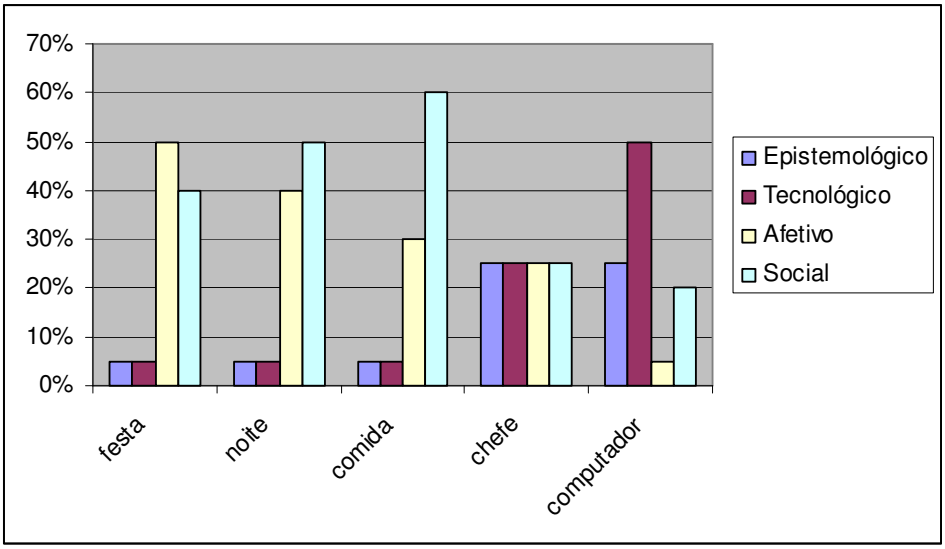


Gráfico 5.1 – Pré-classificação de palavras.
Fonte: do Autor.

O Gráfico 5.1 apresenta um exemplo de palavras classificadas de acordo com as classes de forma manual. Nela podemos perceber a presença de palavras de difícil classificação em relação às classes, mesmo em se tratando de classificação manual. Infelizmente, haverão palavras com uma dificuldade muito grande de classificação, que dependem de outras palavras para terem seu sentido explicitado corretamente.

No presente trabalho a classificação é realizada em relação a cada linha capturada pela ferramenta *keylogger*, pois em diversas situações, palavras que tem determinado sentido em uma frase podem apresentar um sentido totalmente adverso em outra. Porém, a classificação em relação a cada palavra existente é necessária, pois para que se encontre a classe correspondente a uma linha, é necessário, previamente, que cada palavra que compõe a linha esteja devidamente classificada. Sendo assim, cada palavra existente possui um valor referente a cada classe. Este valor vai de zero (0) até cem (100), indicando a pontuação que a palavra possui para cada classe. Um arquivo, chamado "Treinamento.dic", é responsável pelo armazenamento das palavras em relação a cada classe existente. A Figura 5.23 ilustra um exemplo de como as informações são mantidas no arquivo citado.

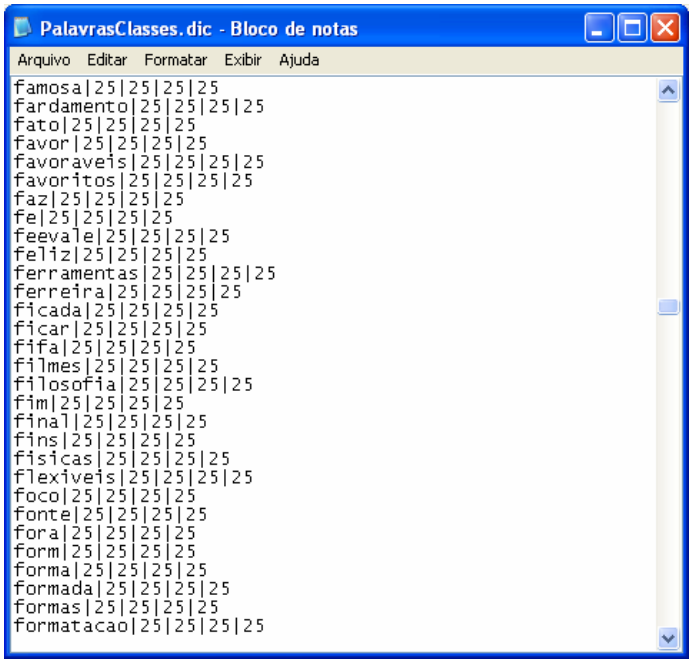


Figura 5.23 – Arquivo responsável pelo armazenamento das palavras em relação às classes.
Fonte: do Autor.

Como pode-se visualizar na Figura 5.23, cada linha do arquivo em questão armazena uma palavra e os respectivos valores de suas classes. No arquivo é utilizado o delimitador "|" para separação dos campos que são, respectivamente, palavra, pontuação da classe epistemológico, pontuação da classe tecnológico, pontuação da classe social e pontuação da classe afetivo. Por padrão, palavras que ainda não passaram pelo processo de classificação, tem os valores de pontuação das classes definidos em zero (0), sendo que o valor mínimo é zero (0) e o máximo é cem (100) para pontuação das palavras em relação às classes. Conforme as palavras passam pelo processo de classificação, os valores referentes a cada classe mudam para indicar o sentido das palavras. A partir do momento em que todas as palavras possuírem uma classificação, é possível visualizar a classificação da linha. Sendo que para isto, basta que os valores de todas as palavras sejam somados em relação a cada classe, ou seja, totalizando o valor de cada classe em relação à linha. A classe que possuir maior pontuação, baseando-se na pontuação das palavras, indica a classificação da linha.

Porém a classificação encontrada para cada linha somente pode ser visualizada após a classificação das palavras envolvidas, ou seja, após o treinamento de classes, que será detalhado a seguir.

5.5 Implantação das ferramentas

A proposta inicial do trabalho foi a de criar um método onde fosse possível a classificação de informações capturadas por um *keylogger*. Sendo assim, cabe definir-se exatamente como a implantação procederá. Inicialmente, foi citada uma possível utilização de ambientes empresariais para aquisição de dados através do uso do *keylogger* desenvolvido, porém, levando-se em considerações que muitas informações capturadas possam vir a ser confidenciais e sigilosas para a empresa em questão, esta alternativa foi descartada. Outro ponto importante, é que anteriormente havia sido citada uma possível análise dos dados capturados em relação ao foco do ambiente em que os mesmos foram adquiridos, fazendo-se uma divisão entre as informações que se enquadrariam ou não no contexto. As informações serão classificadas, na realidade, entre quatro classes diferentes, em relação à seu conteúdo, já apresentadas anteriormente.

