

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Uma Ferramenta para Auxiliar
na Avaliação de Textos Construídos
Colaborativamente em Ambientes
de Ensino-Aprendizagem**

GILLEANES THORWALD ARAUJO GUEDES

Dissertação submetida à avaliação, como
requisito parcial para a obtenção do grau de
Mestre em Ciência da Computação

Prof^a. Dr.^a Rosa Maria Viccari
Orientadora

Porto Alegre, janeiro de 2002

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Guedes, Gilleanes Thorwald Araujo

Uma Ferramenta para Auxiliar na Avaliação de Textos Construídos Colaborativamente em Ambientes de Ensino-Aprendizagem / por Gilleanes Thorwald Araujo Guedes. — Porto Alegre: PPGC da UFRGS, 2002.

94f. il

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-graduação em Computação, Porto Alegre, BR-RS, 2002. Orientadora: Viccari, Rosa Maria.

1. Educação à distância. 2. KDT. 3. AME-A. 4. Análise de textos construídos colaborativamente. I. Viccari, Rosa Maria. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Prof^ª. Wrana Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitor Adjunto de Pós-Graduação: Prof. Jaime Evaldo Fensterseifer

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

Agradecimentos

Gostaria, neste espaço, de agradecer a todos aqueles que me auxiliaram na realização deste trabalho.

À URCAMP – Universidade da Região da Campanha, por ter me proporcionado a possibilidade de realizar este curso.

À minha orientadora, Professora Doutora Rosa Maria Viccari e à minha co-orientadora, Professora Doutora Carmen D’Amico pela orientação e auxílio.

Ao Professor Doutor José Palazzo de Oliveira, pelas sugestões.

Ao Professor Doutor Cláudio Geyer, pelo auxílio.

À doutoranda Patricia Jaques, pelo grande auxílio prestado.

Aos doutorandos, Leandro Krug Wives e Stanley Loh e à mestranda Adriana Pereira, pelo auxílio.

À professora Ana dos Santos pelo auxílio.

Ao meu pai, William Guedes, pelo grande auxílio e apoio.

À minha mãe, Beatriz, à minha esposa, Sílvia e ao meu filho, Ulisses, pelo apoio e compreensão.

Sumário

Lista de Figuras	6
Lista de Tabelas	7
Resumo	8
Abstract	9
1 Introdução	10
1.1 Objetivos	11
2 Trabalhos Relacionados	12
3 Agentes.....	31
3.1 Agentes Pedagógicos	32
3.2 Sistemas Tutores Inteligentes	34
4 KDT-Knowledge Discovery in Text – Descoberta de Conhecimento em Texto	40
5 O Ambiente AME-A (Ambiente Multiagente de Ensino-Aprendizagem)	52
6 A Ferramenta Desenvolvida	61
6.1 O Modelo Entidade-Relacionamento.....	62
6.2 Dicionário de Dados.....	63
6.3 O Funcionamento da Ferramenta.....	65
6.4 A Opção Arquivo	67
6.5 A Opção Avaliação	75
6.5.1 Avaliação Padrão	76
6.5.2 Avaliação por Questões (Queries)	81
6.6 A Opção Cadastros	83
6.6.1 Cadastro de Professores	83
6.6.2 Cadastro de Níveis	84
6.6.3 Cadastro de Palavras Ferramentas (Stop Words)	86
6.6.4 Cadastro de Temas	87
6.6.5 Cadastro de Palavras-Chave	87
6.7 A Opção Relatórios.....	88

