UNIVERSIDADE CANDIDO MENDES - UCAM

EROS ESTEVÃO DE MOURA

UMA FERRAMENTA DE MINERAÇÃO DE TEXTOS PARA ASSISTIR AS DÚVIDAS DOS ALUNOS NO PROCESSO DE ENSINO EM AMBIENTES VIRTUAIS DE APRENDIZAGEM

v.1

CAMPOS DOS GOYTACAZES, RJ
2006

UNIVERSIDADE CANDIDO MENDES - UCAM
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO
MESTRADO EM INTELIGÊNCIA COMPUTACIONAL E PESQUISA OPERACIONAL

Eros Estevão de Moura

UMA FERRAMENTA DE MINERAÇÃO DE TEXTOS PARA ASSISTIR AS DÚVIDAS DOS ALUNOS NO PROCESSO DE ENSINO EM AMBIENTES VIRTUAIS DE APRENDIZAGEM

Dissertação apresentada à Universidade Candido Mendes -UCAM- Campos, como parte dos requisitos para obtenção do grau de MESTRE em Pesquisa Operacional e Inteligência Computacional.

Orientadora: Prof[a]: Sahudy Montenegro González, D.Sc.

CAMPOS DOS GOYTACAZES, RJ
2006

Dedico esta dissertação a minha família

Carla, Luana e Lucas,

cujo apoio em todos os momentos foi fundamental
para o sucesso desse trabalho.

Agradecimentos

Dedico meus sinceros agradecimentos para a professora Sahudy Montenegro González, D.Sc., pela orientação e incentivo. Ao professor Leandro Krug Wives, D.Sc., pela orientação no caminho do Text Mining. A professora Jacqueline Uber Silva, M.Sc., pela paciência e ajuda em Text Mining e a todos os colegas do Mestrado da UCAM-Campos.

Resumo

UMA FERRAMENTA DE MINERAÇÃO DE TEXTOS PARA ASSISTIR AS DÚVIDAS DOS ALUNOS NO PROCESSO DE ENSINO EM AMBIENTES VIRTUAIS DE APRENDIZAGEM

Na Educação a Distância (EaD), um dos maiores obstáculos para o acompanhamento do aprendizado é o contato entre professor e aluno, já que, na maior parte do tempo, é mediado pelo computador, limitando-se às mensagens escritas. Assim, uma das diferenças entre o acompanhamento na educação presencial e na educação a distância está na observação realizada pelo professor, que não pode mais contar com o ”corpo a corpo” da sala de aula. Nos ambientes virtuais de aprendizagem, a observação é feita por meio das interações do aluno com o ambiente. Para lidar com essa situação, este trabalho propõe a utilização de técnicas de mineração de textos para apoiar a aprendizagem dos alunos. Os alunos podem expor suas dúvidas em linguagem natural, que serão respondidas com a melhor resposta possível, obtida a partir da mineração da base de conhecimento preenchida pelo professor especialista da disciplina. A interface pode ficar disponível em listas de discussão, bate-papo (*chats*) ou fóruns, que são mecanismos de comunicação do ambiente de aprendizagem. Para validar as idéias propostas, é apresentado um estudo de caso, desenvolvido dentro do ambiente do Teleduc, os testes feitos e seus resultados.

PALAVRAS CHAVES: educação a distância, mineração de texto, mineração de dados.

Abstract

A TOOL OF TEXT MINING TO ATTEND THE DOUBTS OF THE STUDENTS IN THE PROCESS OF TEACHING IN COMPUTATIONAL ENVIRONMENTS FOR DISTANCE LEARNING

One of the largest obstacles in Distance Learning is, literally, the geographical distance between students and teachers and, as a consequence, the lack of face-to-face contact in the learning process. One of the differences between presential and distance learning is the accompanying process by teachers to students in classrooms. In computational environments for distance learning, this supervision is based on student-environment interactions. To deal with this problem, this paper describes a tool to use text mining to support distance learning. Students can expose their doubts in natural language and the tool returns the best feasible answer obtained from the knowledge base data mining. The knowledge base is constructed by specialist teachers of each discipline. The tool interface can be available in forums, chats and discussion lists in the distance learning environment. A case study was developed using the TelEduc environment. Tests and their results measure the proposal effectiveness.

KEYWORDS Distance Learning, Text Mining, Data Mining.

Lista das Abreviaturas

1. AM - Aprendizado de Máquina
2. ARFF - Attribute-Relation File Format
3. AVA - Ambientes Virtuais de Aprendizagem
4. CHAT - Neologismo para designar aplicações de conversação em tempo real
5. DCBD - Descoberta de Conhecimento em Bases de Dados
6. DCT - Descoberta de Conhecimento em Texto
7. DOC - Documento formato Microsoft Word
8. EaD - Educação a Distância
9. EI - Extração de Informação
10. GPL - General Public License
11. HTML - HyperText Markup Language
12. IC - Inteligência Competitiva
13. IR - Information Retrieval
14. ITA - Intelligent Teaching Assistant
15. JDBC - Java DataBase Connectivity
16. KDD - Knowledge Discovery in Databases
17. MD - Mineração de Dados

18. MT - Mineração de Textos
19. PDF - Portable Document Format
20. PHP - PHP: Hypertext Preprocessor
21. RI - Recuperação da Informação
22. RTF - Rich Text Format
23. SGBD - Sistema Gerenciador de Banco de Dados
24. SQL - Structured Query Language
25. SRI - Sistema de Recuperação de Informação
26. TELEDUC - Ambiente de Ensino a Distância - UNICAMP
27. TIC - Tecnologias de Informação e Comunicação
28. UNIFOR - Universidade de Fortaleza
29. URL - Uniform Resource Locator
30. WEB - World Wide Web
31. WEKA - Waikato Environment for Knowledge Analysis

Sumário

Capítulo 1

Introdução

Um dos principais desafios nos Ambientes Virtuais de Aprendizagem (AVAs) é melhorar a interatividade do aluno dentro do ambiente, criando as condições necessárias para uma aprendizagem personalizada. A interatividade dentro dos ambientes virtuais de aprendizagem é um fator indispensável, pois permite ao seu aluno algum nível de participação, possibilitando interagir no próprio processo. A ação do aluno é o centro do processo. O sistema deve permitir um método de ensino individualizado que respeita o próprio ritmo de aprendizado do aluno.

Uma forma de ter interatividade nos ambientes é criar suporte para o processo educacional de uma maneira inteligente assistindo o professor nas suas tarefas, assim como, ajudando os alunos a aprender. O desenvolvimento de ferramentas visa tirar uma carga dos professores, assistindo-os em tarefas complexas ou tediosas como corrigir exames, exercícios e esclarecer as dúvidas. Assistir o processo de aprendizagem e tratar o professor como um usuário alvo é melhor do que tentar de substitui-lo e essa é a filosofia das ferramentas assistentes inteligentes. Os professores continuam no controle do ensino e são apoiados por essas ferramentas. Acreditamos que com a ferramenta adequada, o aluno estará mais motivado para aumentar sua interatividade com o ambiente. Sendo assim, a avaliação do comportamento, do nível de conhecimento e do grau de interesse do aluno poderá ser mais produtiva.

Um Sistema Tutor Inteligente (STI) é um software educacional voltado para educação que possuem inteligência, para adaptar-se ao estilo de cada aluno para que todos possam aprender, mesmo se os alunos possuem diferentes comportamentos dentro de um AVA. O STI são orientados à aprendizagem do aluno. Entretanto, YA-

CEF (2002) afirma que "auxiliar professores e instrutores a lecionar melhor é uma atividade tão importante quanto ensinar os alunos".

Em KINSHUB; HONG; PATEL (2001), cita-se que tem havido um interesse crescente em integrar o professor como usuário final de um STI. A partir dessa necessidade foram criados os sistemas Assistentes Inteligentes de Ensino (ITAs - *Intelligent Teaching Assistant systems*). O ITA é orientado a alunos e professores. Em LESTA; YACEF (2002) afirma-se que "auxilia os estudantes, mas também assiste ao professor em suas tarefas". A principal característica é focalizar os esforços para a construção de ferramentas que possam assistir ao professor em suas tarefas.

Em LESTA; YACEF (2002) apresenta-se um estudo comparando os resultados obtidos por duas turmas ao longo de um ano. Uma delas usou o sistema proposto pelas autoras para apoio do desenvolvimento da lógica e a outra não recebeu suporte de um sistema ITA. Os resultados apresentados na pesquisa concluíram que os alunos que receberam o suporte de um sistema ITA, obtiveram um crescimento de 22% nas notas de trabalhos realizados semanalmente, enquanto nas provas, as notas aumentaram em 27%. Os resultados positivos servem como uma motivação a mais para a realização da presente dissertação, pois espera-se que novas ferramentas assistentes inteligentes venham a contribuir com a aprendizagem, assim como o trabalho LESTA; YACEF (2002) demonstrou sua contribuição nesta área.

Com o objetivo de automatizar algumas tarefas do professor, este trabalho propõe uma ferramenta assistente inteligente de ensino, TextEaD, que utiliza técnicas de mineração de textos para responder as dúvidas dos alunos no processo de ensino em ambientes virtuais de aprendizagem e que pode ser inserida no contexto de um ambiente ITA. Assim, propõe-se aumentar a interatividade do aluno nos ambientes virtuais de ensino.

Desenvolver ferramentas assistentes de ensino tornam o AVA mais rico e interativo e permitem ao aluno tirar maior proveito do ambiente. Portanto, ferramentas efetivas tornarão o ambiente de aprendizagem mais confiável e os alunos passarão a utilizá-lo com maior freqüência.

A ferramenta tem como finalidade oferecer aos alunos suporte para que possam expor, em linguagem natural, suas dúvidas nas diferentes disciplinas do curso a distância. As perguntas serão respondidas com a melhor resposta possível, a partir de bases de textos agrupadas por temáticas, preenchidas pelos professores especialistas das disciplinas. A interface da ferramenta no ambiente de aprendizagem pode ficar disponível nas interfaces de listas de discussão, *newsgroups*, fórum e salas de bate-papo (*chats*) dentro de qualquer ambiente virtual de aprendizagem, com apenas

poucas linhas de código.

A ferramenta assistente foi desenvolvida em duas fases ou versões. A primeira fase desenvolve a ferramenta utilizando técnicas de mineração de dados e árvores de decisão. Após testes, e devido a resultados infelizes, passou-se a uma segunda fase onde construiu-se a ferramenta utilizando técnicas de mineração de textos. Os resultados obtidos mostram que essa obteve melhores resultados.

Para apresentar tal proposta, este trabalho está organizado em seis capítulos. O Capítulo 2 apresenta uma descrição sobre ambientes virtuais de aprendizagem e sistemas assistentes inteligentes de ensino. No Capítulo 3, é apresentado o referencial teórico sobre mineração de dados e mineração de textos, bases para o construção da ferramenta TextEaD. No Capítulo 4, são descritas duas abordagens da ferramenta, que foram desenvolvidas. O Capítulo 5 apresenta os testes e resultados da aplicação da ferramenta. Por último, o Capítulo 6 apresenta as considerações finais e as propostas para trabalhos futuros.

1.1 Trabalhos Publicados pelo Autor

Como parte desta dissertação, o autor escreveu três artigos aceitos em congressos.

1. Uma Ferramenta para o Apoio ao Aprendizado em um Ambiente de Educação a Distância - Aceito e publicado nos Anais do III Simpósio Mineiro de Sistemas de Informação - SMSI 2006 - SBC.

2. Uma Ferramenta de Mineração de Dados para Apoio ao Aprendizado em um Ambiente de Educação a Distância - Aceito e publicado nos Anais do V Simpósio de Informática da Região Centro do RS, 2006 - SIRC/RS 2006 - SBC.

3. Uma ferramenta de mineração de textos para assistir as dúvidas dos alunos em ambientes virtuais de aprendizagem - Aceito para publicação nos Anais da Conferência Ibero-Americana IADIS WWW/Internet - CIAWI 2007 - IADIS.

Capítulo 2

Ambientes Virtuais e Sistemas Inteligentes

Este capítulo apresenta os principais conceitos sobre ambiente virtual de aprendizagem e sistemas inteligentes como uma evolução no desenvolvimento desses ambientes. Os sistemas tutores inteligentes e, posteriormente, os sistemas assistentes inteligentes de ensino são novas tecnologias que surgiram e estão permitindo aos pesquisadores aprimorar muitas questões em aberto nos ambientes virtuais de aprendizagem.

2.1 Ambientes Virtuais de Ensino e Aprendizagem

Segundo AL. (2005), AVAs são sistemas que integram os diversos recursos da Internet passíveis de emprego educacional, como a transmissão de conteúdos multimídia e ferramentas de comunicação e interação, devidamente orientados por uma base pedagógica adequada aos objetivos e pressupostos da iniciativa educacional almejada.

Neste processo de ensino e aprendizagem, os estudantes devem ser sujeitos do processo de aprendizagem. Para isso, devem ser criadas situações de ensino e apren-

dizagem nas quais eles mesmos possam organizar seus estudos. Os próprios de estudos devem ser iniciados por meio de discussão e interação que é chamado de princípio do estudo por meio de comunicação e interação PETERS (2001). Segundo Landim LANDIM (1999), a interatividade envolve as mediações que constituem o tratamento dos conteúdos e das formas de expressão e relação comunicativa, que possibilitam a aprendizagem à distância.

As novas tecnologias, principalmente o computador e a Internet, têm proporcionado a criação de comunidades virtuais, cujos membros podem comunicar-se síncrona ou assincronamente e sem estarem necessariamente no mesmo lugar. Esses membros podem interagir das mais diversas formas e com os mais variados objetivos.

Um Ambiente Virtual de Aprendizagem (AVA) tem como objetivo apoiar comunidades, organizadas das mais variadas formas, visando o desenvolvimento de atividades individuais e/ou coletivas, tais como o esclarecimento de dúvidas, desenvolvimento de trabalhos em grupo, dentre outras. A interatividade em AVA é fundamental para que os alunos possam organizar suas idéias, compartilhar seus conhecimentos tornado-se sujeitos autônomos de sua aprendizagem LANDIM (1999).

Um grande volume de dados é gerado por meio do uso de diferentes ferramentas de interação. No entanto, é comum não achar nos diferentes AVA tratamento e estruturação adequados para esses dados. Este fato faz com que informações que possam existir não sejam aproveitadas GAVA (2003), perdendo assim uma fonte de informações que poderia estar agregando conhecimento ao AVA. Isso sugere com que o AVA possa ser um ambiente propício para a mineração de dados, uma vez que essa técnica tem como objetivo extrair conhecimento implícito a partir de um grande volume de dados.

Disponibilizar um ambiente virtual de aprendizagem que enriqueça a cooperação e a interatividade é uma das metas atuais à qual a literatura demonstra que estão dirigindo esforços.

2.1.1 Ferramentas de comunicação em Ambientes Virtuais de Aprendizagem

As tecnologias de informação e comunicação estão se tornando ferramentas que cada vez interativas e distribuídas, quando empregadas em AVA, proporcionando aos alunos e professores um conjunto de meios para que possam compartilhar de informações e recursos. As potencialidades da Internet e dos serviços suportados por esta, estão sendo utilizadas no processo de ensino-aprendizagem, não só pela influência da elevada quantidade e variedade de meios que disponibilizam, mas também, pelas múltiplas perspectivas de abordagem que proporcionam.

Os ambientes de aprendizagem baseados na Web surgem como um ambiente flexível que favorece o modo de trabalhar de cada aluno, de forma nomeada, de poder trabalhar ao seu próprio ritmo em qualquer lugar e a qualquer hora, de atuar de modo individual ou em grupo, de aprender a relacionar-se com os outros, comunicar, colaborar e partilhar com outros da sua comunidade de aprendizagem.

Nos ambientes AVA são utilizados vários tipos de ferramentas que possibilitam esta forma de utilização pessoal. Estas ferramentas possuem características que a tornam específicas para cada etapa no processo ensino-aprendizagem. Segundo Perrone PERRONE (2005), estas ferramentas podem ser classificadas por aplicabilidade da seguinte forma:

- Apresentação do Ambiente;
- Socializantes;
- Comunicação do curso;
- Acesso ao conteúdo;
- Construção coletiva;
- Avaliação dos participantes;
- Gerência do ambiente;

Nas ferramentas de comunicação do curso estão inseridas as ferramentas como *chat* ou bate-papo, fórum, listas de discussão e conferências. A ferramenta *chat* utiliza a comunicação síncrona e pode envolver muitas pessoas, possibilitando um bom nível de interatividade. Este trabalho propõe uma ferramenta para melhorar este nível de interatividade, possibilitando, no momento desta comunicação, uma consulta a uma base de conhecimento que pode facilitar o contato entre seus participantes.

2.1.2 O Ambiente TelEduc

Dentre muitos AVAs, existentes hoje, especial destaque merece o TelEduc para este trabalho, pois o estudo de caso, apresentado nos próximos capítulos utiliza o TelEduc, como ambiente de teste da ferramenta assistente.