Decidiu-se então a execução do procedimento para aquisição de dados em uma sala de aula. A proposta é de utilização do *keylogger* diante de alguma atividade proposta pelo

professor a ser realizada pelos alunos. Assim o conteúdo que estará relacionado à atividade da aula torna-se menos extenso e existe menos dificuldade para verificar se as informações capturadas se enquadram na atividade proposta pelo professor. Sendo assim, as ferramentas desenvolvidas foram levadas ao ambiente proposto e iniciou-se a implantação. A sala de aula em questão era composta por 12 alunos, cada um utilizando um computador. A ferramenta Key Monitor foi devidamente instalada em cada computador em uso para que a captura de informações fosse iniciada. Após a instalação da ferramenta Key Monitor em todos os computadores, foi possível realizar a verificação de comunicação entre estes e o computador "servidor", onde estava sendo executado a ferramenta Key Monitor Server. A Figura 5.24 demonstra como é realizada esta verificação.

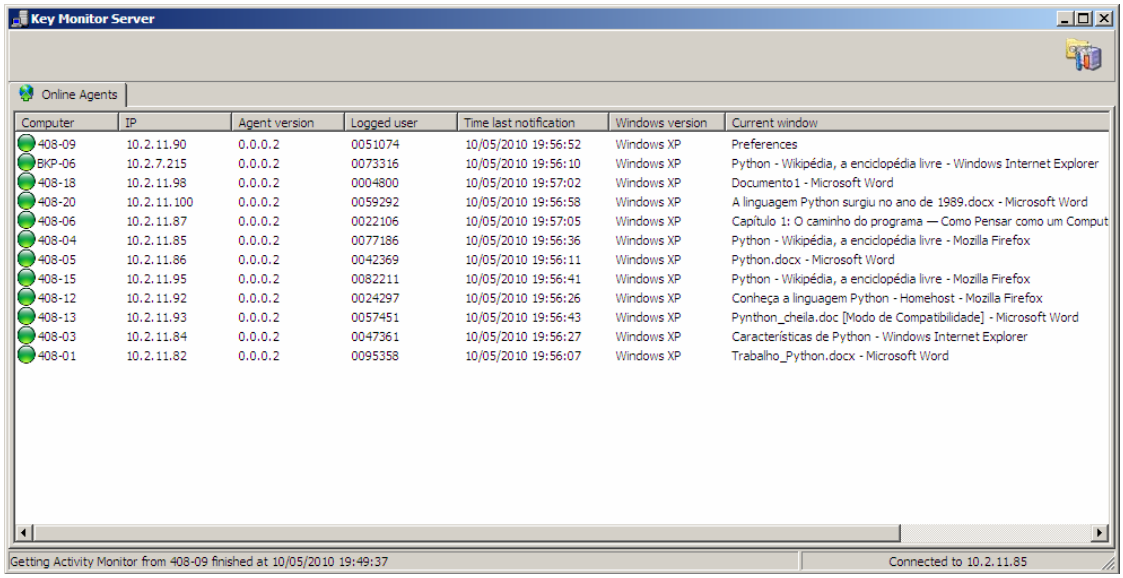
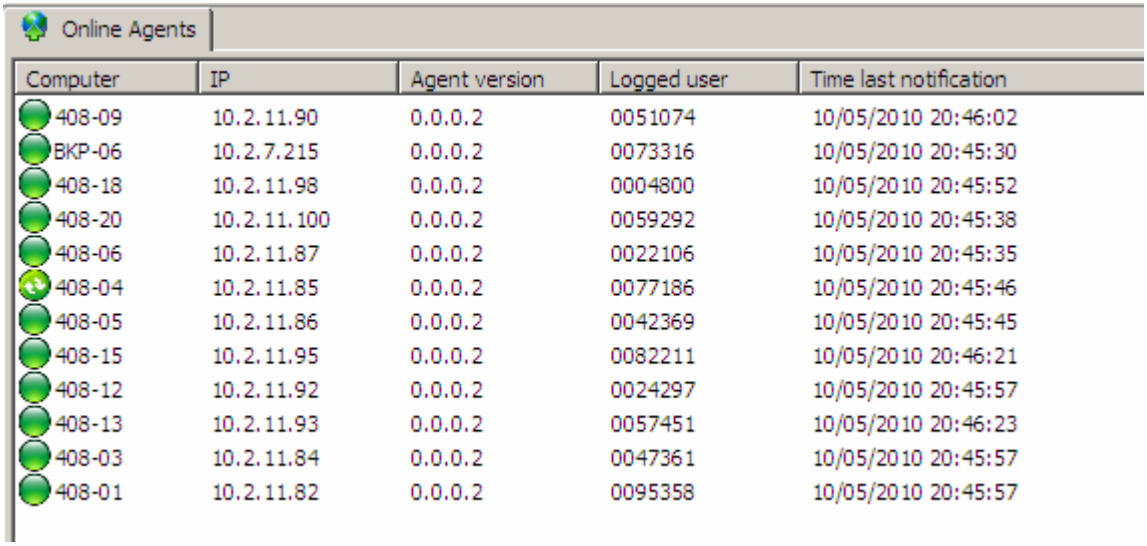


Figura 5.24 – Interface da ferramenta Key Monitor Server confirmando conexão entre computadores.
Fonte: do Autor.

Ao finalizar a implantação das ferramentas na rede, uma atividade foi passada para os alunos pelo professor. Esta baseava-se em realizar uma pesquisa sobra a linguagem de programação Python, e entrega-la no término da aula. Na Figura 5.24 é possível perceber que a tarefa já havia sido iniciada pelos alunos verificando-se a coluna "Current Window", que representa a janela corrente utilizada por cada computador, indicando a pesquisa em relação a atividade proposta.

As ferramentas foram mantidas em execução durante 1 hora e 30 minutos, e durante este período de tempo todas as informações em relação a teclas pressionadas foram capturadas e mantidas em um histórico pela ferramenta Key Monitor. Ao final do período, iniciou-se o processo de aquisição do histórico de teclas referente a cada computador. Para isso, a ferramenta Key Monitor Server baseia-se na lista de computadores existente, ou seja, computadores que estão com um meio de comunicação ativo, para que realize a conexão com cada um dos computadores e baixe os arquivos de histórico. A Figura 5.25 demonstra a execução deste procedimento em andamento, informando em qual computador a transferência do histórico está sendo realizada.