6.8	A Opção Configurações.....	89
7	Conclusão e Trabalhos Futuros	90
	Bibliografia.....	92

Lista de Figuras

FIGURA 5.1 – Arquitetura do ambiente AME-A	52
FIGURA 6.1 – Tela do módulo de Chat.....	61
FIGURA 6.2 – Interação entre o módulo de Chat e a Ferramenta	61
FIGURA 6.3 – Modelo Entidade-Relacionamento	62
FIGURA 6.4 – Tela de Abertura da Ferramenta	65
FIGURA 6.5 – Tela Principal da Ferramenta.....	66
FIGURA 6.6 – Alternativas da Opção Arquivo	67
FIGURA 6.7 – Tela de Seleção de Arquivo.....	68
FIGURA 6.8 – Tela de Identificação dos Aprendizes.....	69
FIGURA 6.9 – Identificação de Aprendizes/Interações.....	70
FIGURA 6.10 – Classificação de Palavras.....	71
FIGURA 6.11 – Classificação de Palavras.....	72
FIGURA 6.12 – Totalizadores do Módulo de Classificação.....	73
FIGURA 6.13 – Alteração da Classificação de uma Palavra ou Frase	73
FIGURA 6.14 – Alteração de uma Palavra ou Frase Desconhecida para Chave.....	74
FIGURA 6.15 – Classificação Dividida por Interação.....	74
FIGURA 6.16 – Gráfico de Palavras/Frases-Chave Mais Utilizadas.....	75
FIGURA 6.17 – Alternativas da Opção Avaliação	76
FIGURA 6.18 – Tela de Avaliação Padrão	77
FIGURA 6.19 – Avaliação Padrão de um Arquivo de Chat	78
FIGURA 6.20 – Utilização das Palavras/Frases-Chave	79
FIGURA 6.21 – Palavras Desconhecidas Mais Utilizadas.....	80
FIGURA 6.22 – Gráfico Comparativo entre o Total de Interações e o Total de Interações Consideradas Válidas de Cada Aluno.....	81
FIGURA 6.23 – Módulo de Avaliação por Questões (Queries).....	82
FIGURA 6.24 – Avaliação por meio de uma Questão (Query)	83
FIGURA 6.25 – Cadastro de professores	84
FIGURA 6.26 – Cadastro de Níveis.....	85
FIGURA 6.27 – Cadastro de Palavras Ferramentas (Stop words)	86
FIGURA 6.28 – Cadastro de Temas.....	87
FIGURA 6.29 – Cadastro de Palavras-Chave	88
FIGURA 6.30 – Módulo de Configurações.....	89

Lista de Tabelas

TABELA 1 – Tabela de Professores.....	63
TABELA 2 – Tabela de Temas Coordenados por Professor.....	63
TABELA 3 – Tabela de Temas.....	63
TABELA 4 – Tabela de Palavras-Chave	63
TABELA 5 – Tabela de Níveis	64
TABELA 6 – Tabela de Stop Words	64
TABELA 7 – Tabela de Aprendizes	64
TABELA 8 – Tabela de Interações.....	64
TABELA 9 – Tabela de Configurações.....	65

Resumo

Este trabalho está relacionado ao Ambiente Multiagente de Ensino-Aprendizagem (AME-A), no qual os agentes que o compõem preocupam-se em ensinar e/ou aprender. A ferramenta descrita neste trabalho baseia-se, em parte, na idéia do agente Promove_Interação, que 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 frequência com que este utiliza as palavras-chave através de dois métodos de avaliação, baseados em algumas técnicas de KDT. O software permite também a classificação de todas as palavras/frases empregadas durante a reunião.

Palavras-Chave: Educação à Distância, KDT, AME-A, Análise de Textos Construídos Colaborativamente.

TITLE: “A TOOL TO HELP IN THE EVALUATION OF COLABORATIVELY BUILT TEXTS IN TEACHING-LEARNING ENVIRONMENTS”

Abstract

This work is related to the learning-teaching multiagent environment, AME-A, in which the agents that compose it are concerned with teaching and/or learning. The tool described in this work is partly based upon the idea of the agent, Promove_Interação, which aims to make possible for several students and teachers to communicate to each other through the Internet and to discuss issues as chosen by the teacher.

Trying to help the teacher’s task to find out whether the students are actually acquiring knowledge, a tool was developed to analyze students’ interactions. The developed algorithm employs a key-sentence/key-word dictionary related to the chosen subject which refers to issues that should be discussed and/or take part in students’ conclusions. When activated, the software identifies the learners and their respective interactions and stores these into a database; following this, it evaluates each learner’s interactions, assessing the frequency with which they use the key-words by means of two evaluation methods, based on some KDT techniques. The software also allows the classification of all the words/sentences employed during the meeting.

Keywords: Long-distance Learning, KDT, AME-A, Collaboratively-Built Text Analysis.

1 Introdução

Nas últimas décadas, pesquisadores e educadores têm procurado melhorar a maneira de ensinar, por meio da utilização de diversas metodologias diferentes e verificar se tais formas de ensino causam melhorias na aprendizagem dos alunos.