O TelEduc * ROCHAM et al. (2001); ROCHA (1992) é um ambiente para a criação, participação e administração de cursos na Web. Ele foi concebido no ano de 1998 tendo como alvo o processo de formação de professores para informática educativa, baseado na metodologia de formação contextualizada desenvolvida por pesquisadores do NIED (Núcleo de Informática Aplicada à Educação) da Unicamp.

O TelEduc foi desenvolvido de forma participativa, ou seja, todas as suas ferramentas foram idealizadas, projetadas e depuradas segundo necessidades relatadas por seus usuários. Com isso, ele apresenta características que o diferenciam dos demais ambientes para educação a distância disponíveis no mercado, como a facilidade de uso por pessoas não especialistas em computação, a flexibilidade em sua utilização, e um conjunto enxuto de funcionalidades.

O mesmo foi concebido tendo como elemento central uma ferramenta que disponibiliza atividades. Isso possibilita a ação onde o aprendizado de conceitos em qualquer domínio do conhecimento é feito a partir da resolução de problemas, com o subsídio de diferentes materiais didáticos como textos, *software*, referências na Internet, dentre outros, que podem ser colocadas para o aluno usando ferramentas como: material de apoio, leituras, perguntas freqüentes, etc.

A intensa comunicação entre os participantes do curso e a ampla visibilidade dos

*TelEduc, Núcleo de Informática Aplicada à Educação, Unicamp - http://www.nied.unicamp.br/oea/soft/teleduc.html - Último acesso: Agosto, 2006

trabalhos desenvolvidos também são pontos importantes, por isso foi desenvolvido um amplo conjunto de ferramentas de comunicação como: o correio eletrônico, grupos de discussão, mural, *portfólio*, diário de bordo e bate-papo.

Atualmente o TelEduc é utilizado por várias instituições de ensino, dentre elas: universidades federais, particulares, centros tecnológicos de educação (CEFET), faculdades e associações.

2.2 Sistemas Tutores Inteligentes

Os sistemas tutores inteligentes são programas computacionais dedicados ao ensino, que utilizam técnicas da Inteligência Artificial VICCARI; GIRAFFA (2003). Proporcionar flexibilidade de acordo às necessidades de aprendizagem de um determinado aluno em um dado momento, estabelece o diferencial em relação aos precursores históricos, os sistemas CAI (*Computer Aided Instruction*). Nos sistemas CAI, a interação não depende das respostas e do desempenho dos alunos. O roteiro definido é acompanhado independentemente das ações do aluno. Os sistemas ICAI (*Intelligent Computer Aided Instruction*) surgiram para melhorar esta limitação, pois incluem técnicas de Inteligência Artificial. Atualmente, esses sistemas são chamados de Sistemas Tutores Inteligentes e Ambientes Inteligentes de Aprendizagem (ILE - *Intelligent Learning Environments*).

2.2.1 Arquitetura dos STIs

A modelagem de STIs é uma tarefa complexa, pois considera os três módulos fundamentais da arquitetura, proposta por Carbonell CARBONELL (1970) e revista por Self SELF (1999). Ao se modelar um STI devemos considerar as características do domínio (conteúdo), o comportamento observável e mensurável do aluno (modelo do aluno) e, o conjunto de estratégias a serem adotadas pelo módulo tutor na busca de um ensino personalizado.

Os STIs apresentam uma estrutura modular. A arquitetura tradicional de um STI, ilustrada na Figura (2.1), considera quatro componentes.

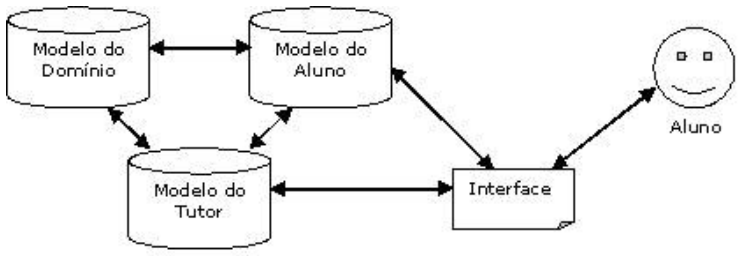

Figura 2.1: Arquitetura tradicional de um STI Fonte: Adaptado de Viccari e Giraffa (2003)

1. *Modelo do Especialista ou Domínio*: refere-se à base de conhecimento da matéria a ser ensinada. Ele inclui o conhecimento correto, na forma do conhecimento de um especialista, que é para ser transferido e aprendido para o aluno.

2. *Modelo do Aluno*: representa as ações e reações do estudante dentro do STI, mapeando todos os aspectos de comportamento que possam influenciar o processo de aprendizagem. Armazena diversos tipos de informação que serão utilizadas para decidir qual o próximo passo que o sistema deve apresentar para o estudante.

3. *Modelo Pedagógico ou Tutor*: trata do conhecimento pedagógico instrucional. O elemento pedagógico é a base para a instrução, e isso determina o que será ministrado e em que ponto. Este modelo pega informação do modelo do aluno e decide o que fazer em seguida. Ele pode, por exemplo, decidir apresentar um novo material ou revisar o material que foi previamente ensinado. Tem, na diferença de experiências entre os diversos aprendizes, a principal dificuldade para a sua implementação. Deve ser altamente adaptável.

4. *Modelo da Interface com o Estudante*: interface utilizada pelo estudante para se comunicar com o STI. Deve ser intuitivo e não oferecer complicadores para a sua utilização.

2.3 Sistemas Assistentes Inteligentes de Ensino

O STI são orientados à aprendizagem do aluno. O professor tem apenas a função de gerenciar o conteúdo instrucional. Entretanto, YACEF (2002) afirma que auxiliar professores e instrutores a lecionar melhor é uma atividade tão importante quanto

ensinar os alunos. Em KINSHUB; HONG; PATEL (2001) cita-se que tem havido um interesse crescente em integrar o professor como usuário final de um STI.

A partir desta necessidade, foram criados os assistentes inteligentes de ensino. Os ITAs são orientados a ambos, alunos e professores. Assim como os STI tradicionais, auxilia os estudantes mas também apóia ao professor em suas tarefas LESTA; YACEF (2002).

O objetivo fundamental dos ITAs é assistir aos professores, disponibilizando informações através de um ambiente, que permita identificar e auxiliar os alunos individualmente, e escolher materiais e atividades que possam auxiliar na superação das dificuldades. Ao assistir o professor, o aluno estará sendo beneficiado também pela melhoria da capacidade de atendimento. Além disso, propõe a automatização de tarefas, e facilitação de consultas referentes ao desempenho e interações realizadas pelos alunos, auxiliando na proposta de novos exercícios e materiais personalizados YACEF (2002).

2.3.1 Arquitetura de um ITA

A arquitetura de um sistema ITA, apresentada na Figura 2.2, engloba, além dos modelos presentes no STI, o módulo do professor e a interface do professor. O módulo do professor é formado por informações sobre o processo de monitoramento do aluno no STI e análise do desempenho obtido. A interface é a maneira de interação do professor com o ambiente.

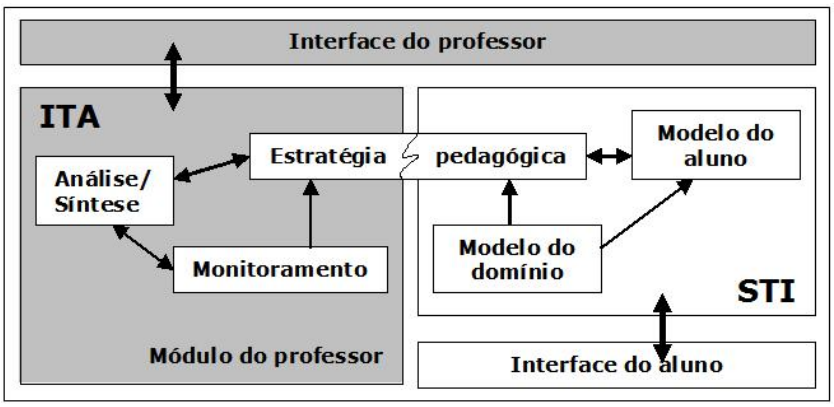

Figura 2.2: Arquitetura de um ITA Fonte: Raabe (2006), adaptado de YACEF (2002)

Em um ITA, o professor humano é assistido pelas ferramentas do módulo do pro-

fessor e, desta forma, o professor permanece presente e no controle do processo de ensino-aprendizagem. A principal característica é focalizar os esforços para a construção de ferramentas que possam assistir ao professor em suas tarefas.

Em busca por soluções, a proposta desta dissertação é o desenvolvimento da ferramenta TextEaD. TextEaD é uma ferramenta assistente de ensino, que pode ser inserida dentro de um ambiente ITA. TextEaD possibilita ao professor identificar as dúvidas dos alunos, e substitui parcialmente sua presença quando oferece assistência aos alunos, individualmente, ao responder suas dúvidas, com a melhor resposta encontrada.

2.4 Trabalhos Relacionados

Atender às necessidades específicas de cada aluno, através de um ambiente de ensino e aprendizagem computadorizado, tem sido o principal foco da área de pesquisa dos STI. Ampliando essa visão, pesquisas recentes vêm apresentando sistemas que visam assistir tanto a alunos quanto a professores (ITAs). No entanto, poucos trabalhos que descrevem ferramentas assistentes de ensino foram encontrados.

Em SILVA; RAABE (2004); RAABE; VARGAS (2006) são apresentados diferentes assistentes dentro do ITA ALICE. Com o objetivo de automatizar algumas tarefas do professor, foram criados assistentes de Detecção de Plágio, de Diagnóstico de Algoritmos, de Modalidade de Mediação, de Identificação de Dificuldades de Aprendizagem e a Personagem Alice. Diferente do restante, esta última caracteriza-se por ser um assistente de interface, apresentado visualmente aos alunos através da personagem Alice, que foi desenvolvido a fim de motivar o aluno ao aprendizado por meio do ambiente.

Em KINSHUB; HONG; PATEL (2001), descreve-se um projeto da Universidade de Massey, onde é desenvolvido um modelo de professor adaptável para ensinar a língua japonesa. Em LESTA; YACEF (2002), apresenta-se um sistema ITA que funciona como intermediário entre professores e alunos na construção de provas formais de lógica. Logic-ITA é uma ferramenta complementar que fornece para os alunos um ambiente para praticar provas formais com retroalimentação e permite ao professor monitorar o progresso e os erros dos alunos.

No trabalho de Oeiras OEIRAS (1998) é proposto o ACEL, um ambiente para o ensino / aprendizagem de línguas a distância. Este ambiente é composto de dois sub-ambientes integrados. O primeiro, denominado ambiente do professor, tem definidas as ferramentas computacionais para um professor preparar e operar cursos de línguas para a rede. Esses cursos a distância são acessados pelos alunos através do ambiente de curso que possui recursos de apoio, de comunicação e o material didático. O protótipo implementado foi testado através de um curso de leitura e produção escrita de Português para Estrangeiros cujo público-alvo foi alunos falantes de Espanhol.

A tecnologia de agentes, desenvolvida para aplicações de Inteligência Artificial, tem se mostrado uma das soluções mais adequadas para implementação do sistema de acompanhamento do aluno e de apoio ao professor em ambientes virtuais de ensino. Estão descritas a seguir algumas propostas para o uso dessa tecnologia.

Em JAQUES; OLIVEIRA (1998), Jaques e Oliveira propõem uma arquitetura Multi-Agente para monitorar os principais mecanismos de comunicação em um ambiente telemático de ensino, entre os quais estão: lista de discussão, *newsgroups* e *chat*. A tarefa dessa agência ou sociedade, isto é, uma coleção de agentes trabalhando em conjunto, seria de coletar dados a partir das discussões que se encontram em andamento, analisar quantitativamente esses dados e transmitir tais informações ao professor. Essa sociedade possui quatro agentes: um agente para coletar informações em cada mecanismo de comunicação (*newsgroups*, *chat* e lista) e um agente do professor para reunir as análises dos demais agentes. As análises dos agentes coletores estão baseadas na identificação de três tipos de associações: aluno-aluno, aluno-assunto, aluno-aluno-assunto.

Já em MENEZES; FUKS; GARCIA (1999), Menezes, Fuks e Garcia propõem uma agência com três agentes assistentes de tarefa para suportar a avaliação informal (acompanhamento) no ambiente AulaNet da Puc-Rio. As interações dos alunos com o ambiente são monitoradas por um agente que cria um histórico da navegação do aluno e percorre as listas de discussão, a fim de verificar a participação de alunos. Outro agente auxilia o professor na consulta ao relação histórica de interações, ao modelo do aluno e a uma base de conhecimentos responsável pela interpretação desse log de interação. Esse agente também é capaz de confrontar as informações decorrentes

dos processos de avaliação informal com as informações resultantes dos processos de avaliação formal do AulaNet. O terceiro agente é capaz de indicar possíveis distorções no design instrucional, refletidas em decorrência do comportamento verificado nos aprendizes.

A ferramenta descrita por Guedes, Viccari e Damico em GUEDES; VICCARI; DAMICO (2002), está relacionada ao Ambiente Multiagente de Ensino-Aprendizagem(AMEA), no qual os agentes que o compõem preocupam-se em ensinar e/ou aprender. Esta ferramenta tem por objetivo possibilitar que diversos aprendizes e professores se comuniquem, através da Internet e discutam assuntos determinados por um professor. Procurando auxiliar a tarefa do professor em determinar se os aprendizes estão realmente adquirindo conhecimento, desenvolveu-se uma ferramenta para analisar as interações dos aprendizes. O algoritmo desenvolvido utiliza um dicionário de palavras/frases-chaves relacionadas ao assunto em questão, referentes a tópicos que deveriam ser discutidos e/ou fazer parte das conclusões dos alunos. Ao ser ativado, o software identifica os aprendizes e suas respectivas interações e as armazena em uma base de dados; em seguida, avalia as interações de cada aprendiz, verificando a freqüência com que este utiliza as palavras-chave. O software permite também a classificação de todas as palavras/frases empregadas durante a reunião.

Já em JAQUES; VICCARI (2005), Jaques e Viccari propõem um agente inteligente para fornecer suporte emocional ao aluno, motivando-o, fazendo-o acreditar em suas próprias habilidades e promovendo um estado de espírito mais positivo no aluno que, de acordo com psicólogos e pedagogos, é melhor para o seu aprendizado. Para escolher as táticas afetivas adequadas, o agente estuda as emoções do aluno a partir do comportamento observável do mesmo, isto é, das ações do aluno na interface do sistema educacional.

Capítulo 3

Fundamentação Teórica sobre Mineração

Este capítulo apresenta uma descrição das tecnologias utilizadas no desenvolvimento do trabalho. A ferramenta assistente TextEaD foi desenvolvida em duas versões. Na primeira versão, foi aplicada mineração de dados utilizando classificação com árvore de decisão. A segunda versão direcionou a ferramenta à aplicação de técnicas de mineração de textos.

3.1 Processo de Descoberta de Conhecimento em Bancos de Dados

As ferramentas de exploração de dados a serem desenvolvidas devem buscar escalabilidade e poder investigativo, este último só podendo ser alcançado através de engenhosas interfaces de interação com o homem, pois se sabe que o processo de descoberta não pode ser totalmente automatizado KEIM; ANKERST; AL. (1995) já que engloba inteligência e criatividade, características que o computador ainda não é capaz de simular.

O homem ainda irá atuar decisivamente na utilização destes sistemas, que devem

auxiliá-lo adequadamente SCHNEIDERMAN (1996). Nesta perspectiva se encaixa a especialidade da ciência de computação denominada Descoberta de Conhecimento em Bancos de Dados ou *Knowledge Discovery in Databases* (KDD). O processo do KDD pode ser observado na Figura 3.1. Um processo complexo que objetiva extrair conhecimento a partir de grandes volumes de dados. O KDD é um processo de investigação constituído por várias etapas: seleção, pré-processamento, transformação, Mineração de Dados (MD) e interpretação/avaliação dos resultados FAYYAD; PIATETSKY-SHAPIRO; AL. (1996). Sua demanda vem impulsionando, principalmente, as pesquisas por novas técnicas de mineração de dados, que é o núcleo de todo processo. Essa tecnologia não está disponível para a maioria das empresas, seja por custo, falta de conhecimento sobre a MD ou por falta de técnicos preparador para este tecnologia.

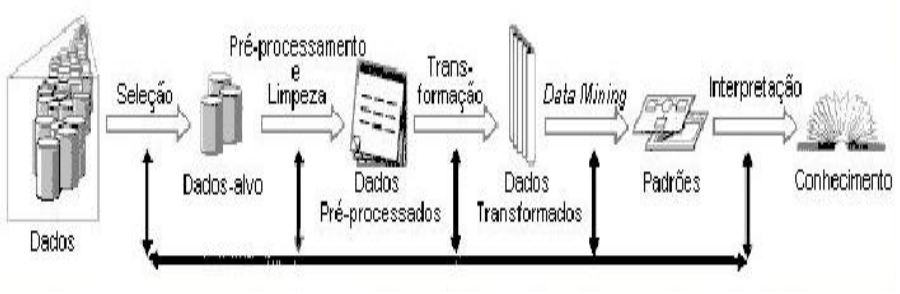

Figura 3.1: Os passos do processo de KDD Fonte: Fayyad et al. (1996)

3.2 Mineração de Dados

A mineração de dados é o processo de extrair informação válida, previamente desconhecida e de máxima abrangência a partir de grandes bases de dados BERRY; LINOFF (1997). Segundo CARVALHO (2005) *data mining* é definido como uso de técnicas automáticas de exploração de grandes volumes de dados de forma a descobrir novos padrões e relações que, devido ao volume de dados, não seriam facilmente descobertas a olho nu pelo ser humano.