Online Agents

Computer	IP	Agent version	Logged user	Time last notification
408-09	10.2.11.90	0.0.0.2	0051074	10/05/2010 20:46:02
BKP-06	10.2.7.215	0.0.0.2	0073316	10/05/2010 20:45:30
408-18	10.2.11.98	0.0.0.2	0004800	10/05/2010 20:45:52
408-20	10.2.11.100	0.0.0.2	0059292	10/05/2010 20:45:38
408-06	10.2.11.87	0.0.0.2	0022106	10/05/2010 20:45:35
408-04	10.2.11.85	0.0.0.2	0077186	10/05/2010 20:45:46
408-05	10.2.11.86	0.0.0.2	0042369	10/05/2010 20:45:45
408-15	10.2.11.95	0.0.0.2	0082211	10/05/2010 20:46:21
408-12	10.2.11.92	0.0.0.2	0024297	10/05/2010 20:45:57
408-13	10.2.11.93	0.0.0.2	0057451	10/05/2010 20:46:23
408-03	10.2.11.84	0.0.0.2	0047361	10/05/2010 20:45:57
408-01	10.2.11.82	0.0.0.2	0095358	10/05/2010 20:45:57

Figura 5.25 – Processo de aquisição dos históricos dos computadores em andamento.
Fonte: do Autor.

No término do processo de aquisição dos dados é possível visualizar os arquivos criados na pasta selecionada como padrão, ainda no ambiente de implantação do sistema. Sendo assim, com a base de dados devidamente criada, a mesma pode ser encaminhada a etapa de pré-processamento e, em seguida, para a etapa de treinamento de classes. Por fim, será possível a visualização de resultados obtidos através da implantação do trabalho.

5.6 Treinamento de classes

A etapa de treinamento de classes visa o aprendizado da ferramenta em relação às classes em que as palavras estão enquadradas. Sendo assim, nesta etapa o usuário deverá interagir com a ferramenta verificando em que classes as linhas ou palavras detectadas foram classificadas e, caso seja necessário, apontar quais classificações estão incorretas, ou até mesmo classificar palavras de forma manual, para que o aprendizado seja realizado. O objetivo principal desta etapa é a classificação das palavras cadastradas para que estas informações possam ser utilizadas para visualização de resultados.

São disponibilizadas, na ferramenta Data Analiser, três formas de execução do processo de treinamento, que podem ser realizadas em conjunto, ou de forma individual. A primeira opção diz respeito ao treinamento usual de classes. Nesta opção o usuário deve avaliar as classificações que a ferramenta fez a em relação às linhas, indicar se a classificação está correta ou indicar uma nova classificação. Na segunda opção é permitida a classificação das palavras do dicionário de maneira manual, onde o usuário tem a possibilidade de definir a classificação de cada palavra. A terceira opção é destinada à importação de textos relativos a alguma das classes, sendo que ao indicar uma classe para o texto em questão, todas as palavras do mesmo são adicionadas ao dicionário com a pontuação definida para a classe selecionada. As três formas de treinamento serão detalhadas a seguir. A Figura 5.26 apresenta a tela de opções de treinamento.

Figura 5.26 – Tela de opções para realização do treinamento de classes.
Fonte: do Autor.

5.6.1 Treinamento de classes supervisionado

Na primeira opção de treinamento de classes, o processo faz uso da base de dados criada a partir da fase de pré-processamento, para carregar todas as linhas que serão utilizadas. O principal objetivo desta forma de treinamento de classes é apresentar para o usuário as linhas existentes e sua respectiva classificação, definida pela ferramenta, para que o mesmo usuário indique se a classificação está correta ou não. A Figura 5.27 ilustra o processo de treinamento de classes em andamento, em que uma linha é apresentada juntamente com a classificação atribuída para o aceite ou alteração de classificação por parte do usuário.

Figura 5.27 – Processo de treinamento de classes em andamento.
Fonte: do Autor.

Como pode-se visualizar na Figura 5.27, é apresentado para o usuário as palavras, restantes após realização da etapa de pré-processamento, que compõem determinada linha. Foram adicionadas algumas informações extras, para auxiliar na classificação da linha caso seja necessário por parte do usuário, relacionadas ao aplicativo que estava em execução e ao título da janela do sistema operacional quando as palavras foram capturadas. A classificação atribuída à linha é informada ao usuário, para que o mesmo indique se foi definida

corretamente. Caso a classificação estiver incorreta, o usuário pode indicar qual a classificação correta para a linha. Ao indicar uma nova classificação a pontuação de cada palavra que compõe a linha é alterada para que, em novas classificações, a ferramenta identifique a linha de maneira correta em relação às classes.

No momento em que a classificação de uma linha é alterada, os valores da nova classe indicada são incrementados em cada palavra que compõe a linha, ao mesmo tempo em que os valores da classe anterior, indicada pela ferramenta antes da indicação de uma nova classificação pelo usuário, são decrementados. A unidade utilizada para incrementação ou decrementação da pontuação neste tipo de treinamento é de 0,1. Caso o usuário indicar que a classificação está correta, somente as palavras da linha que ainda não possuírem classificação sofrem alteração na pontuação, ou seja, o valor respectivo à classe atribuída destas palavras é incrementado.

5.6.2 Classificação manual de palavras

Utilizando a segunda opção de treinamento de classes a classificação de palavras é disponibilizada pela a ferramenta para que a pontuação seja realizada de maneira manual. Ao utilizar esta opção uma lista contendo todas as palavras encontradas nos arquivos resultantes da etapa de pré-processamento é disponibilizada ao usuário. Desta maneira, o usuário pode classificar somente palavras que considerar importantes, poupando a etapa de treinamento de classes. Ou então, a classificação manual pode ser utilizada para aumentar a pontuação de palavras que se identifiquem muito bem com uma única classe, aumentando assim a precisão de classificação para linhas que possuírem a palavra em questão. A Figura 5.28 ilustra a lista apresentada ao usuário para classificação. Como pode-se visualizar, a lista possui os mesmos campos existentes no arquivo de treinamento.