Segundo Piaget, citado em [LIM 84], o que faz o ensino ser inteligente não é o conteúdo, mas a maneira de ensinar. Para Piaget, a inteligência é compreender e inventar. Como a teoria piagetiana é baseada em um construtivismo, conclui-se que a teoria de Piaget é a própria teoria de criatividade, pois sem criatividade não há construção. Todo o ensino que se baseia na imitação do professor, isto é, que depende da aprendizagem de fórmulas, definições e nomenclaturas, não é ensino inteligente. Uma didática piagetiana consiste em descobrir as técnicas de ensinar através do ensaio e do erro, da pesquisa e da solução de problemas novos.

A verdadeira forma de aprendizagem moderna é a estimulação para que o aprendiz, antes de qualquer insinuação do professor, invente maneiras de resolver o problema. Isso é conhecido como educação pela inteligência e consiste em propor problemas aos aprendizes e jamais ensinar as soluções [LIM 84].

Este trabalho está baseado na proposta de D'Amico em [DAM 99] onde é sugerido um ambiente multiagente de ensino-aprendizagem, no qual os agentes que o compõem preocupam-se em ensinar e/ou aprender. A ferramenta descrita neste trabalho baseia-se, em parte, na idéia do agente Promove_Interação, que tem por objetivo possibilitar a diversos aprendizes e professores se comunicarem, através da Internet, e discutirem assuntos determinados por um professor, construindo textos de maneira colaborativa. Os aprendizes se reuniam nas datas estabelecidas e tentariam chegar a conclusões por meio de discussões sobre os assuntos em questão, coordenados por um professor. O agente Promove_Interação é representado neste trabalho por um software de “chat”, que, embora não implemente ainda muitas das características atribuídas ao agente, permite a reunião virtual de diversos aprendizes e a gravação do conteúdo produzido durante a discussão.

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 contidas nos textos por eles construídos durante suas reuniões. Esta ferramenta será também de grande utilidade para o agente Analisa_Aprendizagem, descrito em [DAM 99]. O algoritmo desta ferramenta baseia-se em um dicionário de palavras/frases-chave 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 aqui descrito 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 frequência com que este utiliza as palavras/frases-chave através de dois métodos de avaliação, baseados em algumas técnicas de KDT (Knowledge Discovery in Text – Descoberta de Conhecimento em Texto); além de permitir também a classificação de todas

as palavras/frases empregadas durante a reunião. A ferramenta procura, dessa forma, auxiliar o professor a determinar se seus aprendizes estão ou não adquirindo conhecimento e se não estão desperdiçando tempo em trivialidades.

Pretende-se, com o desenvolvimento desta ferramenta, associada ao software de “chat” acima descrito, enriquecer o ambiente AME-A, fornecendo-lhe, mesmo que muitas das características do agente Promove_Interação não tenham ainda sido implementadas no software de chat, um ambiente em que os diversos aprendizes que utilizam o ambiente AME-A possam virtualmente se reunir, trocar idéias através da discussão de determinados assuntos e produzir textos de maneira colaborativa. Além disso, a ferramenta descrita neste trabalho fornece uma maneira de auxiliar o professor a determinar o avanço de seus aprendizes na aquisição de conhecimentos por meio das discussões, o que permite ao professor verificar os termos mais freqüentemente utilizados, bem como determinar se os aprendizes estão discutindo todos os tópicos propostos e não se estão desviando do assunto em questão.

Este trabalho primeiramente descreverá alguns trabalhos semelhantes, dedicados à análise de textos construídos colaborativamente; em seguida, apresentará uma revisão de literatura a respeito de agentes e seu emprego em ambientes de ensino-aprendizagem, quando passam a chamar-se agentes pedagógicos; passando para uma revisão sobre técnicas de KDT (Knowledge Discovery in Text). Após isso, é descrito o sistema AME-A (Ambiente Multiagente de Ensino-Aprendizagem), no qual este trabalho está baseado; e, finalmente, a ferramenta desenvolvida.

1.1 Objetivos

Desenvolver uma ferramenta que permita auxiliar professores a determinar o rendimento de seus aprendizes durante a construção de textos de forma colaborativa através de discussões virtuais por meio de “chats”.

2 Trabalhos Relacionados

Em seguida serão descritos, brevemente, alguns trabalhos relacionados ao tema descrito nesta dissertação.