3.2.1 Tarefas da Mineração de Dados

A mineração de dados é aplicada para as tarefas descritas a seguir.

Classificação e Regressão. Classificação e regressão usam dados existentes para criar modelos de comportamento de variáveis. A operação de classificação cria automaticamente um modelo a partir de um conjunto inicial de registros. Esse conjunto serve de exemplo e é chamado de conjunto de treinamento. Os registros do conjunto de treinamento devem pertencer a um pequeno grupo de classes predefinidas. O modelo é composto de padrões, essencialmente generalizações em relação aos registros, os quais são usados para diferenciar as classes. Uma vez obtido o modelo, este é usado para classificar automaticamente os demais registros. O modo como as classes são criadas oferece vantagens em relação a métodos estatísticos. Os padrões podem ser produzidos a partir de um conjunto localizado de fenômenos, ao passo que métodos estatísticos devem agir sobre populações inteiras e de distribuição bem conhecida. Desta forma é possível prever características de um pequeno percentual do conjunto de registros, o que não seria alcançado estatisticamente dado à inexpressividade dos registros sendo avaliados. Como exemplo, uma empresa de cartão de crédito poderia examinar algumas características de seus clientes e prever o nível de inadimplência associado. Tais características poderiam incluir renda, histórico de crédito, tipo e localização do emprego.

Associação. Associações são relacionamentos significativos entre itens de dados armazenados. O objetivo da operação é encontrar tendências, a partir de grande número de transações, que possam ser usadas para entender e explorar padrões de comportamento dos dados. Um exemplo seria o de varrer registros de terminais de ponto de venda e descobrir que tipos de itens são vendidos juntos, de forma a redefinir a disposição dos artigos na loja e sua promoção em campanhas publicitárias, permitindo explorar com maior eficácia essas associações.

Segmentação ou *Clustering*. O agrupamento em *clusters* envolve segmentar a informação disponível em conjuntos definidos e homogêneos baseando-se em atributos específicos. O conceito de *clustering* já tem uma longa história em esta-

tística, mas o que tem de novo em MD é o fato de poder também ser aplicada a itens não numéricos. Os resultados de uma operação de clusterização podem ser usados de duas diferentes maneiras: para produzir um sumário da base de dados ou como dados de entrada para outras técnicas, por exemplo, classificação, já que um *cluster* é um grupo menor e de mais fácil manuseio por parte de algoritmos de classificação.

Sumarização e Generalização. Como o próprio nome diz, esta tarefa procura gerar uma caracterização de um conjunto de dados fornecidos. Por exemplo, a partir de um banco de dados de um supermercado, poder-se-ia caracterizar que os clientes que compram cerveja e carne de churrasco são casados, com mais de 30 anos e pertencem a uma determinada faixa salarial.

3.2.2 Técnicas de Mineração de Dados

Especialmente devido ao alto custo envolvido, as ferramentas de mineração de dados vinham sendo usadas, quase que unicamente, por grandes corporações e instituições governamentais. A maior parte das atividades de MD ficava restrita a especialistas, com empresas oferecendo seus serviços de análise, mas sem entregar aos clientes seus métodos e ferramentas. Com o grande aumento do volume de dados nas empresas e com o crescimento do uso de tecnologia de banco de dados, especialmente de *data warehouse*, as técnicas de MD assumiram papel importante no suporte aos processos de tomada de decisão e devem, aos poucos, ganhar mercado dentre empresas de menor porte. No entanto, essas ferramentas ainda requerem de um bom nível de conhecimentos do domínio da aplicação. Em HARRISON (1998), afirma-se que não há uma técnica que resolva todos os problemas de mineração de dados. Diferentes métodos servem para diferentes propósitos, cada método oferece suas vantagens e suas desvantagens sendo mais específico para um problema.

A familiaridade com as técnicas é necessária para facilitar a escolha de uma delas de acordo com os problemas apresentados. A Tabela 3.1 apresenta um resumo das técnicas de mineração de dados normalmente usadas.

Técnicas	Descrição	Tarefas
Descoberta de regras de associação	Estabelece uma correlação estatística entre atributos de dados e conjuntos de dados	Associação
Árvores de decisão	Hierarquização dos dados, baseada em estágios de decisão (nós) e na separação de classes e subconjuntos	Classificação e regressão
Raciocínio baseado em casos	Baseado no método do vizinho mais próximo, combina e compara atributos para estabelecer hierarquia de semelhança	Classificação, segmentação
Algoritmos genéticos	Métodos gerais de busca e otimização, inspirados na teoria da evolução, onde a cada nova geração, soluções melhores têm mais chance de ter "descendentes"	Classificação, segmentação
Redes neurais artificiais	Modelos inspirados na fisiologia do cérebro, onde o conhecimento é fruto do mapa das conexões neuronais e dos pesos dessas conexões	Classificação, segmentação

Tabela 3.1: Técnicas de mineração de dados

Ferramenta (Empresa Fornecedora)	Técnicas de mineração de dados
Alice (Isoft AS (1998))	Árvore de decisão, raciocínio baseado em casos
Clementine (Integral Solutions Limited (ISL, 1996))	Indução de regras, árvores de decisão, redes neurais
Decision Series (Neovista Solutions Inc. (1998))	Árvore de decisão, métodos estatísticos, indução de regras, redes neurais
Intelligent Miner (IBM (1997))	Árvores de decisão, redes neurais
KnowledgeSEEKER (Angoss IL (Groth, 1998))	Árvores de decisão
NeuralWorks Predict (NeuralWare (Groth, 1998))	Rede neural
MineSet (Silicon Graphics Computer Systems (2000))	Métodos estatísticos, árvores de decisão, indução de regras
PolyAnalyst (Megaputer Intelligence Ltd. (1998))	Algoritmo genético
WEKA (The University of Waikato (1993, *freeware*))	Regras de associação, árvore de decisão

Tabela 3.2: Ferramentas de Mineração de Dados

3.2.3 Ferramentas de Mineração de Dados

Existem ferramentas que implementam uma ou mais técnicas de MD. A Tabela 3.2 relaciona algumas dessas ferramentas, fornecendo informações tais como: a empresa fornecedora, as técnicas implementadas de mineração de dados e exemplos de aplicações.

3.2.4 A Ferramenta WEKA

A ferramenta WEKA (*Waikato Environment for Knowledge Analysis*) WITTEN; FRANK (2005) é formado por um conjunto de implementações de algoritmos de diversas técnicas de mineração de dados, na linguagem Java. É um software de domínio público, disponível em http://www.cs.waikato.ac.nz/ml/weka/. Para a implementação da versão da ferramenta TextEaD com mineração de dados, foram utilizadas algumas técnicas

de classificação da ferramenta WEKA. Os algoritmos disponíveis para classificação, em particular para criação de árvores de decisão, podem ser visualizados na Figura 3.2.

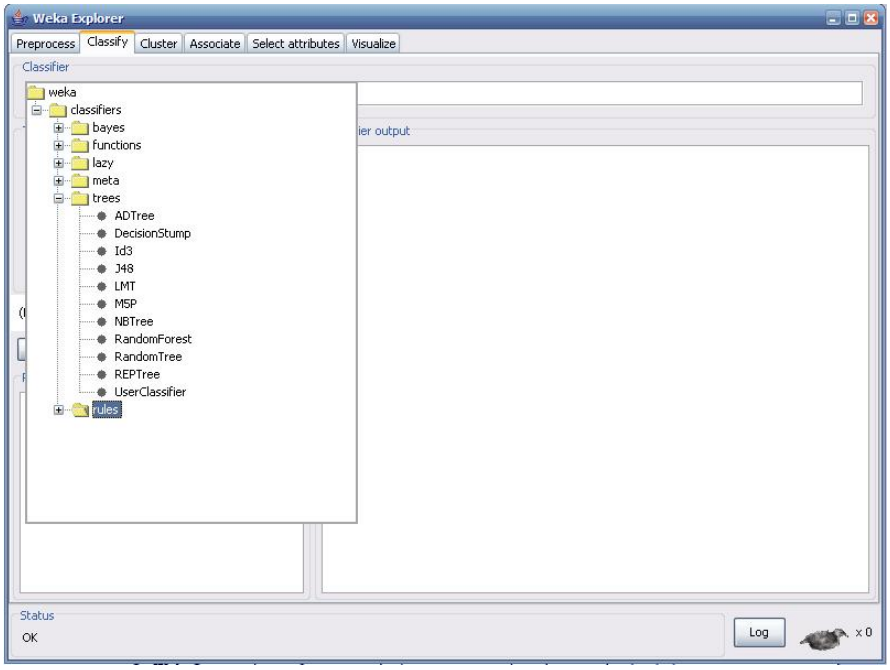

Figura 3.2: Algoritmos desenvolvidos pelo WEKA

O WEKA possui um formato próprio para o arquivo de entrada de dados, o ARFF (*Attribute-Relation File Format*). Antes de aplicar os dados a qualquer algoritmo do pacote WEKA estes devem ser convertidos para o formato ARFF.

O formato ARFF consiste basicamente de três partes. A primeira se compõe pelo nome da relação de dados após a palavra chave `relation`. A segunda começa com a palavra chave `attribute`, seguida pela lista de todos os atributos selecionados para a mineração, definidos por um tipo (`real`, `integer`, `string`) ou por um conjunto de valores. Ao utilizar os valores, estes devem estar entre "{ }"e separados por vírgula. A terceira parte consiste das instâncias ou registros de dados a serem minerados, iniciados pela palavra chave `data`. Os valores dos atributos para cada instância são separados por vírgula. A ausência de um item em um registro deve ser atribuída pelo símbolo de interrogação (`?`). A Figura 3.3 mostra um exemplo de arquivo no formato ARFF que acompanha a instalação do pacote WEKA. O arquivo contém dados sobre parâmetros atmosféricos, tais como: temperatura, umidade, vento e estado do céu. Esses dados foram coletados durante 14 partidas de golfe (registros), e servirão para

gerar as regras que definirão as condições de quando se deve ou não jogar golfe. O atributo alvo da mineração é `joga`, com as classes de valores: `sim` ou `não`.

```
@relation clima

@attribute ceu {sol, nublado, chuva}
@attribute temperatura real
@attribute umidade real
@attribute vento {verdadeiro, falso}
@attribute joga {sim, nao}

@data
sol,85,85,falso,nao
sol,80,90,verdadeiro,nao
nublado,83,86,falso,sim
chuva,70,96,falso,sim
chuva,68,80,falso,sim
chuva,65,70,verdadeiro,nao
nublado,64,65,verdadeiro,sim
sol,72,95,falso,nao
sol,69,70,falso,sim
chuva,75,80,falso,sim
sol,75,70,verdadeiro,sim
nublado,72,90,verdadeiro,sim
nublado,81,75,falso,sim
chuva,71,91,verdadeiro,nao
```

Figura 3.3: Arquivo no formato ARFF do WEKA

O arquivo é carregado no WEKA, conforme ilustra a Figura 3.4. O WEKA oferece opções para as seguintes tarefas de mineração:

- conjunto de algoritmos que implementam os esquemas de aprendizagem que funcionam como classificadores;
- conjunto de algoritmos para geração de grupos ou *clusters*;
- conjunto de algoritmos para gerar regras de associação.

O algoritmo selecionado para criar a árvore de decisão foi o `J48` para classificação. O algoritmo `J48` constrói um modelo de árvore de decisão baseado num conjunto de dados de treinamento, e usa esse modelo para classificar outras instâncias num conjunto de testes. O algoritmo "aprende"árvores de decisão, construindo-as de cima para baixo (*top-down*), começando com a questão: *Qual atributo deve ser testado na raiz da árvore?*. Para responder esta questão, cada atributo é avaliado usando um teste estatístico para determinar quão bem ele classifica o conjunto de dados de treinamento. O ganho de informação, baseado em entropia, tem prevalecido para o cálculo do atributo raiz. A entropia caracteriza a (im)pureza de uma coleção arbitrária

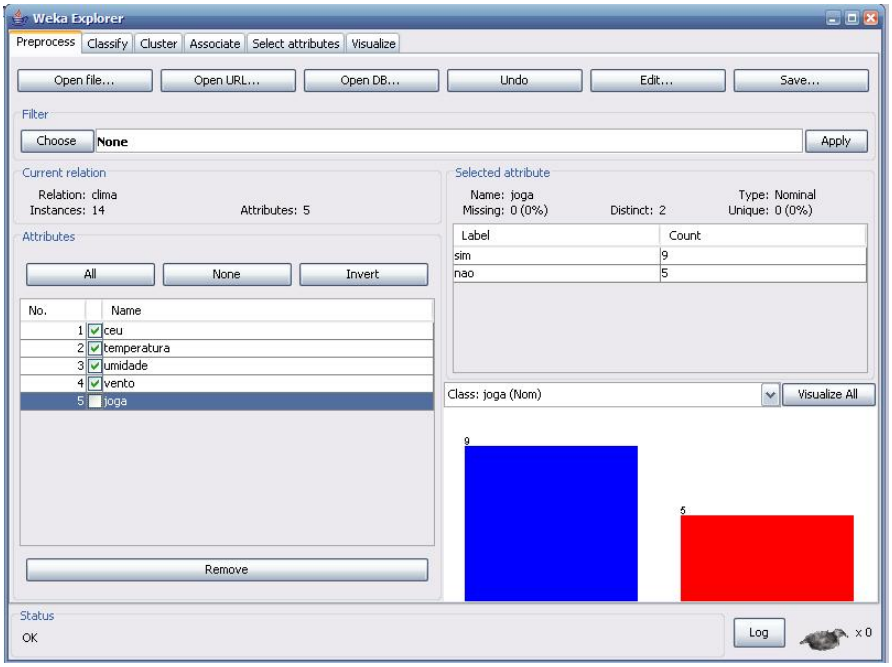

Figura 3.4: Tela do WEKA com o arquivo ARFF carregado

de treinamento. O ganho de informação é a redução esperada na entropia causada pela partição do conjunto de treinamento, de acordo com um determinado atributo. Testados todos os atributos, o atributo com maior ganho de informação é escolhido como raiz da árvore. O resultado gerado pela ferramenta pode ser observado na Figura 3.5.

A Figura 3.5 ilustra a árvore de decisão criada a partir da aplicação do algoritmo `J48` do WEKA.

Com base na árvore de decisão apresentada na Figura 3.6, podem-se extrair as seguintes regras:

```
Se ceu = sol E umidade > 75 então jogar = não
Se ceu = sol E umidade <= 75 então jogar = sim
Se ceu = nublado então jogar = sim
Se ceu = chuva E vento = verdadeiro então jogar = sim
Se ceu = chuva E vento = falso então jogar = não
```

Portanto, existem duas situações em que será permitido jogar: (1) toda vez que estiver fazendo sol e a umidade relativa do ar for menor ou igual que 75%; (2) se estiver chovendo e tiver vento.

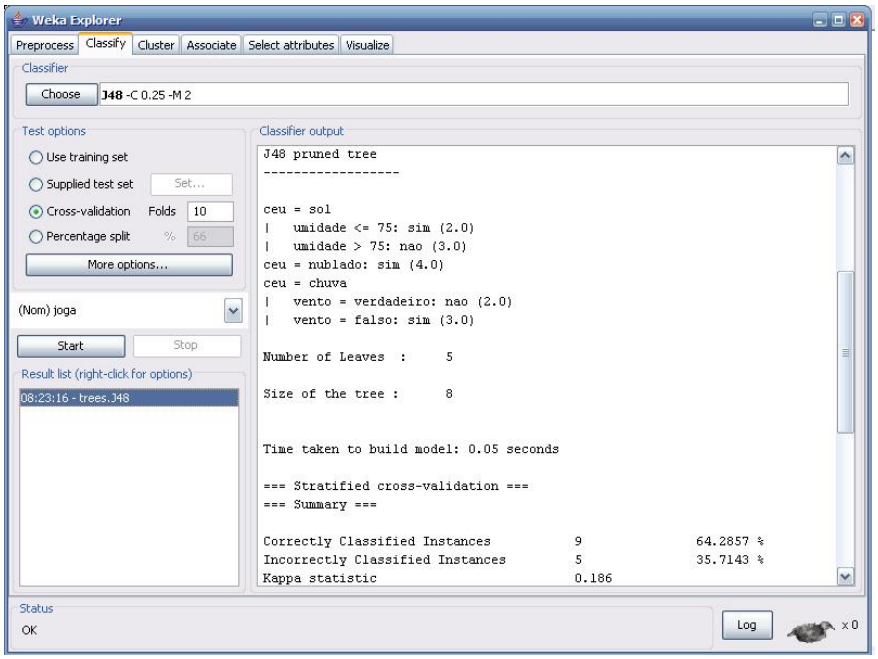

Figura 3.5: Resultado gerado pelo WEKA

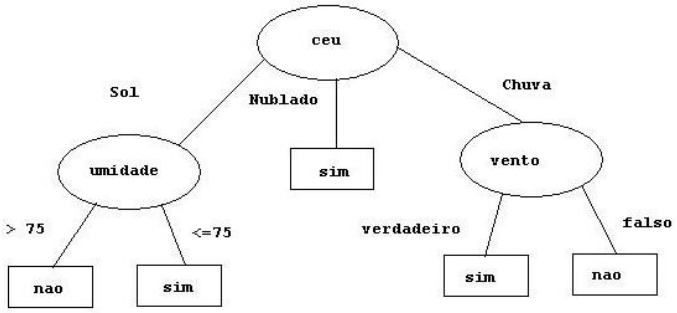

Figura 3.6: Árvore de decisão gerada pelo WEKA

3.3 Tópicos sobre Mineração de Texto

Grande parte das informações das empresas encontram-se de forma não estruturada, tais como: documentos digitalizados, e-mails, memorandos, registros de reclamações de clientes, registros de ocorrências, dentre outros. Em busca da descoberta de novos conhecimentos em textos não estruturados, aplica-se a Mineração de Textos (MT) ou, em inglês, *Text Mining*. Segundo FAYYAD; PIATETSKY-SHAPIRO; AL. (1996), apud SILVA (2002), a descoberta de conhecimento ocorre por meio de complexas interações realizadas entre o homem e uma base de dados, geralmente utilizando uma série heterogênea de ferramentas.