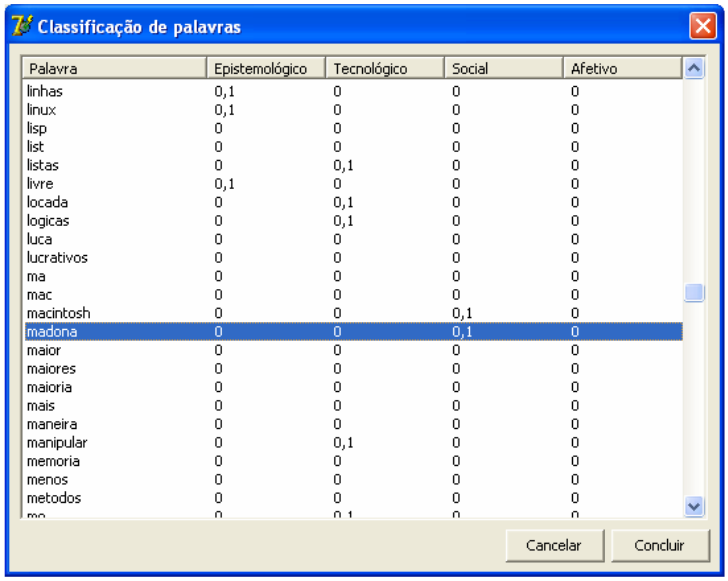


Figura 5.28 – Lista de palavras para classificação manual.
Fonte: do Autor.

Caso o usuário encontre alguma palavra que considere necessária a alteração de pontuação, basta selecionar a mesma e utilizar o duplo clique para que a tela de alteração seja exibida. A Figura 5.29 ilustra a tela para alteração de pontuação de uma palavra em relação às classes. Após finalizar todas as alterações que considerar relevantes, a classificação manual pode ser encerrada. Assim, o arquivo de treinamento é atualizado e pode ser utilizado na visualização de classificações.

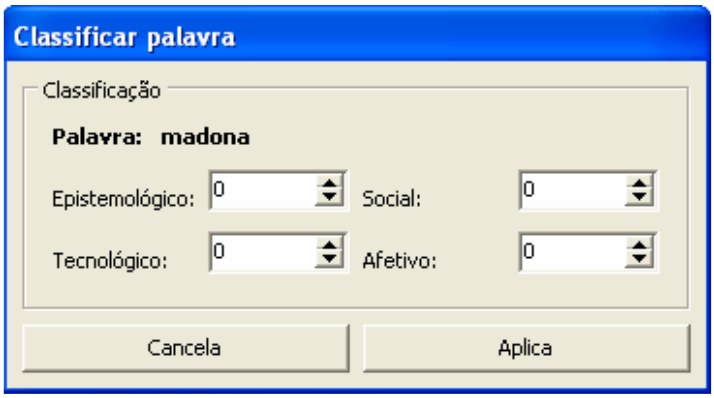


Figura 5.29 – Tela para alteração de pontuação de uma palavra.
Fonte: do Autor.

5.6.3 Treinamento de classes por importação de texto

A terceira opção para treinamento de classes possibilita ao usuário a importação de um texto, ou conjunto de palavras, indicando uma classificação para o mesmo. No momento da confirmação do texto selecionado, todas as palavras que compõem o mesmo serão adicionadas no dicionário e também adicionadas ao arquivo de treinamento, responsável pelo armazenamento das palavras com a respectiva pontuação das classes. Existe uma opção para que, ao adicionar as palavras do texto ao dicionário, seja informado ao usuário caso a palavra não seja encontrada no dicionário, se o mesmo deseja adiciona-la realmente. Este método para treinamento de classes foi desenvolvido para que fosse possível o menor esforço do usuário para a classificação das palavras, visando um aprendizado com o menor nível de supervisão. Nesta opção a pontuação atribuída a classe selecionada, em cada palavra, recebe o valor de 1, por se tratar de uma indicação direta de um texto relacionado ao conteúdo classe. A figura 5.30 ilustra a utilização da opção de importação de texto como treinamento de classes.

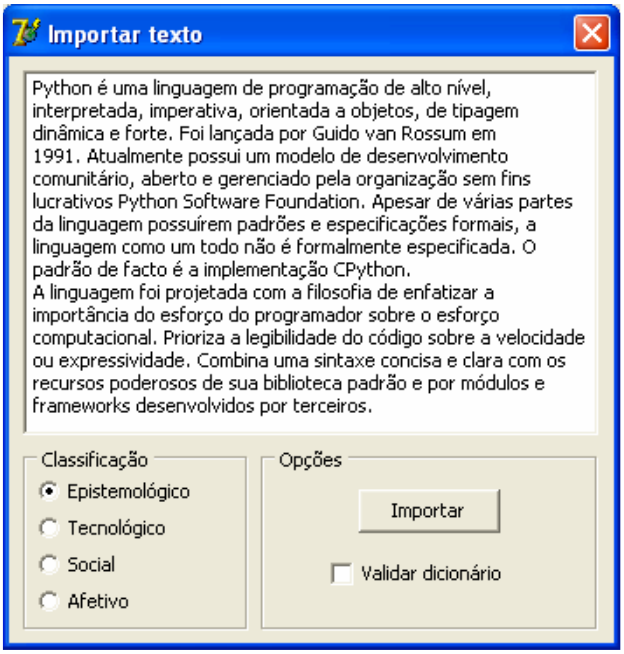


Figura 5.30 – Treinamento de classes a partir da importação de um texto.
Fonte: do Autor.

No término do treinamento de classes de todas as linhas existentes, o arquivo contendo a pontuação das palavras em relação às classes é atualizado. A partir deste momento a visualização de resultados estatísticos se torna possível, pois todas as palavras estão

devidamente classificadas. A Figura 5.31 ilustra o arquivo “Treinamento.dic” após a execução do treinamento de classes.

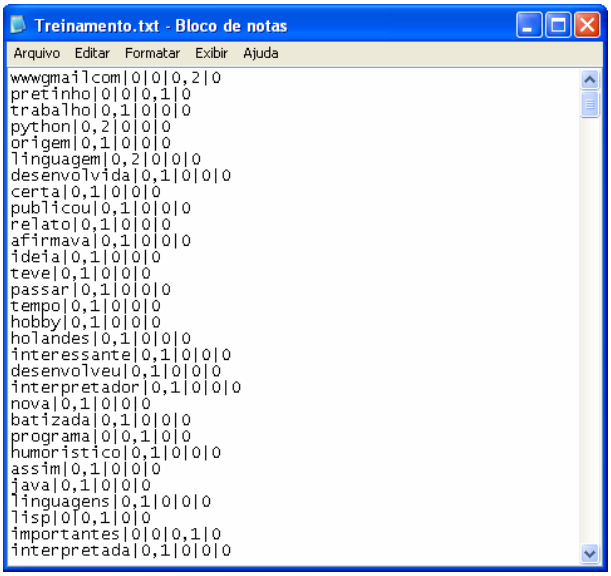


Figura 5.31 – Arquivo “Treinamento.dic” após treinamento de classes.
Fonte: do Autor.