Segundo Jaques em [JAQ 99b], tem ocorrido, na área de educação à distância, uma ênfase crescente em aprendizado colaborativo, de maneira a estimular grupos de aprendizes a interagir e trocar idéias, auxiliando-se mutuamente. O uso de um ambiente colaborativo, no entanto, exige um grande esforço do professor para monitorar toda a discussão entre os alunos, para evitar que estes percam tempo com assuntos outros que não o do tema escolhido pelo professor. Maçada e Tijiboy, citados em [JAQ 99b], afirmam que, atualmente, muitas pesquisas na área de ensino à distância, procuram desenvolver ferramentas e metodologias que sejam apropriadas ao processo de ensino-aprendizagem em ambientes de grupos colaborativos.

Jaques em [JAQ 99a] desenvolveu um sistema multiagente para monitorar a interatividade entre aprendizes em ambientes telemáticos de ensino. O sistema é composto por quatro agentes, sendo que três deles, chamados coletores, monitoram uma determinada ferramenta (e-mail, news ou chat) e um quarto, chamado agente do professor, disponibiliza os dados analisados ao professor, para auxiliá-lo a determinar se os alunos não estão fugindo de seus objetivos.

Este sistema tenta identificar três tipos básicos de associações nas interações de grupos de aprendizes:

- Aluno-aluno: Identificação de quais alunos mais interagem entre si.
- Aluno-assunto: Identificação de quais assuntos cada aprendiz mais discute; procura identificar tópicos de maior interesse.
- Aluno-aluno-assunto: Identificação de quais assuntos mais interessam a um grupo de aprendizes.

O sistema de Jaques também realiza análises estatísticas, tais como número de mensagens trocadas e percentual de participação, com o objetivo de auxiliar o professor a determinar a participação e a colaboração dos aprendizes.

O sistema baseia-se em um dicionário léxico-morfológico para identificar os substantivos e verbos mais utilizados nos textos, de forma a determinar quais os assuntos mais discutidos. O sistema de Jaques segue os seguintes passos:

- Identificação dos substantivos e verbos nas mensagens;
- Seleção das palavras com maior número de ocorrências;

- Verificação da ocorrência de sinônimos;
- Uso de um thesaurus para verificar palavras que sejam sub-assuntos de outras, por meio da identificação de relações de hierarquia entre as palavras e procura de sinônimos; tanto as relações como os sinônimos são definidos pelo professor.

Barros em [BAR 2000] propôs uma abordagem para caracterizar comportamento grupal e individual em trabalho colaborativo suportado por computador onde é apresentada uma descrição qualitativa orientada a um processo de uma atividade mediada de grupo sob três perspectivas:

- Um desempenho de grupo em referência a outros grupos.
- Cada membro em referência a outros membros do grupo.
- O grupo por si mesmo.

Na abordagem de Barros, a colaboração é baseada em conversação. Barros propôs um método para calcular automaticamente estes atributos para processos em que atividade e interações conjuntas são realizadas por meio de mensagens semi-estruturadas.

O sistema de Barros, chamado DEGREE (Distance Environment for Group Experiences – Ambiente à Distância para Experiências de Grupo), está baseado nos seguintes princípios:

- Construção conjunta de uma resolução de um problema.
- Coordenação de membros do grupo para planejamento de tarefas.
- Semi-estruturação dos mecanismos de interações.
- Focalização, tanto no processo de aprendizagem, como no resultado da aprendizagem, representando explicitamente os processos de produção e interação.

O sistema de Barros se divide em 4 níveis:

- Nível de Configuração, onde os professores especificam tarefas, recursos e grupos.
- Nível de Desempenho, onde um grupo de alunos pode realizar atividades colaborativas com suporte do sistema. O sistema gerencia as intervenções dos usuários, chamadas contribuições, suportando a co-construção de uma solução em um processo de discussão argumentativo e colaborativo. Todas as colaborações são gravadas.

- Nível de Análise, onde são analisadas as interações dos usuários e feitas intervenções de forma a melhorá-las. Este nível fornece ferramentas para análise quantitativa e qualitativa das interações desenvolvidas pelos aprendizes. As atitudes dos alunos são observadas e valorizadas quando estes trabalham em grupo. O sistema envia mensagens com sugestões para melhorar as intervenções dos alunos.
- Nível de Organização, onde os resultados dos processos e experiências de aprendizado colaborativo, são reunidos, selecionados e armazenados.