Em LOH; OLIVEIRA; ALMEIDA (2003), afirma-se que as três grandes áreas que lidam com informações em grandes bases de dados são: mineração de dados para dados estruturados; extração de informação para dados não estruturados e recuperação da informação para textos. A tecnologia de mineração de texto serve para identificar os conceitos presentes nos textos. Conceitos representam ”entes” do mundo real (entidades, eventos, objetos, sentimentos) e podem permitir entender que temas estão sendo tratados nos textos. Em seguida, a exploração pode utilizar um processo automático de mineração. Esta mineração pode ser feita analisando-se a distribuição dos conceitos em coleções (a freqüência ou probabilidade com que aparecem) e a relação dos conceitos entre si, para descobrir associações e dependências.

A MT é uma técnica para a análise de textos, que permite: recuperar informações, extrair dados, resumir documentos, descobrir padrões de associações e regras para classificação. Além disto, é possível realizar análises qualitativas ou quantitativas e, desta forma, soluciona grande parte dos problemas relacionados à busca, recuperação e análise de textos. Podemos observar o processo de MT na Figura 3.7.

Os sistemas de informação com suporte à MT podem beneficiar os usuários, auxiliando-os a coletar e analisar os dados necessários à tomada de decisão e permitindo com que se posicionem melhor em suas atribuições, aumentando a eficiência e reduzindo erros.

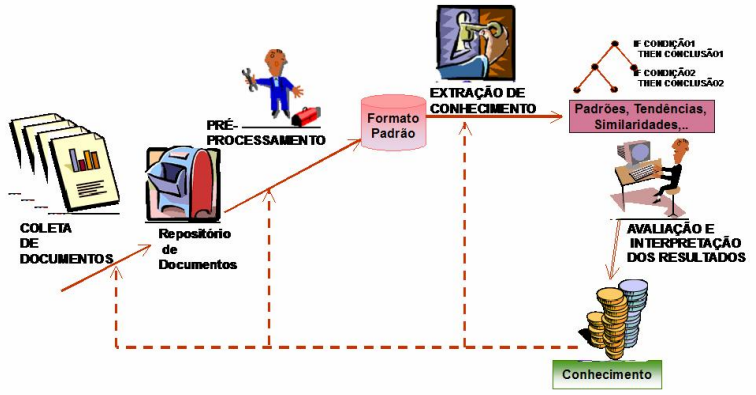

Figura 3.7: Os passos do processo Mineração de Texto

3.3.1 Conceitos básicos sobre Mineração de Texto

Para um melhor entendimento sobre como se realiza a mineração de textos, é necessário o conhecimento de alguns conceitos básicos.

Stopwords

Segundo SANTOS (2002) *stopwords* ”são palavras que ocorrem freqüentemente em textos. Uma vez que elas são muito comuns, sua presença não contribui significativamente para a determinação do conteúdo do documento”. Podemos então concluir que elas podem ser descartadas do documento, para fins de mineração. As *stopwords* podem ser: os artigos, preposições, pronomes e demais palavras utilizadas para auxiliar na construção sintática das orações. Para Wives WIVES (1999a), a tradução pode ser como ”palavras negativas”, ”palavra ferramenta” ou ”palavras vazias”. Sua remoção contribui ao aumentar a rapidez da operação, pois uma busca que emprega a preposição *de*, certamente, recupera quase todos os registros em uma base de dados. A Figura 3.8 sugere uma lista padrão das palavras em português que normalmente se encaixam no conceito de *stopwords*. Caso exista uma palavra que se repita muito na base de dados e que não tenha importância no processo de busca, esta deve ser incluída na lista. Então, é possível dizer que a lista das *stopwords* pode variar, dependendo da aplicação.

a	desligado	faz	ou
acerca	deve	fazer	outro
agora	devem	fazia	para
algumas	deverá	fez	parte
alguns	direita	fim	pegar
ali	diz	foi	pelo
ambos	dizer	foram	pessoas
antes	dois	horas	pode
ao	dos	iniciar	poderá
apontar	e	início	podia
aquela	é	ir	por
aquelas	ela	irá	porque
aquele	ele	isto	povo
aqueles	eles	ligado	primeiro
aqui	em	maioria	qual
atrás	enquanto	maiorias	qualquer
bem	então	mais	quando
bom	está	mas	quê
cada	estado	mesmo	quem
caminho	estão	meu	quieto
cima	estar	muito	saber
com	estará	muitos	são
como	este	não	sem
comprido	estes	nome	ser
conhecido	esteve	nós	seu
corrente	estive	nosso	somente
das	estivemos	novo	tal
debaixo	estiveram	o	também
dentro	eu	onde	tanto
desde	fará	os	tem

Figura 3.8: Lista de *Stopwords*

Corpus

Corpus, segundo Sardinha SARDINHA (2006), ”é um conjunto de dados lingüísticos, organizados seguindo alguns critérios, de maneira que sejam representativos do conjunto de dados, armazenados de tal modo que possam ser processados por computador, com a finalidade de gerar resultados úteis para a descrição e análise”. Existem dois tipos de *corpus*:

- um *corpus* de estudo, representado em uma lista de freqüência de palavras. O corpus de estudo é aquele que se pretende descrever;
- um *corpus* de referência, também formatado como uma lista de freqüência de palavras. Também é conhecido como ”*corpus* de controle”, e funciona como termo de comparação para a análise. A sua função é a de fornecer uma norma com a qual se fará a comparação das freqüências do *corpus* de estudo. A comparação é feita através de uma prova estatística selecionada pelo usuário. As palavras cujas freqüências no *corpus* de estudo forem significativamente maiores segundo o resultado da prova estatística são consideradas chaves, e passam a compor uma listagem específica de palavras-chaves.

O *corpus* de referência não deve conter documentos relacionados ao tema a ser analisado, pois ao comparar-se o corpus de referência com o corpus de estudo, a diferença entre eles será uma lista de palavras-chaves ou *keywords*.

Keywords

Keywords ou palavras-chave são palavras cuja freqüência é estatisticamente diferente no *corpus* de estudo em relação ao corpus de referência. Segundo SARDINHA (2006), para se analisar quais são as *keywords* necessitam-se dois elementos básicos: o *corpus* de estudo e *corpus* de referência. Estas palavras passam a compor uma listagem específica de palavras-chave mais significativas no texto.

Em SARDINHA (2006) é sugerido o seguinte algoritmo para extração de palavras-chave:

1. Selecionar o primeiro item na lista de palavras do corpus de estudo.
2. Procurar por este item na lista de palavras do corpus de referência.
3. Se o item constar no *corpus* de referência, ir para o passo a seguir, senão ir para o passo sete.
4. Comparar as freqüências através de uma prova estatística escolhida pelo usuário.
5. Se o resultado da comparação for estatisticamente significativo, copiar esta palavra para uma nova lista, e chamá-la de lista de palavras-chave.
6. Repetir este procedimento até o último item da lista de palavras do *corpus* de estudo.
7. Se um item constante da lista de palavras do *corpus* de estudo não aparecer na lista de palavras do *corpus* de referência, assumir freqüência zero para este item no *corpus* de referência.
8. Executar os passos 4, 5 e 6.

Collocations

As colocações são expressões compostas ou agrupamentos de palavras, onde o significado é a soma dos significados das partes mais algum componente semântico adicional não previsto pelas partes SANTOS (2002). Manning & Schütze MANNING; SHÜTZE (1999) definem *collocation* como ”uma expressão que consiste em duas ou mais palavras que correspondem a algum modo convencional de dizer alguma coisa”.

Choueka MANNING; SHÜTZE (1999) afirma:

> ”uma colocação é definida como uma seqüência de duas ou mais palavras consecutivas que têm características de uma unidade sintática e semântica, e cujo significado ou conotação exato e não-ambíguo não possa ser derivado diretamente a partir do significado ou conotação de seus componentes”.

Para MANNING; SHÜTZE (1999), ”colocações de uma dada palavra são afirmações dos lugares comuns ou habituais daquela palavra”. Um exemplo disso está na expressão ”chutar o balde”. Apesar de não está falando sobre baldes, mas é uma forma de expressão utilizada no idioma português para expressar que não se está preocupado com o que vai acontecer depois de determinada atitude, demonstrando que a semântica da frase está sendo levado em conta.

Stemming

Porter PORTER (1997) afirma que ”*stemming* consiste em converter cada palavra para seu radical (*stem*). Por exemplo, as palavras ’*learning*’ e ’*learned*’ são ambas convertidas para o *stem* ’*learn*’. Segundo Chaves CHAVES (2004), *stemming* consiste em reduzir as palavras a seu radical, por meio da retirada dos seus afixos.

O propósito, segundo CHAVES (2004), é:

> ”Chegar a um *stem* que captura uma palavra com generalidade suficiente para permitir um sucesso na combinação de caracteres, mas sem perder muito detalhe e precisão. Um exemplo típico de um *stem* é ’conect’ que é o *stem* de ’conectar’, ’conectado’ e ’conectando’. Dois erros típicos que costumam ocorrer durante o processo de *stemming* são *overstemming* e *understemming*. *Overstemming* se dá quando a cadeia de caracteres removida não é um sufixo, mas parte do *stem*. Por exemplo, a palavra

gramática, após ser processada por um *stemmer*, é transformada no *stem* grama. Neste caso, a cadeia de caracteres removida eliminou parte do *stem* correto, a saber 'gramát'. Já *understemming* ocorre quando um sufixo não é removido completamente. Por exemplo, quando a palavra 'referência' é transformada no *stem* 'referênc', ao invés do *stem* considerado correto 'refer'.

Em 1980, Martin Porter PORTER (1997) desenvolveu um algoritmo que tem por objetivo o tratamento de *stemming* para a língua inglesa. Tem sido adaptado para várias línguas latinas, tais como espanhol e português.

3.3.2 Limpeza dos dados

Para construir bons modelos, precisa-se de dados *limpos*. Entretanto, os dados em muitas organizações possuem baixa qualidade. Valores ausentes, valores ilegais, combinações inexistentes e erros de grafia podem alterar os resultados da mineração. Os recursos de transformações e limpeza dos dados (*data cleaning*) aumentam o valor desses dados.

Em NETO; DINIZ (2000), afirma-se que "a qualidade dos dados é essencial para a obtenção de resultados confiáveis. Portanto, dados limpos e compreensíveis são requisitos básicos para o sucesso da mineração". As atividades de obtenção e limpeza dos dados normalmente consomem mais da metade do tempo dedicado ao projeto. Porém, a limpeza dos dados pode evitar que a consolidação dos dados seja distorcida. Geralmente, os erros são pequenos e simples, onde uma letra é adicionada, trocada ou omitida. No entanto, esses erros são difíceis de serem encontrados em um conjunto de dados. Segundo Han & Kamber HAN; KAMBER (2001), as tarefas para limpeza da base são:

1. preencher os valores que estiverem faltando;
2. identificar *outliers*, ou seja, valores distantes da média e retirar esses ruídos dos dados;
3. verificar a consistência dos dados;
4. resolver a redundância causada pela integração dos dados.

3.3.3 Descoberta reativa e pró-ativa de conhecimento

Existem dois modos de descoberta de conhecimento a partir da mineração: reativa ou pró-ativa. Na descoberta reativa, é necessário que se entenda qual o interesse ou objetivo do usuário para limitar o espaço de busca na entrada ou filtrar os resultados na saída LOH; WIVES; OLIVEIRA (2000). O usuário tem uma idéia vaga do que pode ser a solução ou de onde pode encontrá-la. O usuário possui algumas hipóteses iniciais que serão utilizadas para direcionar o processo de descoberta. Neste caso, é necessário que haja algum tipo de pré-processamento, para selecionar atributos ou valores de atributos.

Na descoberta pró-ativa, ao contrário da reativa, a solução do problema é encontrada automaticamente, sem a intervenção do usuário. Segundo Loh & Oliveira LOH; WIVES; OLIVEIRA (2000), uma expressão comum para definir o modo pró-ativo é: ”diga-me o que há de relevante nesse conjunto de dados”.

Neste trabalho é utilizado o modo reativo de descoberta de conhecimento.

3.3.4 Tarefas de Descoberta de Conhecimento em Textos

Qualquer um dos métodos de descoberta de MD, explicados na Seção 3.2 pode ser aplicado em textos. Na literatura, encontram-se as seguintes tarefas de descoberta de conhecimento de textos: classificação e categorização, sumarização, *clustering*, regras de associação e Recuperação de Informação (RI, ou *Information Retrieval*). Nesta dissertação, optou-se pelo método de recuperação de informações para auxiliar no processo de mineração dos textos.

A recuperação de informação auxilia pessoas a encontrar informações relevantes em documentos não estruturados. Este método é particularmente interessante quando da utilização do modo de descoberta reativa, pois neste caso, é necessário que se conheça o interesse ou objetivo do usuário. Sabendo-se qual é o interesse do usuário, é possível preparar um domínio do problema, permitindo diminuir a possibilidade de haver um problema chamado de sobrecarga de informações (*information overload*). A sobrecarga de informação acontece quando o usuário recebe um conjunto muito grande de documentos que não satisfazem a sua pergunta. Um exemplo

típico da utilização da recuperação de informação sem domínio é um sistema de busca (*search engines*) na Web, onde uma pesquisa pode retornar um número imenso de respostas possíveis, obrigando o usuário a fazer uma análise de cada resposta encontrada.

3.3.5 Recuperação de Informação

A Recuperação de Informação estuda o armazenamento e recuperação automática de documentos, que são objetos de dados, geralmente textos. Um Sistema de Recuperação de Informação (SRI), segundo GEY (1992), pode ser estruturado conforme a Figura 3.9. Os componentes de um SRI incluem os documentos, as necessidades do usuário que faz uma consulta e o processo de recuperação que, baseado nas estruturas de dados e da pergunta formulada, recupera uma lista de documentos considerados relevantes.

O processo de indexação envolve a criação de estruturas de dados associadas à parte textual dos documentos. Estas estruturas podem conter dados sobre as características dos termos na coleção de documentos, tais como a freqüência de cada termo em um documento FRAKES; BAEZA-YATES (1992). O processo de especificação da consulta, geralmente, pode ser uma tarefa difícil. Há, com freqüência, uma diferença semântica entre a necessidade de um usuário e o que ele expressa em sua pergunta. Essa diferença é gerada pela limitação do conhecimento do usuário sobre o domínio da pesquisa e pelo formalismo existente na linguagem de consulta. O processo de recuperação gera uma relação de documentos resultados para responder a pergunta formulada pelo usuário. Os índices são construídos, a partir do processo de indexação, para um conjunto de documentos e são utilizados para acelerar esta tarefa. Além disso, a relação de documentos recuperados é classificada em ordem decrescente de um grau de relevância entre o documento e a consulta formulada.

3.3.6 Avaliação de Sistemas de Recuperação de Informação

Os SRIs podem ser avaliados através de consultas que fazem parte de uma conjunto referência. Para este conjunto referência, é fornecido um conjunto ideal de do-

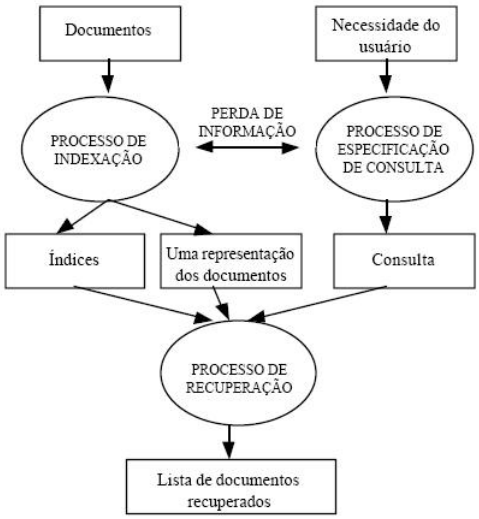

Figura 3.9: Componentes de um sistema de recuperação de informação Fonte: Gey (1992)

cumentos resposta. Este conjunto ideal de documentos resposta é criado por especialistas nos temas envolvidos. Assim, é possível avaliar o SRI através da comparação das respostas geradas por este sistema e o conjunto ideal de respostas. Feito isto, o conjunto resultado é examinado e comparado com o conjunto ideal, obtendo-se dois índices de avaliação: precisão (*precision*) e revocação (*recall*).