6 RESULTADOS OBTIDOS

Com a utilização da base de dados criada na implantação das ferramentas em ambiente de ensino, a etapa de pré-processamento foi iniciada, através da utilização da ferramenta desenvolvida Data Analiser. Sendo assim, ao término de carregamento da base de dados, a primeira providência foi a verificação de novas palavras, encontradas na implantação pelo Key Monitor, que ainda não estivessem cadastradas no dicionário. Muitas palavras desconhecidas foram encontradas e estas foram então adicionadas ao dicionário. Ao total, foram adicionadas cerca de 600 novas palavras ao dicionário, todas estas oriundas da implantação. Uma quantidade razoável de possíveis palavras, também capturadas durante a implantação, não pôde ser diagnosticada em relação a sua ortografia, sendo assim, estas foram desconsideradas do processo de análise.

Após a realização da etapa de pré-processamento, a nova base de dados foi criada, e assim os dados puderam ser destinados ao processo de treinamento de classes. A etapa de treinamento de classes foi iniciada com a utilização da opção de importação de textos. Foram importados textos relacionados ao assunto da atividade proposta no ambiente de ensino, ou seja, sobre a linguagem de programação Python, e classificados como Epistemológico. Os textos em questão foram selecionados de fontes da internet, como por exemplo, da enciclopédia livre *online* Wikipédia. Após a inserção dos textos foi realizada uma pré-visualização dos resultados, a fim de observar o impacto da opção de treinamento por importação de texto. Os resultados da primeira etapa de treinamento de classes podem ser visualizados na Figura 6.1.

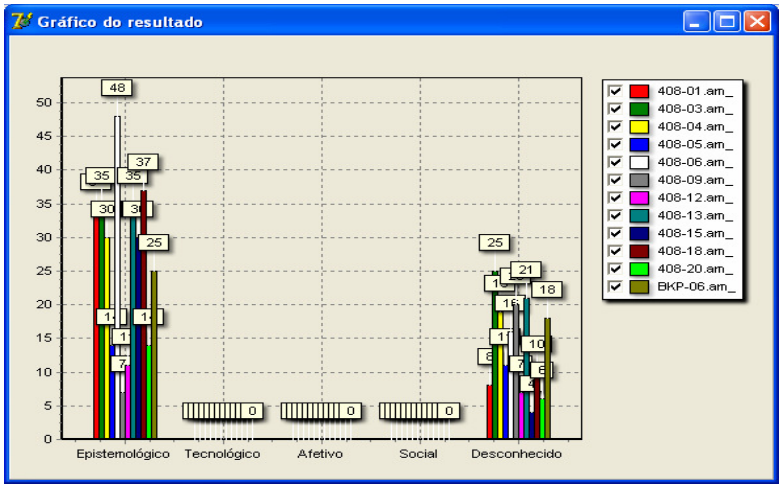


Figura 6.1 – Resultados após treinamento de classes por importação de texto.
Fonte: do Autor.

Como se pode visualizar na Figura 6.1, grande parte das linhas foram classificadas como epistemológico, isso indica que o treinamento de classes por importação é de grande importância na classificação das linhas envolvidas devido a sua facilidade de utilização. O restante das linhas foi classificado como “Desconhecido”, devido ao fato destas não terem nenhuma pontuação para alguma das classes existentes. Notam-se também os valores em zero para as classes Tecnológico, Afetivo e Social, por não terem nenhuma linha classificada como tal. O número total de linhas oriundas da implantação e que sobreviveram à etapa de pré-processamento foi de 485. Com a utilização do treinamento de classe baseado em importação de texto, cerca de 320 linhas foram classificadas, ou seja, 66% das linhas obtiveram classificação.

A opção de treinamento por importação de texto foi utilizada somente para a classe “Epistemológico”, pois as outras classes apresentam um conteúdo muito abrangente. Sendo assim, mesmo com importações de muitos textos poderia ocorrer que nenhuma linha fosse atingida na classificação.

Visando uma redução do número de linhas classificadas como “Desconhecido”, a opção de classificação manual de palavras foi utilizada, para visualização das palavras envolvidas. Deste modo, caso fosse diagnosticada a presença de palavras intuitivas a alguma classe, estas seriam classificadas de acordo com tal percepção do usuário. Sendo assim, muitas palavras foram encontradas e classificadas manualmente. A maioria das palavras classificadas manualmente estavam relacionadas a assuntos sociais. A Tabela 6.1 apresenta algumas das

palavras, identificadas pelo usuário, que foram classificadas de maneira manual, com sua respectiva classe.

Palavra	Classificação
Bolão	Social
Campeão	Social
Fardamento	Social
Fifa	Social
Goleiro	Social
Gols	Social
Grêmio	Social
Inter	Social
Ipconfig	Tecnológico
Jogadores	Social
Jornal	Social
Msn	Social
Pânico	Social
Pascal	Tecnológico
Tatuagens	Social
Tv	Social
Uol	Social

Tabela 6.1 – Palavras classificadas manualmente.
Fonte: do Autor.

Após a classificação manual de algumas palavras, pôde-se notar uma alteração em relação ao número de linhas classificadas anteriormente como “Desconhecida”, por não terem pontuação referente à nenhuma classe. O número destas linhas passou de 165 para 130, o que indica que a classificação manual não causou tanto impacto quanto era esperado. Isto se dá ao fato de existir uma grande dificuldade de classificação de palavras de maneira individual, ou seja, não existe o auxílio de outras palavras para indicação do verdadeiro significado da mesma. Sendo assim, o número de palavras classificadas manualmente é geralmente um número baixo, pois esta classificação só pode ser utilizada em palavras que indiquem forte participação em alguma das classes. A Figura 6.2 apresenta o impacto que a classificação manual causou em relação aos resultados.