Na abordagem de Barros, os aprendizes têm de escolher um tipo específico de interação, chamado de Contribuições. Estas contribuições recebem 4 tipos de valores referentes à iniciativa, criatividade, elaboração e conformidade. Os tipos de contribuição possíveis são Proposta, Contra-proposta, Questão, Comentário, Esclarecimento e Concordância.

O sistema de Barros grava todos os acessos e ações realizadas pelos usuários quando eles estão resolvendo as tarefas. A informação registrada automaticamente inclui a identificação do usuário, a data, a hora, o computador host, a experiência de aprendizado, o grupo, a atividade, a tarefa, o ambiente de trabalho e o tipo de ação, o que permite criar uma variedade de “queries” combinando uma seleção de critérios. O sistema possui três módulos de análise:

- Análise Quantitativa, onde são realizadas queries contendo diversos parâmetros, na qual a saída pode ser tanto textual como gráfica.
- Análise Qualitativa, considera a configuração da experiência, o processo e o resultado, mais o conhecimento subjetivo do observador sobre a interpretação dos valores observados. A análise é realizada em três dimensões: comparação global de um grupo com outros grupos, comparação de cada aluno com seus colegas e uma análise de tarefa de grupo. O sistema também envia mensagens sobre os resultados da análise, destacando o que pode ser melhorado na atividade do grupo.
- Análise de Retorno, que utiliza a saída do sistema coordenador e representa graficamente a evolução do grupo para um subconjunto selecionado das variáveis observadas.

Na análise quantitativa, uma interface baseada na web é fornecida pela seleção de opções diferentes e combinadas para uma query. A interação do usuário pode ser observada de diferentes formas:

- Evolução do número de contribuições do usuário em uma experiência durante um período.
- Contagem de um número de acessos horários para um grupo em atividade.

- Número de contribuições por usuário e por todos os membros do grupo em atividade.
- Número de contribuições por usuário e por tipo para cada sub-tarefa de uma tarefa.
- Contribuições de um grupo relacionadas a uma tarefa, pelo tipo de contribuição, mostradas como um gráfico em barra.
- Número de contribuições por membro para uma atividade
- Evolução de uma discussão para uma sub-tarefa, graficamente representada.

Na Análise Qualitativa, os dados para a análise de processo de uma experiência de aprendizagem vêm de duas fontes:

- A configuração da experiência: tarefa, estrutura da conversação, funções e grupos.
- As contribuições dos aprendizes, organizadas pelo sistema como estruturas em árvore para cada espaço de trabalho de tarefa e o conjunto de mensagens intercambiadas através do espaço de coordenação associado a cada atividade.

Os atributos para realizar a comparação do comportamento do grupo são calculados a partir dos dados sobre a definição da tarefa e o processo realizado ou concluído a partir de “inferência fuzzy”, usando um conjunto de regras de avaliação. A partir destes dados são considerados e computados os seguintes atributos:

- Para cada elaboração de espaço de trabalho:
 - O número médio de contribuições, o número de contribuições do grupo dividido pelo número de membros do grupo.
 - O tamanho médio das contribuições, o tamanho médio do conteúdo da contribuição, em caracteres.
 - A profundidade da árvore da contribuição, profundidade máxima das árvores relacionadas ao espaço de trabalho.
 - Interatividade, percentual de contribuições respondidas ou ligadas a outras contribuições feitas por um aluno que não o contribuidor.
- Para cada espaço de coordenação, onde são produzidos comentários alheios à tarefa, são considerados o número total de mensagens capturadas pelas variáveis “coordination messages”.

O método da análise individual é o mesmo da de grupo, mas são considerados um conjunto diferente de atributos e regras, alguns computados a partir dos dados da experiência para cada aluno no grupo levando em conta o valor médio do grupo. Relacionados a cada aluno, são considerados os seguintes atributos: o número de contribuições que ele adicionou, o tamanho médio de suas contribuições, a média de iniciativa, a criatividade, elaboração e conformidade de suas contribuições. Os novos atributos para a análise individual são:

- Número de contribuições deste aluno que foram respondidas.
- Número de contribuições criadas por outros que foram respondidas por este aluno.
- Número de contribuições criadas pelo aluno que foram continuadas ou respondidas por ele mesmo.
- Número de contribuições no estágio de proposta que foram elaboradas pelo aluno.