A precisão avalia se os documentos recuperados são relevantes e a revocação avalia se os documentos relevantes são recuperados. Sejam:

- `N` é conjunto de documentos relevantes identificados por especialistas
- `I` é o conjunto de documentos recuperados pelo sistema
- `P` é a precisão do sistema
- `R` é a revocação do sistema

A *precisão* é dada por 3.1. A *revocação* é dada por 3.2.

$$P = \frac{|N \bigcap I|}{|I|} \tag{3.1}$$

$$R = \frac{|N \bigcap I|}{|N|} \tag{3.2}$$

Por exemplo, para avaliar a recuperação de informação, se:

- os documentos relevantes são: {2, 33, 55, 23, 99}; e
- um sistema recupera o conjunto resultado: {55, 25, 99, 6, 2, 8, 21, 7, 29, 14, 33, 40, 36, 22, 81, 23 }.

O nível de revocação de 20% é atingido ao encontrar o primeiro documento relevante (55). A precisão é de $1/1 = 100\%$.

Para revocação de 40%, a precisão é igual a $2/3 = 66\%$.

3.3.7 Modelos Clássicos de Recuperação de Informação

Os modelos clássicos utilizados no processo de RI são: booleano, vetorial e probabilístico. Eles apresentam estratégias de busca de documentos relevantes para uma consulta. Estes modelos consideram que cada documento é descrito por um conjunto de palavras chaves, chamadas termos de indexação. Associa-se a cada termo de indexação $\mathtt{t}_i$ em um documento $\mathtt{d}_j$, um peso $\mathtt{w}_{ij} \geq 0$, que quantifica a correlação entre os termos e o documento. Além dos modelos clássicos, modelos muito mais avançados de recuperação de informação têm sido propostos ao longo dos anos, dentre estes, destacam-se modelos baseados em bases de conhecimento, lógica *fuzzy* e redes neurais.

Modelo Booleano

O modelo booleano é um dos modelos clássicos que considera uma consulta como uma expressão booleana convencional. Os termos são conectados através dos operadores lógicos `AND`, `OR` e `NOT`. Um documento é considerado relevante ou não relevante a uma consulta. Portanto, não existe resultado parcial e não há informação que permita a ordenação do resultado da consulta. Este modelo é muito mais utilizado para recuperação de dados do que para recuperação de informação. Esse modelo é apropriado para quem entende de álgebra booleana, mas o usuário, na maioria dos casos, não entende.

As consultas são construídas como uma combinação dos conjuntos que descrevem todas as possibilidades para o conjunto resposta da consulta, chamadas de *min-termos*. Geralmente, para n termos, temos: $k = 2^n$ min-termos e 2^k consultas. Por exemplo, para n=3 termos são k=8 min-termos e 2^8=256 possíveis consultas.

O tamanho da base de dados afeta tanto as estratégias de consultas, quanto os resultados obtidos. Este modelo é altamente utilizado em sistemas comerciais.

Vantagens do modelo booleano:

- a expressividade é completa, se o usuário souber exatamente o que quer;
- o modelo é facilmente programável e exato.

Desvantagens do modelo booleano:

- as pessoas lidam com conhecimento parcial;
- a saída pode ser nula, ou haver *overload*;
- a saída não é ordenada.

Algumas formas de tentar melhorar os resultados gerados pelo modelo booleano são as seguintes.

- Atribuição de pesos aos termos. A carga semântica dos termos é completamente diferente, quando se tem, por exemplo, uma consulta com dois termos ou duas consultas distintas com um termo cada.
- Utilização de conjuntos *fuzzy*. A pertinência ou não de um elemento a um conjunto varia entre 0 e 1, não é exata.
- Categorização da recuperação de informação. Dividir a consulta em classes e conceitos, tentar encontrar os documentos baseados nos conceitos.
- *Passage retrieval*. O conjunto de termos a ser procurado deve aparecer o mais próximo possível, por exemplo, em uma mesma página (uma possível passagem). É uma técnica mais eficiente que a de RI, porém muito mais difícil de ser implementada. A proximidade é importante.

- Ordenação da saída. Uma vez estabelecida alguma forma de determinar que termos são mais importantes para determinada consulta, é possível ordenar os resultados.

Modelo Vetorial

O modelo espaço-vetorial (ou, simplesmente, vetorial) foi desenvolvido por Gerard Salton SALTON et al. (1997), para ser utilizado num SRI chamado SMART. No modelo vetorial, cada documento é representado como um vetor de termos. Cada termo possui um valor associado, que indica o grau de importância (peso ou *weight*) deste no documento. Formalmente, cada documento possui um vetor associado que é constituído por pares de elementos na forma $\texttt{(palavra}_1\texttt{, peso}_1\texttt{)}$, $\texttt{(palavra}_2\texttt{, peso}_2\texttt{)}$,..., $\texttt{(palavra}_n\texttt{, peso}_n\texttt{)}$.

Os pesos são usados para computar a similaridade entre cada documento armazenado e uma consulta feita pelo usuário. Segundo SALTON; MCGILL (1983), o peso de um termo em um documento pode ser calculado de diversas formas. Geralmente, esses métodos de cálculo de peso se baseiam no número de ocorrências do termo no documento (freqüência).

Vantagens do modelo vetorial:

- a atribuição de pesos aos termos melhora o desempenho;
- o modelo é baseado em similaridade, que é melhor que a exatidão do modelo booleano;
- os documentos são ordenados de acordo com seu grau de similaridade com a consulta.

Desvantagens do modelo vetorial:

- a ausência de ortogonalidade entre os termos (podem-se encontrar relações entre termos que aparentemente não têm nada em comum);
- o modelo é generalizado;
- um documento relevante pode não conter termos da consulta.

Modelo Probabilístico

O modelo probabilístico possui esta denominação justamente por trabalhar com conceitos provenientes da área de probabilidade e estatística. Neste modelo, os termos indexados dos documentos e das consultas não possuem pesos pré-definidos. A ordenação dos documentos é calculada pesando dinamicamente os termos da consulta com relação aos documentos. O modelo é baseado no princípio da ordenação probabilística (*Probability Ranking Principle*), onde busca-se saber a probabilidade de um documento $\mathtt{D}$ ser ou não relevante para uma consulta $\mathtt{C}_a$. Tal informação pode ser obtida se se assume que a distribuição de termos na coleção seja capaz de informar a relevância provável para um documento qualquer da coleção.

3.3.8 Ferramentas de Mineração de Texto

Segundo FURTADO (2004), as principais ferramentas de mineração de texto são:

TextAnalyst Fornecedor: *Megaputer Intelligence, Inc.*

A ferramenta disponibiliza sumarização de texto. Trabalha com texto não-estruturado, como artigos e informes e produz um sumário preciso. Está disponível em diversas línguas, como inglês, francês, alemão, espanhol, italiano, russo e holandês.

S-Miner Fornecedor: Coordenação dos Programas de Pós-graduação de Engenharia, COPPE, UFRJ

O S-Miner RODRIGUES; OLIVEIRA; SOUZA (2004) foi desenvolvido para minerar textos. Inicialmente, o texto é submetido ao algoritmo de geração de palavras (*tokens*). A tokenização consiste na identificação de palavras. Esta técnica sugere que os *tokens* sejam definidos como uma *string* de caracteres alfanuméricos sem espaços. Após a quebra do texto em *tokens*, o processo prossegue com a retirada das palavras que não possuem relevância significativa no texto, as chamadas *stopwords*. Esta lista de palavras irrelevantes é fortemente dependente da língua e do contexto utilizados. O S-Miner suporta inglês e português (do Brasil).

Eureka Fornecedor: Leandro Krug Wives, Universidade Federal do Rio Grande do Sul

A ferramenta Eureka WIVES (1999b) se propõe a analisar um conjunto de textos não formatados e a identificar e agrupar aqueles considerados semelhantes semanticamente. Dentre as facilidades oferecidas pela ferramenta estão a possibilidade de configuração de listas de *stopwords* e da escolha de um entre quatro algoritmos de agrupamento.

$\boldsymbol{Clear\ Research\ Suite}$ Fornecedor: *ClearForest Corporation*

A ferramenta faz aplicações de análise, extração de características, visualização de inter-relações complexas entre empresas, pessoas, eventos, etc., no mundo dos negócios. O motor de extração de informação pode, dinamicamente, identificar relacionamentos entre pessoas, companhias e grandes repositórios de textos não estruturados, incluindo novos fontes, páginas da Web e informes internos.

$\boldsymbol{BrandPulse}$ Fornecedor: *Intelliseek Inc.Planetfeedback*

A ferramenta permite buscar opiniões e tendências na Internet, monitorando bases de dados públicas, quadros de discussão, opiniões, boatos e oportunidades em tempo real. Identifica às mudanças de necessidades do consumidor e suas opiniões.

$\boldsymbol{TrackEngine}$ Fornecedor: *NexLabs Pte Ltd.*

A ferramenta monitora *Web sites* corporativos, salas de bate-papo e quadros incorporados em mensagens. Pode alertar o usuário pró-ativamente de qualquer novo conteúdo, através de um *e-mail* alerta.

$\boldsymbol{Strategy}$ Fornecedor: *Strategy Software, Inc.*

A ferramenta cria uma base multidimensional para a análise eficaz e a tomada de decisão, fornecendo ao usuário um meio de organizar informações diferentes de maneira estruturada.

$\boldsymbol{PlanBee}$ Fornecedor: *Thoughtshare Communication Inc.*

A ferramenta permite consolidar *Web pages*, documentos texto, arquivos de imagem, arquivos PDFs, arquivos de áudio e arquivos de vídeo.

Capítulo 4

A Ferramenta TextEaD

O papel do professor nos AVAs é gerenciar o conteúdo instrucional e acompanhar a aprendizagem dos alunos. Criar ferramentas inteligentes que facilitem e auxiliem o professor no seu papel dentro do ambiente, e automatizem o maior número de tarefas possíveis é significativo e fundamental. Tanto a mineração de dados quanto a mineração de textos aplica técnicas de Inteligência Artificial e são consideradas formas inteligentes de descobrir conhecimento dentro de bancos de dados. Ao aplicar mineração, podemos obter resultados em forma de conhecimento que ajudem o professor em suas tarefas.

A seguir, este capítulo apresenta o objetivo da ferramenta TextEaD, diagrama de funcionamento, sua arquitetura geral, princípios de projeto e, na busca por soluções, as duas abordagens. Na primeira versão, foi aplicada mineração de dados utilizando classificação com árvore de decisão. A segunda versão direcionou a ferramenta à aplicação de técnicas de mineração de textos. O objetivo final das duas abordagens é apresentar ao aluno a melhor resposta encontrada para sua dúvida.

4.1 Objetivo

O objetivo deste trabalho é desenvolver uma ferramenta inteligente para assistir as dúvidas dos alunos, substituindo, parcialmente, o professor no ambiente. A ferramenta TextEaD utiliza técnicas de mineração para recuperar de uma base de textos por disciplina fornecida pelo professor, as melhores respostas possíveis às perguntas submetidas pelos alunos na linguagem natural. A ferramenta pode ser considerada um assistente inteligente de dúvidas dentro do contexto de um sistema ITA.

4.2 Princípios de Projeto

O desenvolvimento da ferramenta adotou o paradigma de Software Livre. Assim, as tecnologias envolvidas são de código aberto (*open source*), utilizando a licença GPL (*General Public License* ou Licença Pública Geral), visando tornar seu uso, estudo, aperfeiçoamento e distribuição possíveis e incentivados.

4.3 Modelagem do Funcionamento

A Figura 4.1 mostra o diagrama de atividades ou fluxograma do assistente de dúvidas dentro de um AVA. O funcionamento de assistente é inicializado a partir da submissão de uma dúvida por um aluno. A partir da solicitação, pelo aluno, é executado o assistente para auxiliá-lo. Caso afirmativo, é realizada a análise léxica que permite identificar os *tokens* ou palavras chaves da pergunta do aluno. As palavras chaves da pergunta são processadas, a fim de encontrar resultados nos textos armazenados na base. Os resultados são apresentados ao aluno como as respostas a sua dúvida. Caso as palavras chaves não forem encontradas na base de textos, um relatório é enviado ao professor com a dúvida do aluno para informar a ausência de resposta àquela pergunta.

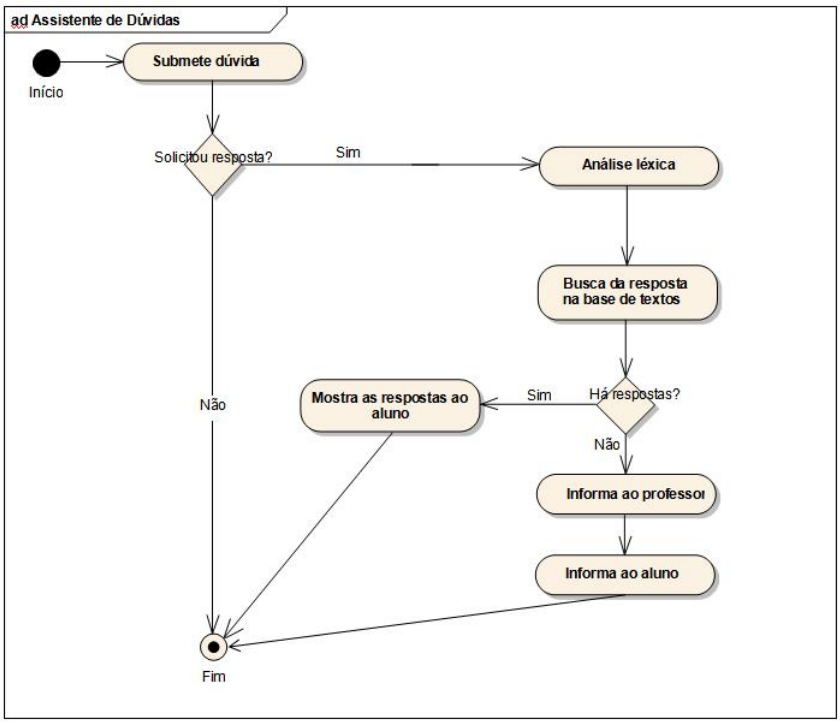

Figura 4.1: Diagrama de atividade do assistente de dúvidas

4.4 Arquitetura Geral

A Figura 4.2 apresenta a arquitetura geral da ferramenta. A ferramenta TextEaD é aplicada em duas etapas. Na primeira etapa, o módulo de inteligência constrói a estrutura de índice da mineração. Esta estrutura é uma representação lógica dos textos armazenados na base de textos, criada a partir da mineração, tanto de dados quanto de textos, que permitirá a recuperação eficiente das respostas a partir da pergunta do aluno.

A segunda etapa é iniciada na aplicação da ferramenta. A aplicação dentro do ambiente virtual tem duas interfaces: a interface do aluno e a interface do professor. A interface do aluno é onde o aluno vai submeter sua pergunta e os resultados serão apresentados. A dúvida é expressa em linguagem natural (português). Uma vez submetida, é realizada a análise léxica, que permite identificar as palavras relevantes da pergunta do aluno, descartando as palavras que se encontram na base das *stopwords*. As palavras chaves são processadas utilizando a estrutura gerada no módulo de inteligência e as melhores respostas à dúvida do aluno são recuperadas.

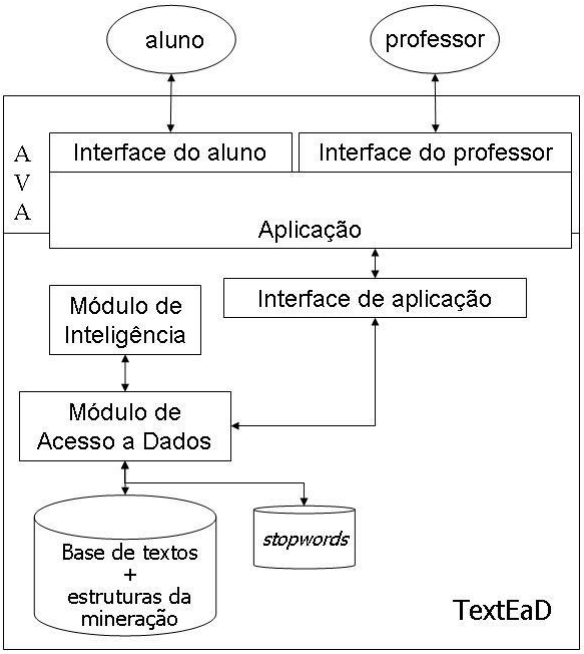

Figura 4.2: Arquitetura geral da ferramenta TextEaD

A interface do professor permite a entrada de dados. A entrada de dados consiste na inserção dos textos, classificados por disciplina, pelo professor atuando como especialista da disciplina. Também, essa interface é utilizada para enviar relatórios ao professor, caso a dúvida não consiga ser respondida com os dados da base. Este processo é interativo e iterativo, pois o professor precisa atualizar a base com os novos textos que respondam as dúvidas, e isto precisa ser feito toda vez que encontrar uma dúvida sem resposta. O objetivo é manter um *feedback* com o professor, que será responsável por armazenar as respostas.