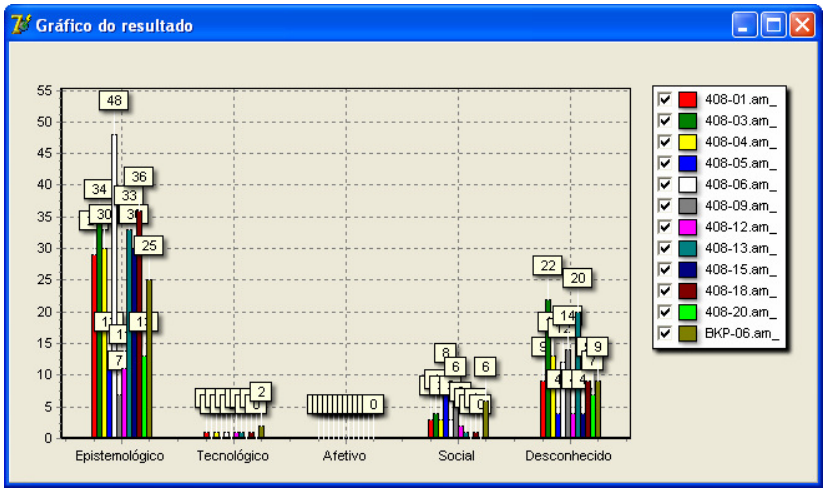


Figura 6.2 – Resultados após a etapa de classificação manual de palavras.
Fonte: do Autor.

Como um número muito alto de linhas classificadas como “Desconhecido” ainda estava presente dentre os dados, decidiu-se realizar a opção de treinamento de classes supervisionada. Sendo assim, o usuário terá de indicar a classificação correta das linhas restantes. Durante a realização do treinamento supervisionado, foram encontradas muitas linhas compostas por palavras que apresentavam um sentido totalmente adverso ao sentido exposto pelas mesmas caso fossem classificadas individualmente. A Tabela 6.2 apresenta alguns destes casos encontrados durante o treinamento supervisionado, juntamente com a respectiva classificação. A classificação para estes casos somente pôde ser atribuída visualizando-se as palavras coletivamente.

Conjunto de palavras	Classificação
Antônio, nunes	Social
Bolão, participar, prêmio	Social
Colheita, feliz	Social
Copa, bolão, resultados	Social
Favoritos, melhores, pra, copa	Social
Pânico, tv	Social
Pretinho, básico	Social

Tabela 6.2 – Exemplos de casos encontrados durante o treinamento supervisionado.
Fonte: do Autor.

Um fator de grande importância no treinamento supervisionado foi a apresentação do título da janela e o nome do aplicativo relacionado às palavras capturadas durante o mesmo, pois em muitos casos serviu como auxílio para decidir a classificação correta para as linhas. Sendo assim, o treinamento supervisionado foi realizado nas 130 linhas que ainda não haviam recebido a devida classificação e somente 30 destas puderam ser classificadas pela interpretação do usuário. Cerca de 100 linhas não puderam ser classificadas, pois não possuíam a quantidade de informações necessárias para tal fato. Muitas linhas possuíam poucas palavras, e estas não apresentavam nenhuma relação com nenhuma das classes, deste modo, este foi o principal motivos por existirem muitas linhas sem classificação. A Tabela 6.3 apresenta um resultado geral da implantação, levando-se em consideração também as linhas que não puderam ser classificadas.

Classificação	Número de linhas	Percentual
Epistemológico	306	65%
Tecnológico	11	2%
Social	50	11%
Afetivo	0	0%
Desconhecido	103	22%
Total	**470**	

Tabela 6.3 – Resultado geral da implantação.
Fonte: do Autor.

O resultado geral da classificação de linhas pode ser visualizado na ferramenta desenvolvida, com o objetivo de disponibilizar uma percepção geral dos valores obtidos ao usuário. A Figura 6.3 apresenta a utilização deste recurso presente na ferramenta para visualização do resultado final.

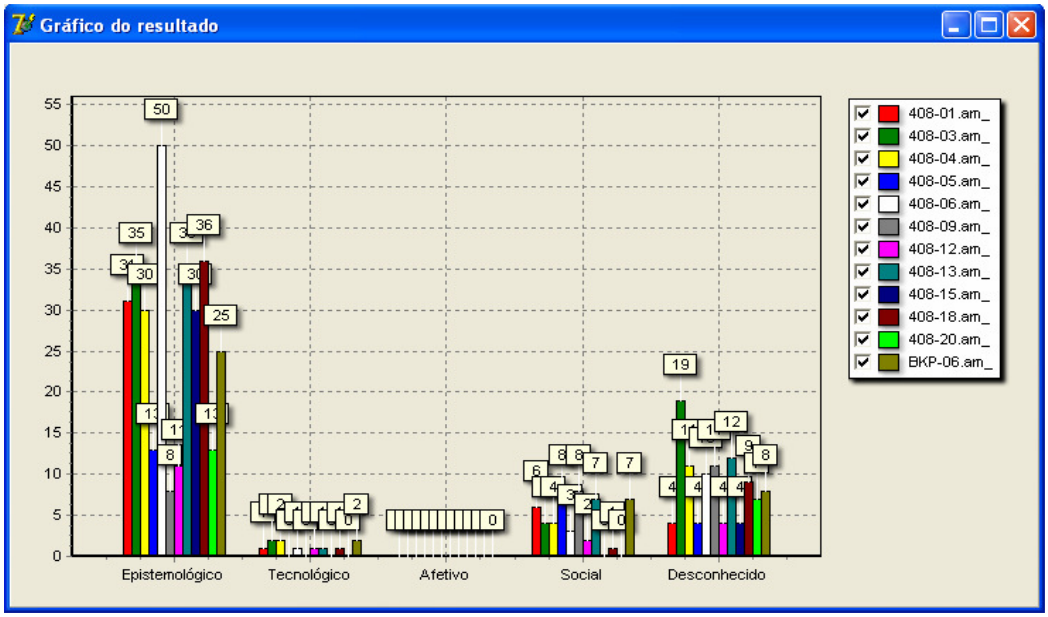


Figura 6.3 – Resultado final da classificação.
Fonte: do Autor.

Os resultados também estão disponíveis de forma individual, ou seja, baseando-se somente em um usuário. No caso do presente trabalho, a separação foi realizada em relação a cada computador do ambiente de implantação. Sendo assim, a Figura 6.4 ilustra a utilização de resultado individual baseando-se no computador denominado “408-01”.

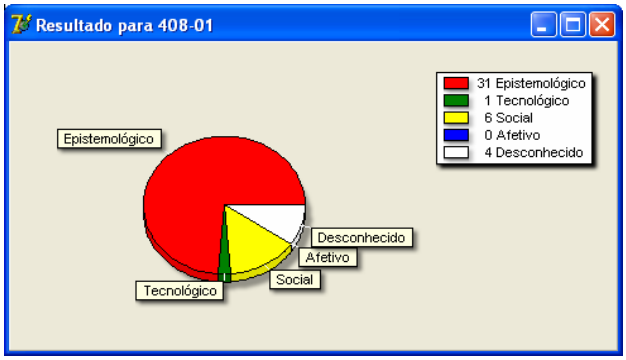


Figura 6.4 – Gráfico gerado pela ferramenta Data Analiser ilustrando resultado individual.
Fonte: do Autor.