4.5 Abordagem 1: TextEaD com Mineração de Dados

Esta primeira versão da ferramenta aplica mineração de dados sobre a base de textos. A base de textos é composta pelo textos, o código e cinco palavras chaves associadas, como mostra a Figura 4.3. As palavras chaves são definidas pelo professor, especialista da disciplina. A ordem de relevância das palavras, depende do ponto de vista da professor. Para o funcionamento correto da ferramenta, a ordem das palavras deve ser respeitada, sendo a primeira o conceito mais genêrico até a quinta com o mais específico.

Tanto a definição das cinco palavras chave para cada texto na base, quanto a atualização da base de textos são responsabilidades do professor da disciplina.

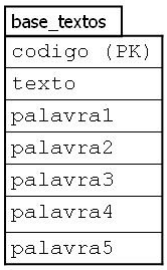

Figura 4.3: Estrutura da base de textos

Além da base de textos, uma base auxiliar é criada para armazenar as palavras descartáveis ou *stopwords*. Esta segunda base oferece suporte à análise da pergunta em linguagem português para a aplicação.

4.5.1 Arquitetura

A Figura 4.4 mostra a arquitetura em três módulos.

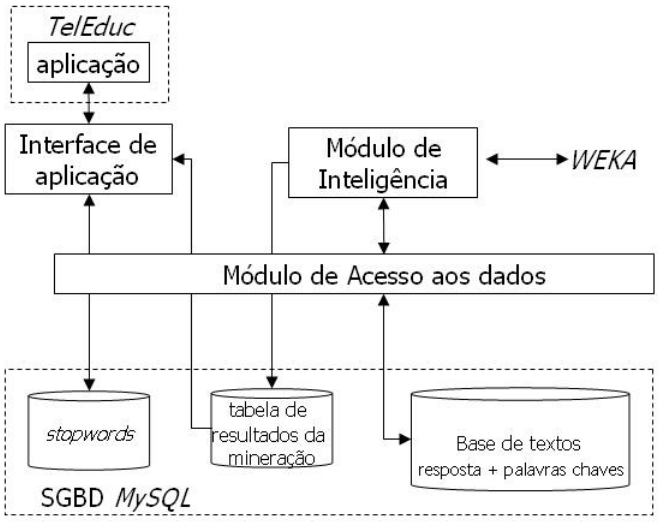

Figura 4.4: Arquitetura específica à primeira abordagem

A aplicação da ferramenta foi desenvolvida sobre o ambiente TelEduc. O primeiro e o segundo módulo foram desenvolvidos em Java (JDK 1.5.0.03) por ser tecnologia aberta, orientada a objeto e altamente portável. O terceiro módulo foi desenvolvido

em PHP (*PHP: Hypertext Preprocessor*) por haver a necessidade de ficar integrada ao AVA escolhido.

É bom destacar que, a pesar de a aplicação ser desenvolvida sobre o ambiente TelEduc, ela pode ficar em qualquer ambiente de EaD que possua integração à linguagem PHP. A base de textos foi armazenada no SGBD PostgreSQL. A seguir são descritos detalhes de implementação de cada módulo.

Módulo de Acesso aos Dados

A função deste módulo é acessar e preparar a fonte de dados para o Módulo de Inteligência. A ferramenta WEKA tem um formato próprio do arquivo de entrada dos dados. Como parte deste módulo, um aplicativo de tradução foi desenvolvido que: (1) lê a base armazenada no banco de dados e (2) cria um arquivo de saída no formato ARFF do WEKA. O módulo de acesso a dados tem a função de servir de interface a todas as transações (consultas e armazenamento de dados) para SGBD. Tanto o módulo de inteligência quanto a interface de aplicação utilizam o módulo de acesso aos dados. Este componente utiliza os *drivers* disponíveis na interface JDBC (*Java DataBase Connectivity*) da linguagem Java para acesso a diversos sistemas gerenciadores de bancos de dados.

Módulo de Inteligência

Este módulo tem o objetivo de fornecer o suporte inteligente da ferramenta. Para isto, classifica as palavras chaves da base de textos com apoio da ferramenta WEKA. Qualquer dos algoritmos de classificação, disponíveis pelo WEKA, pode ser utilizado para organizar os dados da base. Por exemplo, os dados podem ser classificados de acordo com um algoritmo de árvore de decisão. A estrutura gerada pela mineração vai ser utilizada, mais tarde, pela aplicação para responder as dúvidas. Após o estudo e testes de vários algoritmos de classificação, implementados no WEKA, escolheram-se os algoritmos de árvore de decisão. A Figura 4.5 mostra a interface do módulo de inteligência com os diferentes algoritmos disponíveis para aplicar a mineração: `ID3`, `J48`, `NBTree`, dentre outros.

As cinco palavras chaves armazenadas na base de textos são utilizadas como os atributos para aplicar a mineração de dados. Os dados na base de textos

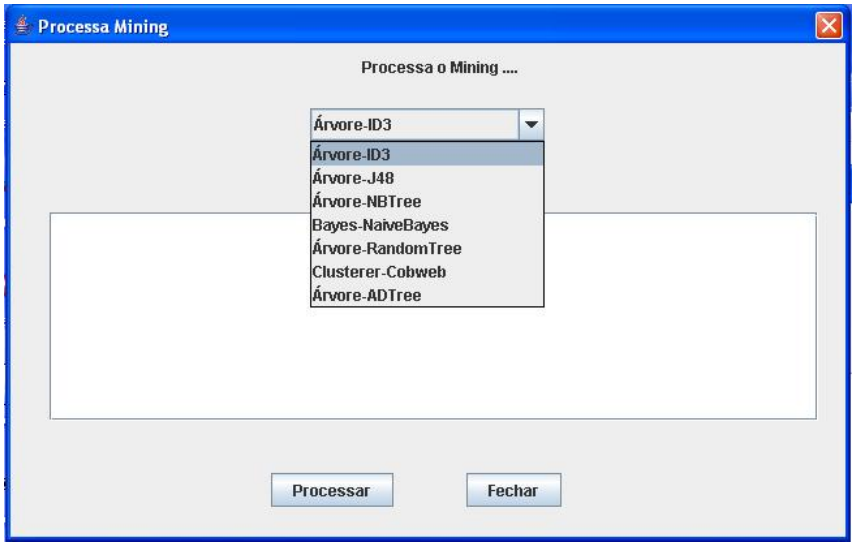

Figura 4.5: Módulo da inteligência

são o conjunto de treinamento para criar a árvore de decisão. Os dados são solicitados e preparados pelo módulo de acesso aos dados em formato ARFF para uso da ferramenta WEKA. O resultado obtido em WEKA é apresentado na tela, possibilitando ao especialista validar o resultado da técnica de mineração de dados escolhida. Confirmada sua escolha, este resultado é armazenado em uma tabela, chamada de tabela de resultados da mineração. Esta tabela é utilizada pela aplicação para pesquisar as respostas às dúvidas dos alunos.

Interface de aplicação

Este módulo funciona como interface entre a ferramenta e a aplicação inserida no ambiente virtual de aprendizagem. Sua função é ser a intermediária entre a submissão de uma dúvida e a obtenção das respostas, fazendo a tradução da dúvida em palavras chaves e buscando a similaridade dessas palavras na árvore de decisão armazenada. A aplicação foi desenvolvida dentro do ambiente TelEduc, inserindo algumas poucas linhas de código em PHP, apresentadas no Apêndice ??. A tela da sala de bate-papo do TelEduc, apresentada na Figura 4.6, mostra, na parte inferior, a aplicação ou assistente às dúvidas do TextEaD.

Espera-se que a construção da frase de uma pergunta esteja gramaticalmente correta em português. Aceitam-se frases como: *o que é um objeto?*, *objeto* e *orientação a objeto?*. O algoritmo implementado na interface de aplicação

Figura 4.6: Página do bate-papo do TelEduc

possibilita encontrar o melhor conjunto de respostas existente na base de textos, caso exista. Com a aplicação, o aluno passa a ter a sua disposição um campo na janela no qual pode digitar uma dúvida qualquer. A aplicação envia uma mensagem para a interface e a ferramenta:

1. Recebe a pergunta em linguagem natural.
2. Retira as palavras que estão na tabela de palavras descartáveis; as palavras que ficaram são consideradas as palavras chaves, que devem ser utilizadas na pesquisa. Este processamento é descrito a seguir.
   (a) *Limpeza de pontos e símbolos*: varre a pergunta procurando caracteres na faixa de 1 a 31 e de 127 a 191 e símbolos como ?.
   (b) *Limpeza do texto*: já com a pergunta sem pontuação nem símbolos, as *stopwords* são removidas.
3. Constrói uma consulta, como a conjunção de restrições com as palavras chave e baseada no resultado da classificação armazenada na tabela de resultados da mineração. As palavras chave representam os valores sobre os atributos da árvore, mapeadas uma a uma a partir da raiz.
4. Executa esta consulta. Caso não encontre resultados, subtrai do predicado a última condição (uma das palavras chaves da pergunta). Isto significa subir um nível na árvore. Assim, percorre a árvore de decisão, das folhas para a raiz (de baixo para cima), até encontrar um conjunto não vazio de

respostas. Se ao chegar ao nível mais alto (a raiz), não encontra resultados, retorna o conjunto vazio para a aplicação e envia ao professor uma mensagem com o texto da dúvida para este atualizar a base de textos.

A aplicação retorna para o aluno a resposta ou, caso não encontrada, um texto que informe que não foi achada resposta para a pergunta submetida.

4.5.2 Exemplo

A Figura 4.7 mostra um exemplo de árvore que pode ter sido gerada para as cinco palavras chave, sendo a raiz da árvore o atributo `palavra5`.

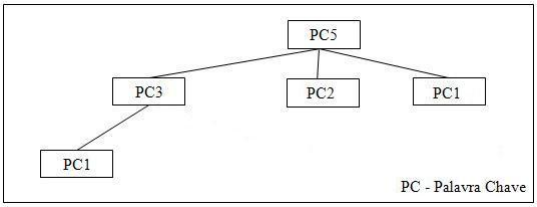

Figura 4.7: Exemplo de árvore de decisão

Neste exemplo, a palavra chave 5 é a primeira a definir a classificação das respostas. As arestas da raiz levam às palavras chave 3, 2 e 1, no mesmo nível de classificação. A partir da raiz, a interface de aplicação constrói as consultas com cada uma das arestas da árvore, para buscar a resposta correspondente à dúvida do aluno. A primeira ramificação dessa árvore pode ser traduzida na consulta SQL seguinte, sendo `var_pc1`, `var_pc2` e `var_pc3` os valores das palavras chave extraídas da dúvida.

```
select texto
from base_textos
where palavra5 = :var_pc1
and palavra3 = :var_pc2
and palavra1 = :var_pc3
```

Caso não seja encontrada a resposta, a consulta é refeita, retirando a última restrição, neste caso:

```
and palavra1 = :var_pc3
```

4.6 Desvantagens

Esta seção pretende discutir as desvantagens da ferramenta em sua primeira versão, justificando, assim, o porquê da migração para uma segunda versão aplicando mineração de textos.

A utilização da mineração de dados mostrou-se eficiente como um mecanismo para auxiliar o algoritmo de recuperação de respostas. Porém, nos testes realizados ficou demonstrado a vulnerabilidade da ferramenta quanto à correta ordem de definição das palavras chaves para os textos na base. A inserção das palavras chave é um processo subjetivo, que depende da ordem definida pelo professor. No exemplo seguinte, cadastraram-se três palavras chave por texto. A Figura 4.8(a), apresenta um exemplo de árvore criada e armazenada no módulo de inteligência, a partir da análise das palavras cadastradas na base. A ordem das palavras-chaves é, segundo esperado pela ferramenta, da mais genérica à mais específica: *paradigma*, *orientação* e *objeto*. O resultado aponta aos textos de código 1 e 3.

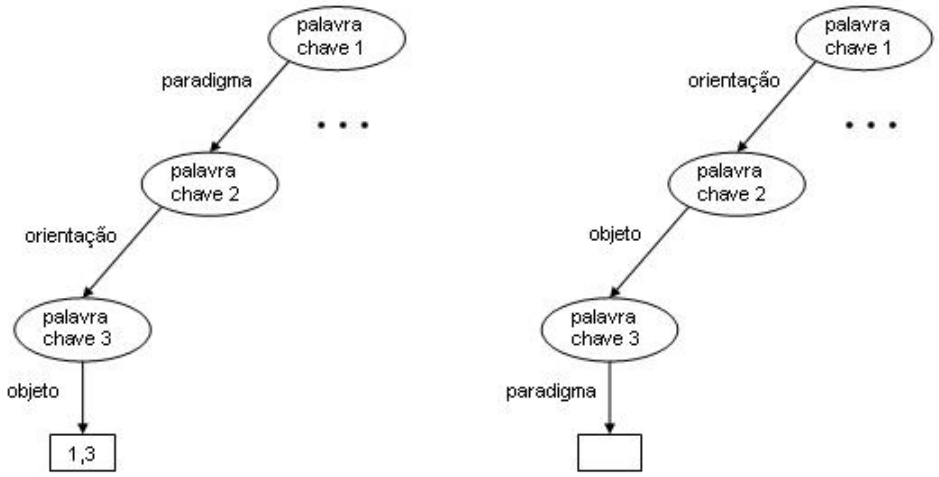

Figura 4.8: (a) Entrada de dados correta das palavras chaves (b) Entrada de dados incorreta das palavras chaves

No entanto, o especialista poderia ter definido a seguinte ordem que pode ser observada na Figura 4.8(b): *orientação*, *objeto* e *paradigma*. Esta situação resultaria em problemas de classificação, pois a raiz da árvore deve conter a palavra mais genérica (isto é, espera-se que seja a palavra chave *paradigma*).

Quando um aluno submeter a pergunta *o que é paradigma de orientação a objeto*,

a interface de aplicação extrai as palavras chaves da pergunta como sendo:

`palavra chave 1`: *paradigma*
`palavra chave 2`: *orientação*
`palavra chave 3`: *objeto*

e constrói a consulta que corresponde à primeira ramificação da árvore da Figura 4.8(a). No entanto, não seria o caso para a árvore da Figura 4.8(b). Observe-se que caso não obtenham-se resultados, exclui-se da consulta a palavra mais específica (*objeto*) e submete-se novamente a consulta com as duas primeiras palavras.

Como conseqüência, a ferramenta fica dependente do critério do professor. No entanto, se o número de erros deste tipo for pequeno em relação ao tamanho da base de textos, não haverá problemas. Em outro caso, esse tipo de situação irá representar um problema na eficácia da ferramenta.

Os testes realizados indicaram que o número de textos com erros de definição das palavras chave influenciou o suficiente para gerar a árvore de decisão errada. Os bons resultados, apresentados no próximo capítulo, foram obtidos com a situação manualmente controlada, tomando-se o devido cuidado no cadastramento das palavras. Esse ponto fraco motivou o estudo de soluções alternativas que pudessem fornecer melhores respostas. Essas pesquisas levaram a estudar e aplicar a mineração de texto.

4.7 Abordagem 2: TextEaD com Mineração de Textos

Esta segunda versão da ferramenta aplica mineração de textos sobre a base de textos. A base de textos (Figura 4.9) é, dessa vez, composta pelos textos, seu título e seu respectivo código. Nesta versão, é possível descartar a necessidade do professor precisar definir ou fazer qualquer outra operação com as palavras chaves. As palavras chaves são descobertas através do processamento do número de vezes que cada palavra é encontrada no texto. A técnica utilizada é a recuperação de informação, focando no modo de descoberta reativa, uma vez que se assume que a base de textos

está formada por textos específicos à matéria do curso em que o aluno está inserido. A base de textos continua sendo mantida pelo professor da disciplina. O modo de descoberta reativa foi o escolhido pelos seguintes motivos:

- a base de textos é formada por assuntos específicos a uma determinada matéria;
- há conhecimento do que é interessante para o aluno, exposto na formalização da dúvida submetida;
- não há o risco de sobrecarga de informações, uma vez que a base de textos é formada por assuntos específicos.

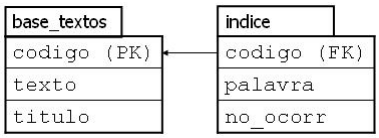

Figura 4.9: Estrutura da base de textos

A partir desta seção, explicam-se apenas aqueles aspectos que possuem alguma diferença com a versão anterior.

4.7.1 Arquitetura

A Figura 4.10 mostra a arquitetura da ferramenta, nesta abordagem, em quatro módulos. A base de *stopwords*, além de dar suporte ao processamento da pergunta do aluno, dá suporte ao processamento da linguagem natural utilizada nos textos das respostas armazenadas na base de textos.

Todos os módulos da ferramenta foram desenvolvidos na linguagem PHP 5 (*PHP: Hypertext Preprocessor*), utilizando o SGBD MySQL, na versão 5.0, reusando o código da primeira versão quando possível.

A Figura 4.11 mostra a interface principal da ferramenta TextEaD.

O módulo de acesso aos dados e a interface de aplicação mantém as mesmas funções e interfaces da primeira versão da ferramenta. O módulo de manutenção e de inteligência e a recuperação de respostas na interface de aplicação são descritos a seguir.

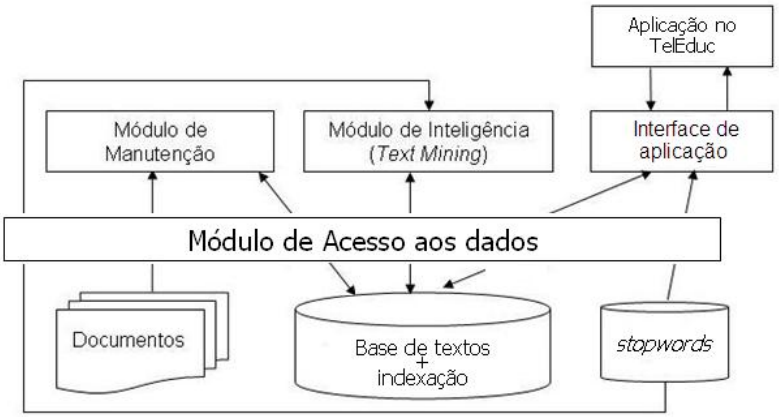

Figura 4.10: Arquitetura específica à segunda abordagem

Figura 4.11: Tela Principal da ferramenta TextEaD

Módulo de Manutenção

A função deste módulo é possibilitar a administração da base de textos, através da inclusão, alteração, consulta e exclusão de textos. A interface do módulo é apresentada na Figura 4.12. As Figuras 4.13 e 4.13 mostram as interfaces para consulta dos textos existentes na base e para modificação ou exclusão de textos.

É importante destacar como é fundamental incluir conteúdo correto e de forma correta na base de textos. O conteúdo correto é a base para o sucesso da mineração. Portanto, deve haver uma consciência por parte dos professores da necessidade de utilizar e manter corretamente o conteúdo da base de textos.

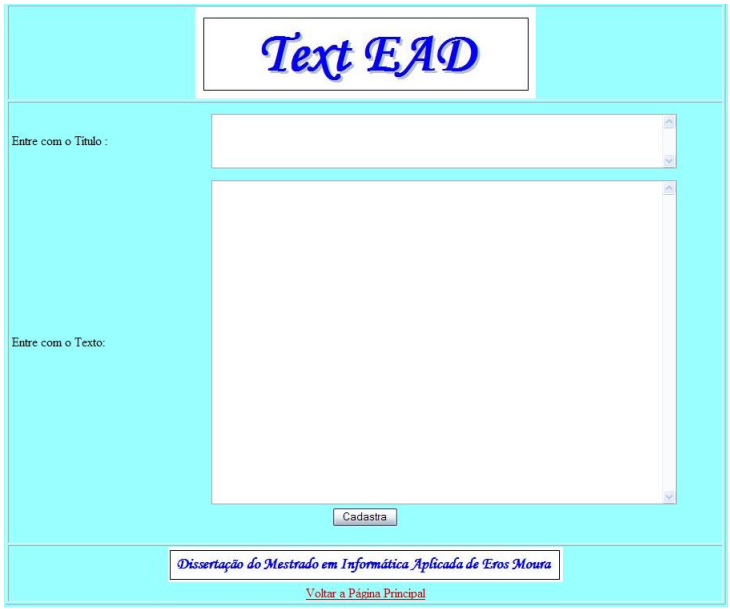

Figura 4.12: Tela de Cadastro na Base de Textos

A possibilidade de modificar dados sobre a base de textos está associada a níveis de prioridades estabelecidos para os usuários, que garantem o tipo de acesso aos dados.

Módulo de Inteligência

Neste módulo, os textos da base serão processados para permitir a recuperação da melhor resposta possível, sempre que o conteúdo da base e a pergunta estejam no idioma português. Para cada texto é gerada uma relação de palavras e o número de vezes que a mesma aparece no texto, criando assim uma estrutura de índices. Esta estrutura será utilizada no mecanismo de recuperação de

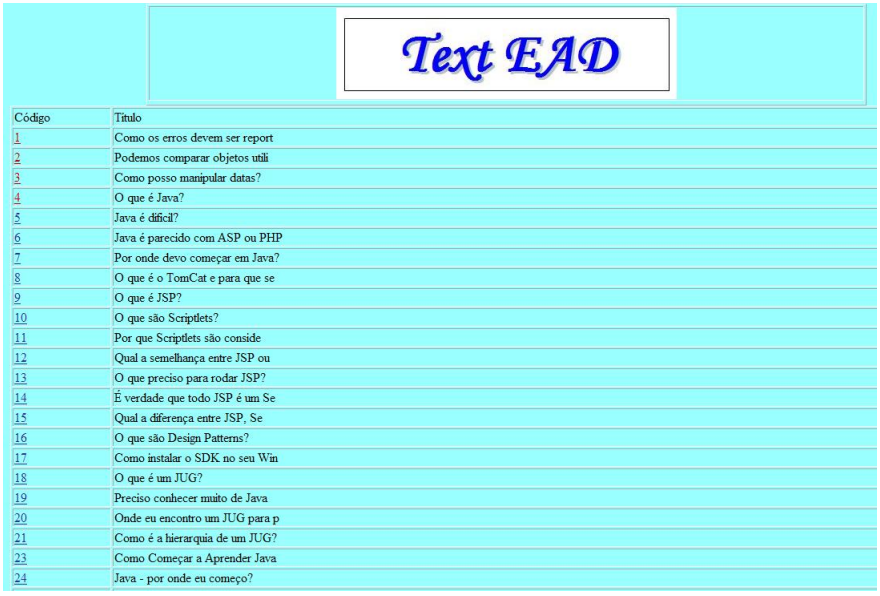

Figura 4.13: Tela de Consulta na Base de Textos

Figura 4.14: Tela de Modificações na Base de Textos

informação. Os passos do processamento dos textos são descritos a seguir.

1. *Limpeza dos índices*: no caso de existência de índices antigos é feita uma limpeza na estrutura. A cada aplicação da mineração de texto é feita uma re-leitura de toda a base. O tempo de processamento está diretamente relacionado ao tamanho da base de textos anterior e à configuração de *hardware/software* que está sendo utilizada.
2. *Leitura das stopwords*: neste passo são lidas todas as palavras cadastradas como *stopwords* na base e colocadas em memória em um vetor.
3. *Leitura dos documentos*: neste passo ocorre a leitura de todos os documentos existentes na base. O tempo de processamento está diretamente relacionado ao tamanho da base de textos anterior e à configuração de *hardware/software* que está sendo utilizada. Para cada documento:
   (a) *Limpeza de pontos e símbolos*: cada documento é varrido à procura de caracteres que estão na faixa de 1 a 31 e de 127 a 191, além da procura por símbolos como:

   \ . ? ; * (") - < > = + / % | & ^ ~

   (b) *Limpeza do texto*: já com o documento sem pontuação e outros símbolos, todas as *stopwords* encontradas no texto são removidas. É importante ressaltar a importância de uma boa lista de *stopwords*, pois do contrário, palavras sem significado para pesquisa serão indexadas.
   (c) *Indexação do texto*: este é o passo final do algoritmo. Neste ponto, o documento tornou-se um conjunto finito de palavras com significado para o contexto: a pergunta feita pelo aluno. Para realizar a indexação, a primeira ação é verificar o número de ocorrências da palavra em um determinado texto, incluindo seu título. O índice é criado com a palavra e o número de ocorrências da palavra no documento (Figura 4.9).

É importante destacar que o algoritmo determina o número de ocorrências das palavras por texto, possibilitando assim uma análise muito mais precisa no momento da pesquisa. Ao completar o processamento, a ferramenta apresenta na tela as estatísticas do número de textos analisados, número de palavras indexadas no título e número de palavras indexadas no texto.

Interface de aplicação

O ponto diferente do processamento na interface de aplicação está na recuperação das respostas. A seleção dos documentos que serão retornados é um ponto fundamental para o sucesso do trabalho, mas que depende da entrada de dados, a lista de *stopwords* e uma pergunta razoavelmente bem elaborada. Espera-se que a construção da frase de uma pergunta esteja gramaticalmente correta em português.

O algoritmo de seleção executa os seguintes passos:

1. para cada palavra chave da dúvida do aluno, constrói-se uma consulta para verificar quais são os documentos que possuem a maior número de ocorrências da palavra;
2. tenta encontrar a resposta, utilizando a conjunção de todos as palavras no predicado da condição da consulta;
3. seleciona até três documentos com o maior número de ocorrências das palavras, combinando todas as palavras;
4. se não encontrar resultados, fazer a retirada de palavras da consulta, e tentar de novo até encontrar a resposta. O critério de retirada das palavras começa com a palavra com menor ocorrência na base de textos e termina naquela com maior ocorrência;
5. se chegando à última retirada, a resposta não foi encontrada, o algoritmo retornará a falha na busca e na tela será mostrado que não foi possível encontrar uma resposta para a pergunta feita.

A aplicação no TelEduc apresenta as respostas na interface ilustrada na Figura 4.15.

Um fator importante no processamento é o tempo de resposta. De nada adiantaria um algoritmo que retornasse os melhores documentos em um tempo maior do que o aceitável para a situação. Como o contexto é um ambiente virtual *on-line*, obter uma resposta após um minuto de processamento seria inaceitável. O próximo capítulo apresenta a descrição do ambiente de validação de ambas as abordagens da ferramenta TextEaD, assim como, os testes realizados e os resultados obtidos.

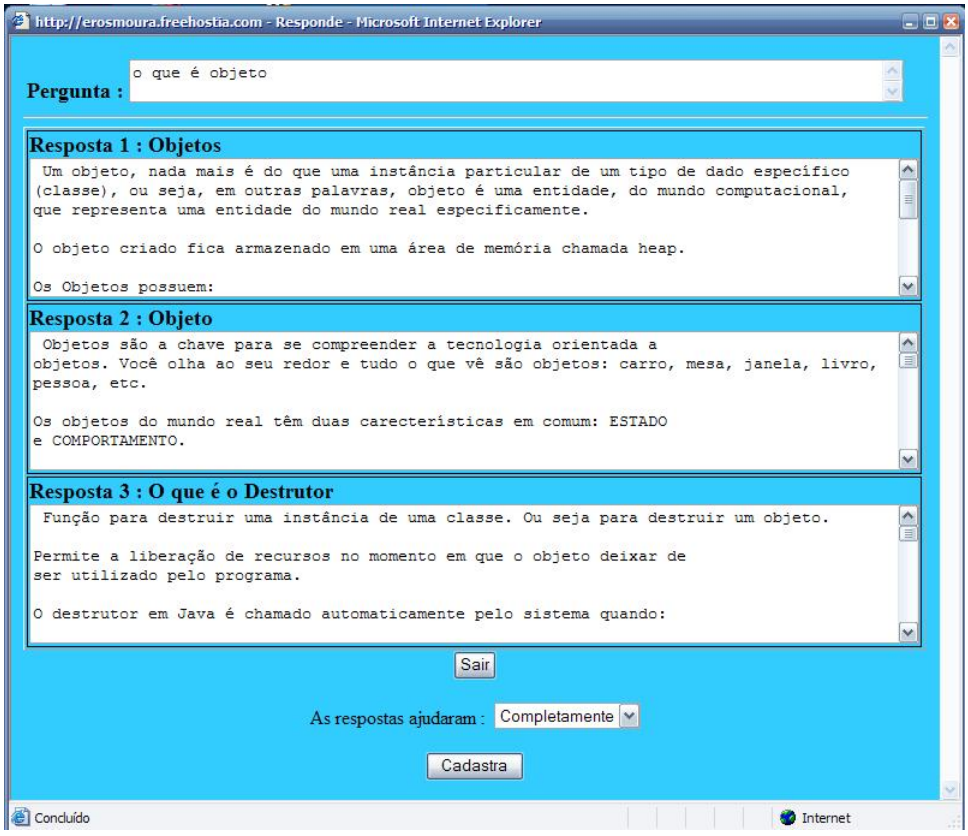

Figura 4.15: Resposta da ferramenta

Capítulo 5

Testes e Resultados

Com o objetivo de validar a eficácia da ferramenta, instalaram-se o TelEduc, junto com a aplicação e a ferramenta TextEaD, no Centro Universitário São Camilo - ES. A participação dos alunos foi opcional, através de convite, para complementar conteúdos não abordados de forma convencional em sala de aula.

O ambiente criado é formado por uma máquina Pentium IV 2.8MHz, com 256 Mb de memória RAM e 80 Gb de *hard disk*. Foi utilizada a distribuição do Linux chamada de Kurumin, com PHP 5, MySQL 4.0.3 e Apache 2.0.2.

A seguir, apresentam-se os resultados obtidos na fase de validação e testes das duas abordagens desenvolvidas.

5.1 O ambiente para a Mineração de Dados

A base de textos foi montada com 580 respostas, cadastradas pelos professores. Esse conjunto de respostas teve como domínio quatro disciplinas na área de Sistemas de Informação, como mostra a Tabela 5.1. Cada uma das respostas cadastradas teve de uma até cinco palavras chave.

Montada a base de textos, foram feitas a seleção, preparação e limpeza dos dados. Em seguida, os dados foram recuperados e submetidos aos algoritmos disponíveis na

Assunto Principal	Quantidade
Programação Orienta a Objeto	85
Linguagem de Programação Java	245
Linguagem de Programação PHP	130
Banco de Dados Relacional	120

Tabela 5.1: Classificação das respostas cadastradas

ferramenta. Após 230 simulações de conjuntos de palavras chave, chegou-se a conclusão que o algoritmo que apresentou melhores resultados foi o `C4.5`, implementado pelo método `J48`. O algoritmo `C4.5` pode ser estudado em QUINLAN (1993).

A árvore de decisão resultante foi armazenada numa tabela de resultados. Na árvore, os atributos, que aparecem nos primeiros nodos, são os atributos de maior ganho de informação.

A utilização da mineração de dados mais o algoritmo implementado na camada de *interface* possibilita encontrar o melhor conjunto de respostas existente na base de textos, caso exista. A parte do código da aplicação foi inserida no TelEduc, mais precisamente no código da janela de bate-papo. O projeto tentou minimizar as alterações no ambiente TelEduc, visando facilitar a utilização dessa ferramenta pelo seu grupo de usuários.

O aluno passa a ter a sua disposição um campo para digitar sua dúvida, na janela de bate-papo do TelEduc (Fig. 5.1) e clica em *Fazer Pergunta à Base de Conhecimento* para submetê-la.

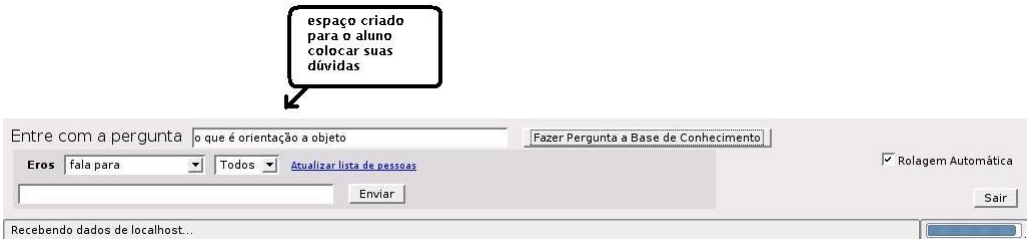

Figura 5.1: Página do bate-papo do TelEduc

As respostas resultantes são visualizadas na tela, como mostra a Figura 5.2.