Analisando-se os resultados finais obtidos através da implantação pode-se perceber que a maioria do conteúdo classificado estava de acordo com a atividade proposta em sala de aula. Uma pequena parte foi classificada como não pertencente ao domínio da atividade,

conforme pode ser visualizado na Tabela 6.3. E outra parte de conteúdo não pode ser avaliado em relação às classes existentes, por se tratar de dados insignificantes. Sendo assim os resultados obtidos foram encaminhados ao professor referente a disciplina em que a implantação foi realizada.

CONCLUSÃO

Pode-se afirmar que a mineração de dados é comumente utilizada em aplicações onde existe uma dificuldade muito grande de análise das informações de maneira manual. Desta forma, torna-se imprescindível o desenvolvimento de ferramentas que auxiliem o ser humano, de forma automática e inteligente, na tarefa de analisar, interpretar as informações contidas nos dados.

O principal objetivo do presente trabalho foi o desenvolvimento de uma ferramenta que auxiliasse na tarefa de classificação de dados capturados por aplicativos *keyloggers*. Sendo assim, também houve a necessidade de desenvolvimento de uma ferramenta que cumprisse a função de *keylogger*, para geração dos dados, e uma ferramenta responsável pela centralização de todos os dados gerados pela mesma. Todas as ferramentas citadas foram desenvolvidas e aplicadas com sucesso na implantação apresentada no decorrer do trabalho.

A implantação realizada em ambiente de ensino proporcionou uma análise relacionada ao desempenho e postura dos alunos em relação a atividade proposta pelo professor em aula. Sendo assim, foi possível identificar quais alunos mantiveram suas ações relacionadas à atividade proposta, e os alunos que se focaram e atividades sociais ou assuntos particulares, assim como o percentual de utilização, para cada classificação existente, por aluno.

A ferramenta desenvolvida para classificação dos dados apresentou resultados significativos em relação aos dados gerados na implantação no ambiente de ensino. Mostrou-se bastante eficaz na classificação das linhas, auxiliando o usuário para que o mesmo não necessitasse verificar manualmente a classificação correta para cada uma destas. O tempo necessário para geração de resultados na ferramenta foi considerado bom, porém algumas

etapas tais como cadastramento de palavras no dicionário e treinamento de classes supervisionado acabaram influenciando o mesmo. Para enfrentar este problema a opção de treinamento de classes por importação mostrou-se muito eficaz, pelo fato de não depender destas duas etapas. O método utilizado para pontuação de palavras em relação às classes demonstrou boa desenvoltura, pelo fato de basear-se na experiência obtida através das informações fornecidas pelo usuário.

Durante a etapa de verificação de resultados obtidos a partir da implantação do trabalho proposto foram encontradas algumas dificuldades, tais como: um número considerável de linhas não pôde ser classificada, pois estas apresentavam informações insuficientes para serem relacionadas a alguma classe, este fato veio a influenciar nos resultados finais da implantação; e a necessidade de treinamento supervisionado destas linhas não classificadas. Porém, após o termino do mesmo, foi evidenciado que apenas uma pequena parte das linhas possuía realmente uma alternativa à classificação, ou seja, a maioria das linhas não classificadas estavam desta forma por realmente não apresentar indícios a nenhuma classe existente.

Outra dificuldade encontrada está relacionada a quantidade significativa de palavras ortograficamente incorretas, capturadas pela ferramenta *keylogger*. Estas poderiam influenciar em alguns resultados, caso estivessem ortograficamente corretas. Porém, existe uma possibilidade de que estas palavras não tenham realmente significância, e que tenham sido descartadas pelo usuário responsável por sua criação, ou seja, o usuário que estava utilizando o computador no qual o *keylogger* estava instalado.

Como sugestões para trabalhos futuros, podem ser citados: um processo de melhoria no trabalho já realizado, em relação aos dados gerados pela ferramenta *keylogger*, pois muitas palavras tiveram de ser desconsideradas devido a sua ortografia incorreta, ou seja, a criação de uma etapa focada na identificação de todas as palavras que não foram encontradas no dicionário em decorrência de erros de digitação; a utilização de dados capturados pela ferramenta *keylogger* em aplicativos mais difundidos atualmente na área de mineração de dados, como por exemplo o aplicativo WEKA.

REFERÊNCIAS BIBLIOGRAFICAS

ADEODATO, Paulo J. L.; SANTOS, Roberto A. F.; VASCONCELOS, Germano C.; SANTOS, Carlos F.. **Uma Aplicação de Mineração de Dados á Previsão de No-Show no Agendamento de Serviços Médicos**. Recife, PE: 2005. Artigo. Universidade Federal de Pernambuco (UFPE) – Centro de Informática.

AMARAL, Fernanda Cristina Naliato. **Data Mining: Técnicas e aplicações para o marketing direto**. São Paulo, SP: Editora Berkeley, 2001.

ARANHA, Christian; PASSOS, Emanual. **A Tecnologia de Mineração de Textos**. Revista Eletrônica de Sistemas de Informação, Nº2, 2006. Artigo.

BARION, Eliana Cristina Nogueira; LAGO, Decio. **Revista de Ciências Exatas e Tecnologia – Mineração de Textos**. Valinhos, SP: 2008. Vol. III, Nº. 3. Anhanguera Educacional S.A.

BASSANI, Patrícia B. Scherer. **Trocas interindividuais no fórum de discussão: um estudo sobre as comunidades de aprendizagem em espaços de educação à distância**. Centro Universitário Feevale, Novo Hamburgo, RS. 2009. Disponível em: http://www.dcc.unesc.net/sulcomp/06/artigos/sessaoOral/21993.pdf . Acesso em 10 de maio de 2010.

BEPPLER, Márcio Duarte; FERNANDES, Anita Maria da Rocha. **Aplicação de Text mining para a Extração de Conhecimento Jurisprudencial**. São José, SC: 2005. Artigo. Curso de Ciência da Computação - Universidade do Vale do Itajaí.

BRAGA, Luis Paulo Vieira**. Introdução à Mineração de Dados**. 2ª Edição. Rio de Janeiro, RJ: E-Papers, 2005.

BUNESCU, Razvan; MOONEY, Raymon J.. **Knowledge from Text Using Information Extraction**. Austin, TX: 1995. University of Texas at Austin. Department of Computer Sciences.

CIAMPA, Mark. Security Awareness: **Applying Practical Security in Your World**. Editora Course Technology, 3ª Edição, 2009.

FAYYAD, Usama M; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic; UTHURUSAMY, Ramasamy; WIEDERHOLD, Gio. **Advances in Knowledge Discovery and Data Mining.** California, USA: American Association for Artificial Inteligence, 1996.

FELDMAN, Ronen; SANGER, James. **The Text Mining Handbook: advanced approaches in analyzing unstructured data**. New York, NY: 2007. Cambridge University Press.

FRANK, Eibe; WITTEN, Ian H.. **Data mining: Know it all.** Burlington, MA: Elsevier, 2009.

FRANK, Eibe; WITTEN, Ian H.. **Data mining: Pratical machine learning tools and techniques with Java implementations.** San Diego, CA: Morgan Kaufmann, 2000.

GOLDSCHMIDT, Ronaldo; PASSOS, Emmanuel. **Data Mining: Um Guia Prático:Conceitos, Técnicas, Ferramentas, Orientações e Aplicações**. Rio de Janeiro, RJ: Elsevier, 2005.

GOMES, Igor Ruiz; MONTEIRO, Leda de Oliveira; OLIVEIRA, Thiago. **Etapas do Processo de Mineração de Textos – uma abordagem aplicada a textos em Português do**

Brasil. Belém, PA: 2006. Artigo. Anais do XXVI Congresso da SBC. Workshop de Computações e Aplicações.

GONÇALVES, Eduardo Corrêa. **Extração de árvores de decisão com a ferramenta de Data Mining Weka**. Artigo disponível em <http://www.devmedia.com.br/articles/viewcomp.asp?comp=3388>. Acesso em 05 Nov. 2009.

HAN, Jiawei; Kamber, Micheline. **Data Mining: Concepts and Techniques**. San Diego, CA: Editora Morgann Kaufmann, 2001.

HESS, Guillermo Nudelman; WAGNER, Eduardo Antônio. **Mineração de dados aplicada na análise de pesquisas de clima organizacional**. Campinas, SP: 2008. IV Workshop em Algoritmos e Aplicações de Mineração de Dados.

JERÔNIMO, Paulo Marcelo. **Estudo sobre Data Mining. Data Warehouse. Cases – Data Warehouse.** Novo Hamburgo, RS: Monografia (Bacharelado em Ciência da Computação) – Institutos de ciências exatas e tecnológicas, Centro Universitário Feevale, 2001.

KIM, Jin Sung. **Customized Recommendation Mechanism Based on Web Data Mining and Case-Based Reasoning. Intelligent Agents for Data Mining and Information Retrieval.** Hershey, PA: Idea Group, 2004.

KRANZ, Paulo Henrique. **Business Intelligence: Estudo Aplicado em Cooperativa Médica**. Novo Hamburgo, RS: 2004. Monografia (Bacharelado em Ciência da Computação) – Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale.

LINDGREN, Eva; SULLIVAN, Kirk P. H. **Computer keystroke logging and writing: methods and applications**. Editora Emerald Group Publishing, 2006.

LOH, Stanley. **Text Mining por Stanley Log**. Blog disponível em < http://miningtext.blogspot.com/2008/11/listas-de-stopwords-stoplist-portugues.html>. Acesso em 10 Nov. 2009. Publicado em 26 Nov. 2008.

MIRANDA, Marihá Renaty Ferrari. **Do direito à intimidade do empregado em confronto com o controle patronal dos meios tecnológicos (e-mail e internet)**. Jus Navigandi, Teresina, ano 14, n. 2485, 21 abr. 2010. Disponível em: <http://jus2.uol.com.br/doutrina/texto.asp?id=14726>. Acesso em: 27 de maio 2010.

PINHEIRO, Emeline Piva. **Crimes virtuais: Uma Análise da criminalidade informática e da resposta estatal**. 2006. Disponível em: <http://www.buscalegis.ufsc.br/arquivos/Vi-%203.03.pdf>. Acesso em: 17 ago. 2008, p. 5.

PRADO, Hércules Antônio; OLIVEIRA, José Palazzo Moreira; FERNEDA, Edilson; WIVES, Leandro Krug; SILVA, Edilberto Magalhães; LOH, Stanley. **Text Mining in the Context of Business Intelligence**. 2005.

SANTOS, Daiana Pereira. **Mineração em notas fiscais de entrada de empresa calçadista**. Novo Hamburgo, RS: 2008. Monografia (Bacharelado em Ciência da Computação) – Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale.

SILVA, Cassiana Fagundes da; GALHO, Thaís Silva. **Mineração de Textos utilizando Técnicas de Aprendizado de Máquina e Lógica Difusa**. Faculdade Seama, Macapá, AP; Universidade Luterana do Brasil (ULBRA), Gravataí, RS. 2006. Disponível em: <http://www.dcc.unesc.net/sulcomp/06/artigos/sessaoOral/21993.pdf>. Acesso em 02 de maio 2010.

SILVA, Cláudio Aurélio. **Descoberta de conhecimento em uma base de dados de uma academia**. Novo Hamburgo, RS: 2008. Monografia (Bacharelado em Ciência da Computação) – Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale, 2008.

SILVA, Edilberto Magalhães. **Descoberta de Conhecimento com o uso de *Text Mining*: Cruzando o Abismo de Moore**. Brasília, DF: 2002. Dissertação (Pós-graduação em Gestão do Conhecimento e da Tecnologia). Universidade Católica de Brasília.

SILVA, Jaqueline Uber. **Text Mining com uma Aplicação na Validação dos Registros de Ocorrência Policiais na Região da Grande Florianópolis**. Florianópolis, SC: 2005. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Santa Catarina

SHOLOM, Weiss M.. **Text Mining: predictive methods for analyzing unstructured information**. NewYork, NY: Springer, 2005.

STAHNKE, Fernando Rafael. **Uso de Data Mining no mercado financeiro**. Novo Hamburgo, RS: 2008. Monografia (Bacharelado em Ciência da Computação) – Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale.

TONIAZZO, Lucar Hernani Giovenardi. **Módulo de recuperação de conhecimento para auxílio na tomada de decisão em um sistema de ouvidoria**. Novo Hamburgo, RS: 2005. Monografia (Bacharelado em Ciência da Computação) – Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale.

TWO CROWS CORPORATION. **Introduction to Data Mining and Knowledge Discovery**. Potomac, MD: 1999.

ULBRICH, Henrique César; VALLE, James Della. **Universidade Hacker**. São Paulo, SP: Digerati, 2003. 2ª Edição.