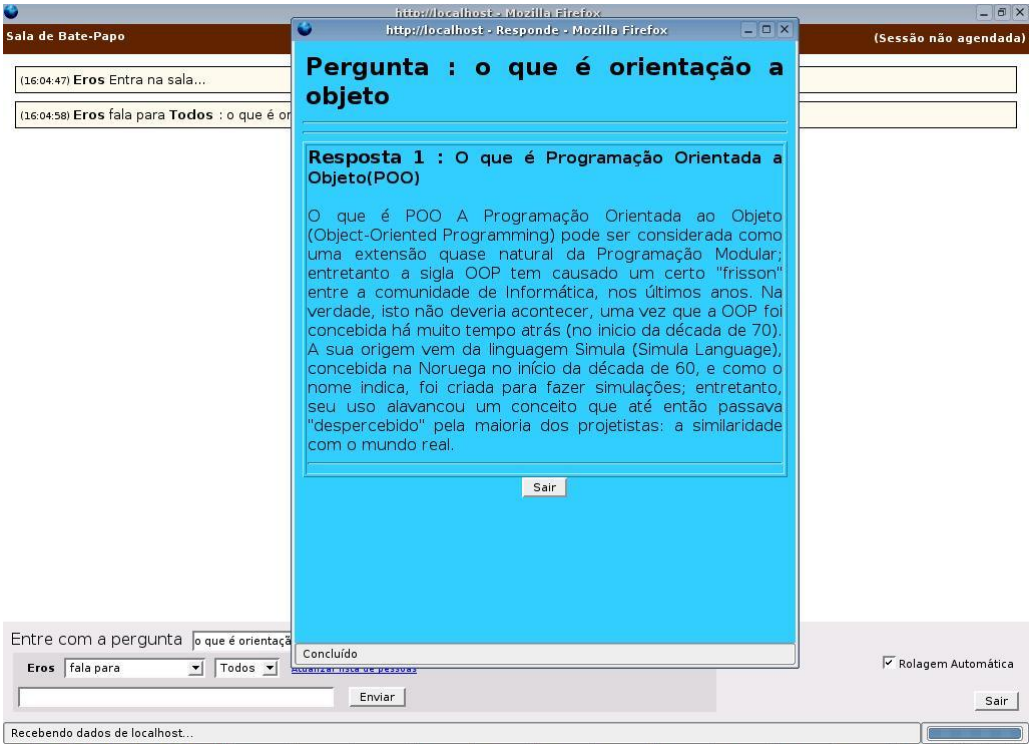

Figura 5.2: Resposta à dúvida submetida pelo aluno

5.1.1 Resultados com Mineração de Dados

Nos testes realizados, foi utilizado um conjunto de perguntas que podem ser classificadas por assuntos mostrados na Tabela 5.2.

	Assunto Principal	**Quantidade**
1	Programação Orienta a Objeto	45
2	Linguagem de Programação Java	35
3	Linguagem de Programação PHP	20
4	Banco de Dados Relacional	15

Tabela 5.2: Classificação das perguntas realizadas

Como estudo de caso foi feito um teste com 115 (cento e quinze) perguntas. Dentre elas: *o que é um objeto* (associada ao tópico 1) , *o que é integridade de dados* (associada ao tópico 4), *o que é método* (associada ao tópico 2), dentre outras.

O objetivo era simular um ambiente real com perguntas, das mais variadas formas. A resposta foi a esperada em 75% das perguntas. Os 25% não satisfatórios foram atribuídos à qualidade da base de textos, pois se verificou que para o correto funcionamento do algoritmo de classificação na mineração de dados há a necessidade da

correta disposição das palavras chave, de tal forma que a palavra mais abrangente venha sempre antes de uma mais específica. Isto nem sempre é viável, pois é preciso contar com a subjetividade do professor na sua escolha e avaliação das palavras chave.

5.2 O ambiente para a Mineração de Texto

A base de textos foi montada para as áreas de Ciência da Computação. A Tabela 5.3 descreve seu conteúdo.

	Assunto Principal	**Quantidade**
1	Programação Orientada a Objeto	45
2	Linguagem de Programação Java	35

Tabela 5.3: Classificação dos textos por áreas de conhecimento

5.2.1 Resultados com Mineração de Texto

Os alunos dos cursos de Ciência da Computação e Sistemas de Informação da Universidade Cândido Mendes (campus Campos dos Goytacazes) e do Centro Universitário São Camilo (ES) fizeram os testes da ferramenta. O objetivo foi simular um ambiente real com perguntas, das mais variadas formas.

Além da pergunta, o aluno podia especificar o grau de satisfação com a resposta retornada pela ferramenta, classificado em bom, $regular$ e $ruim$. Uma tabela foi criada para armazenar as perguntas feitas pelos alunos e o grau de satisfação. Isto permitiu descartar aquelas perguntas cujo conteúdo não encontrava-se incluído na base de conhecimento.

Os testes chegaram a completar 150 (cento e cinqüenta) perguntas. Dentre elas: *o que é objeto* (associada ao tópico 1), *como declarar uma classe em Java* (associada ao tópico 2), dentre outras.

A resposta foi satisfatória em 85,33% das perguntas, sendo que este cálculo exclui as perguntas descartadas. Em 3,33% das perguntas efetuadas obteve grau de satisfação regular e 11,34% obteve avaliação ruim. Os casos não satisfatórios pode-

riam ser atribuídos à qualidade da base de textos, pois verificou-se que alguns textos cadastrados na base eram muito abrangentes e muito longos.

O tempo de resposta às perguntas foi considerado satisfatório (em média dois segundos).

5.3 Discussão dos Resultados

No processo de mineração de texto, podem-se utilizar algoritmos de mineração de dados, sempre que a aplicação da técnica escolhida traga algum benefício comprovado na obtenção das respostas. Vários ensaios, principalmente utilizando algoritmos de classificação, e em especial, de árvores de decisão (`ID3` e `C4.5`) foram testados. Os resultados, após a aplicação das técnicas de classificação, não melhoraram. Adicionalmente, a aplicação das técnicas de mineração de dados exigiriam maior tempo de processamento e configuração de máquina e *software* mais robustos.

Os resultados obtidos consideram-se, em ambas as versões da ferramenta Text-EaD, satisfatórios para o escopo desta dissertação, que pretende demonstrar a importância de se ter ferramentas de apoio aos alunos e às atividades do professor nos AVAs.

Capítulo 6

Conclusão

Em um ambiente virtual de aprendizagem, o professor tem a responsabilidade de gerenciar o conteúdo instrucional e acompanhar a aprendizagem dos alunos. Criar ferramentas inteligentes que facilitem e auxiliem o professor no seu papel dentro do ambiente é muito importante. Uma tentativa interessante foi desenvolvida no escopo deste trabalho para assistir os alunos na busca de respostas a suas dúvidas.

A utilização da mineração de dados mostrou-se eficiente como um mecanismo para auxiliar o algoritmo de recuperação de respostas. Porém, nos testes realizados ficou demonstrado alguns pontos fracos da ferramenta quanto à correta ordenação das palavras-chaves pelo especialista ao inserir na base de dados. Esta vulnerabilidade motivou pesquisas na utilização de outras técnicas para recuperação de informação, em especial, a mineração de texto.

A utilização da mineração de texto mostrou-se eficiente como um mecanismo para auxiliar a pesquisa e recuperação de respostas para os alunos. Em ambos os casos, a eficácia da ferramenta vai depender da subjetividade do professor que administra a base de textos. Os textos devem ser o suficientemente claros e objetivos e relacionados a um determinado tema. O controle e a prevenção de erros devem ser feitos pelo próprio professor, cuja seriedade deve prevalecer na realização desse trabalho.

O objetivo da dissertação foi satisfeito, disponibilizando a ferramenta com tecno-

logias de *software* livre. A ferramenta pode ser acessada no *site* `sourceforge.net` registrado como *Open Source Software Project*.

6.1 Trabalhos Futuros

A elaboração de novos e mais amplos testes, utilizando escopos variados e com um volume maior de textos é um dos pontos a ser considerado em trabalhos futuros. Sugere-se, também, o estudo da possibilidade de recuperar informações sobre documentos em diferentes formatos, como PDF, DOC, RTF e HTML.

Pretende-se, que a ferramenta inclua técnicas de mineração de texto sobre bases históricas não estruturadas que contenham informações sobre os alunos e seu comportamento. Com o uso do computador, tornou-se possível capturar algumas características do aprendiz à distância. A linguagem corporal, o grau de interesse, a participação, o comportamento social, podem ser vistos pela ótica computacional, considerando, basicamente, as interações do aluno com o ambiente de ensino. A freqüência de sua participação em listas de discussão, conferências, fóruns e salas de bate-papo, por exemplo, podem retratar sua sociabilidade.

Referências Bibliográficas

AL., M. N. C. et. Qualificando - ambiente virtual de aprendizagem para ensino de Engenharia de Produção à distância via Internet. **Revista Produção On Line**, [S.l.], v.5, n.2, 2005.

BERRY, M. J. A. .; LINOFF, G. **Data mining techniques**. [S.l.]: John Wiley & Sons, 1997.

CARBONELL, J. R. CAI: an artificial intelligence approach to computer assisted instruction. **IEEE Transactions on Man Machine Systems**, [S.l.], v.11, n.4, p.190–202, 1970.

CARVALHO, L. **Data Mining**. [S.l.]: Ciência Moderna, 2005.

CHAVES, M. Mapeamento e comparação de similaridade entre estruturas ontológicas. **Dissertação (Mestrado em Ciências da Computação) Pontífica Universidade Católica do Rio Grande do Sul, Porto Alegre, 2004**, [S.l.], 2004.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; AL. et. The KDD process for extracting usefulknowledge from volumes of data. **Communications of the ACM. 39: 27 - 34**, [S.l.], novembro 1996.

FRAKES, W.; BAEZA-YATES, R. **Information Retrieval Data Structures and Algorithms**. 1992.

FURTADO, M. **INTELIGÊNCIA COMPETITIVA PARA O ENSINO SUPERIOR PRIVADO**: uma abordagem atravÉs da mineraÇÃo de textos. 2004. Dissertação (Mes-

trado em Ciência da Computação) — UNIVERSIDADE FEDERAL DO RIO DE JANEIRO.

GAVA, T. **Estações de Aprenizagem**: um modelo baseado em ontologias - ufes. 2003. Dissertação (Mestrado em Ciência da Computação) — UFES.

GEY, F. **Models in Information Retrieval**. 1992.

GUEDES, G.; VICCARI, R.; DAMICO, C. Uma ferramenta para auxiliar a avaliação de textos construídos colaborativamente em ambientes de ensino-aprendizagem. **Revista do CCEI (Centro de Ciências da Economia e Informática), URCAMP**, [S.l.], v.6, n.9, 2002.

HAN, J.; KAMBER, M. Data Mining: concepts and techniques. **Morgan Kaufmann Publishers, New York, USA, 2001. 550p.**, [S.l.], 2001.

HARRISON, T. Intranet data warehouse. **Editora Berkeley**, [S.l.], 1998.

JAQUES, P.; OLIVEIRA, F. Agentes de Software para Análise das Interações em um Ambiente de Ensino a Distância. In: III INFOEDUCAR, 1998, Fortaleza, Brasil. **Anais...** [S.l.: s.n.], 1998.

JAQUES, P.; VICCARI, R. PAT: um agente pedagógico animado para interagir afetivamente com o aluno. **Novas Tecnologias na Educação CINTED-UFRGS**, [S.l.], v.3, n.1, 2005.

KEIM, D. A.; ANKERST, M.; AL. et. Recursive Pattern: a technique for visualizing very large amounts of data. **Proceedings of the 6th IEEE Visualization Conference, Atlanta, GA**, [S.l.], 1995.

KINSHUB, A. T.; HONG, H.; PATEL, A. Human Teacher in Intelligent Tutoring System: a forgotten entity. In: IEEE INTERNATIONAL CONFERENCE ON ADVANCED LEARNING TECHNOLOGIES, 2001, Madison, USA. **Proceedings...** [S.l.: s.n.], 2001. p.227–230.

LANDIM, C. **Educação à Distância**: algumas considerações. [S.l.]: Cibercultura, 1999.

LESTA, L.; YACEF, K. An Intelligent Teaching Assistant System for Logic. In: ITS '02: PROCEEDINGS OF THE 6^{th} INTERNATIONAL CONFERENCE ON INTELLIGENT TUTORING SYSTEMS, 2002, London, UK. **Anais...** Springer-Verlag, 2002. p.421–431.

LOH, S.; OLIVEIRA, J.; ALMEIDA, M. Knowledge discovery in texts for constructing decision support systems. **The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies - Special Issue on Text and Web Mining, Journal of Applied Intelligence, Kluwer Academic Publishers, v.18, p.357.**, [S.l.], 2003.

LOH, S.; WIVES, L.; OLIVEIRA, J. Descoberta proativa de conhecimento em coleções textuais: iniciando sem hipóteses. **OFICINA DE INTELIGÊNCIA ARTIFICIAL, 4. 2000. Pelotas. EDUCAT, 2000. 143-154p**, [S.l.], 2000.

MANNING, C.; SHÜTZE, H. Foundations of statistical natural language processing. **Cambridge. The Mit Press, 1999. 620p.**, [S.l.], 1999.

MENEZES, R.; FUKS, H.; GARCIA, A. Utilizando Agentes no Suporte à Avaliação Informal no Ambiente de Instrução Baseada na Web AulaNet. In: X SIMPóSIO DE BRASILEIRO DE INFORMáTICA NA EDUCAçãO, 1999, Fortaleza, Brasil. **Anais...** [S.l.: s.n.], 1999.

NETO, F.; DINIZ, C. Data mining: uma introdução. **São Paulo. Associação Brasileira de Estatística, 2000. 123p.**, [S.l.], 2000.

OEIRAS, J. ACEL - Ambiente Computacional Auxiliar ao Ensino/Aprendizagem a Distância de Línguas. **UNICAMP**, [S.l.], 1998.

PERRONE, J. **EDUCNET**: ambiente interativo para educaÇÃo a distÂncia on-line. 2005. Dissertação (Mestrado em Ciência da Computação) — Fundação Visconde de Cairu - Salvador - Bahia.

PETERS, O. Didática do Ensino a Distância. São Leopoldo,RS UNISINOS. **Revista Brasileira de Aprendizagem Aberta e a Distância**, [S.l.], 2001.

PORTER, M. An algorithm for suffix stripping. In Readings in Information Retrieval, 313-316. **Morgan Kaufmann**, [S.l.], 1997.

QUINLAN, J. **C4.5**: programs for machine learning. 1993.

RAABE, A. L. A.; VARGAS, A. Ampliando a Satisfação dos Usuários Utilizando Assistentes Animados na Interface de Softwares Educacionais. **Revista da Unifebe**, [S.l.], v.4, n.4, p.39–46, 2006.

ROCHA, H. **Projeto TelEduc**: pesquisa e desenvolvimento de tecnologia para educação à distância. 1992.

ROCHAM, H.; OEIRAS, J.; FREIRE, F.; ROMANI, L. Design de ambientes para EaD: re significações do usuário. **Anais do IHC 2001 - IV Workshop sobre Fatores Humanos em Sistemas Computacionais - Florianópolis, SC, pg 84-9**, [S.l.], 2001.

RODRIGUES, S.; OLIVEIRA, J.; SOUZA, J. **Competence mining for virtual scientific community creation**. 2004.

SALTON, G.; MCGILL, M. **Introduction to Modern Information Retrieval**. 1983.

SALTON, G.; SINGHAL, A.; MITRA, M.; BUCKLEY, C. Automatic Text Structuring and Summarization. **Inf. Process. Manage.**, [S.l.], v.33, n.2, p.193–207, 1997.

SANTOS, M. Extraindo regras de associação a partir de textos. **Dissertação (Mestrado em Informática Aplicada) - Pontífica Universidade Católica do Paraná-Curitiba**, [S.l.], 2002.

SARDINHA, T. O banco de palavras-chave. **Programa de Estudos Pós-Graduados em Linguística Aplicada e Estudos da Linguagem Pontifícia Universidade Católica de São Paulo**, [S.l.], 2006.

SCHNEIDERMAN, B. The Eyes Have it: a task by data type taxonomy of information visualizations. **IEEE Symposium on Visual Languages, IEEE Computer Society**, [S.l.], 1996.

SELF, J. The defining characteristics of intelligent tutoring systems research: it's care, precisely. **International Journal of Artificial Intelligence in Education**, [S.l.], v.10, p.350–364, 1999.

SILVA, E. Descoberta de conhecimento com o uso de text mining: cruzando o abismo de moore. 2002. 175f. **Dissertação de Mestrado em Gestão do Conhecimento**

e Tecnologia da Informação, Universidade Católica de Brasília, Brasília., [S.l.], 2002.

SILVA, J. M. C. da; RAABE, A. L. A. Construção de Ferramentas Interativas para Apoio a Aprendizagem via Internet. In: III SEMINáRIO DE INICIAçãO CIENTíFICA, 2004, Itajaí, SC. **Anais...** [S.l.: s.n.], 2004.

VICCARI, R. M.; GIRAFFA, L. M. M. Fundamentos dos Sistemas Tutores Inteligentes. In: BARONE, D. O. (Ed.). **Sociedades artificiais**: a nova fronteira da inteligência das máquinas. [S.l.]: Porto Alegre: Bookman, 2003.

WITTEN, I. H.; FRANK, E. **Data Mining**: practical machine learning tools and techniques. [S.l.]: Morgan Kaufmann, 2005.

WIVES, L. Um estudo sobre agrupamento de documentos textuais em processamento de informações não estruturadas usando técnicas de Clustering. **Dissertação (Mestrado em Ciência da Computação). Programa de Pósgraduação em Gestão do conhecimento da Universidade Federal do Rio Grande do Sul, Porto Alegre, 1999, 84p.**, [S.l.], 1999.

WIVES, L. **Um Estudo sobre Agrupamentos de Documentos Textuais em Processamento de Informações não Estruturados Usando Técnicas de Clustering.** 1999.

YACEF, K. Intelligent teaching assistant systems. In: INTERNATIONAL CONFERENCE ON COMPUTERS IN EDUCATION (ICCE'02), 2002, Auckland, New Zealand. **Proceedings...** IEEE, 2002. v.1, p.136–140.