Todos estes atributos são combinados para inferir novos atributos com o objetivo de obter uma avaliação individual.

A análise baseada no resumo de comportamento de tarefa de grupo dedica-se aos aspectos de distribuição do trabalho entre os componentes do grupo e da evolução da atividade do grupo em um período de tempo; os dados são utilizados como nas análises anteriores; todavia, considera-se que, durante uma discussão argumentativa para resolução de uma tarefa de grupo existem diferentes estágios. Foram identificados três estágios principais:

- Proposta, contribuições que envolvem o fornecimento relevante de informação com idéias que possam estar no resultado final da tarefa.
- Argumentação, contribuições que envolvem a discussão de idéias, seu refinamento ou esclarecimento de coisas que possam ajudar a avançar a discussão do grupo.
- Concordância, contribuições que manifestam concordância implícita com as idéias de colegas. Estas contribuições são usadas para finalizar a discussão.

Estes estágios podem ser repetidos e interligados em um processo colaborativo. Um tipo de contribuição só pode pertencer a um estágio.

A análise de retorno (feedback), tem como objetivo melhorar as interações dos alunos fornecendo retornos diretamente aos estudantes e procura avaliar se os retornos afetam positivamente o desempenho dos alunos. Isto engloba a monitoração do

comportamento dos aprendizes antes e depois que o retorno é fornecido, a capacidade de sugerir correlações possíveis onde as mudanças são observadas e a capacidade de verificar se o retorno foi apropriado ou não. Esta última tarefa é principalmente complexa e provavelmente deve ser feita pelo tutor.

Flores, citado em [SOL 2001], com o objetivo de estudar as interações organizacionais, utilizou uma abordagem baseada em menus de atos de conversação como Request, Promise e Accept. Esta abordagem foi considerada coerciva demais em suas primeiras versões. McManus e Aiken, citados em [SOL 2001], desenvolveram a Rede de Habilidades Colaborativas, uma taxonomia de atos de conversação, onde a cada ato de conversação foi atribuído um introdutor de frase para indicar a intenção do ato ao sistema, os introdutores de frase costumam ser do tipo “Você acha” ou “Eu discordo, porque”. Este sistema estabeleceu uma ordem definindo quais atos deveriam vir antes e depois de outros atos. Baker e Lund, citados em [SOL 2001] compararam o comportamento de alunos na resolução de problemas ao utilizar uma interface com introdutores de sentença e uma interface de chat não estruturada. A interface com introdutores de sentença obteve melhores resultados.

Soller em [SOL 2001], dando continuidade nos estudos descritos acima, tentou distinguir as interações que foram efetivamente produzidas por interação de grupo daquelas que não foram. Soller desenvolveu um Modelo de Aprendizado Colaborativo (Collaborative Learning Model ou CL Model), para prover suporte imediato para as mudanças de necessidades sociais e pedagógicas dos estudantes, procurando auxiliar um sistema de aprendizado colaborativo inteligente a identificar e solucionar áreas de problema de interações de grupo.

Soller classificou as características que devem ser exibidas durante as interações de aprendizado colaborativo efetivo em cinco categorias: participação, habilidades de conversação de aprendizado ativo, análise de desempenho e processamento de grupo e interação promocional.. O Modelo de Aprendizado Colaborativo de Soller propõe estratégias para cada indicação de aprendizado colaborativo efetivo, para melhorar as interações de grupo, além de detectar problemas e auxiliar os aprendizes. O modelo de Soller também fornece aos desenvolvedores do Sistema de Aprendizado Colaborativo Inteligente um framework e um conjunto de recomendações para ajudar grupos a adquirirem habilidades de aprendizado colaborativo efetivo.

Soller afirma que analisar as interações de alunos não é uma tarefa trivial, uma vez que as tecnologias de compreensão de linguagem natural combinadas com ferramentas de aprendizado colaborativo suportado por computador ainda são muito limitadas para compreender e interpretar as interações dos estudantes. Assim, Soller optou por usar uma interface de comunicação estruturada que exige que os aprendizes utilizem introdutores de sentença, para poder identificar a intenção do estudante, sem compreender totalmente o significado de sua interação.

Eleuterio em [ELE 01] descreve AMANDA (Agent de Modélisation et ANalyse de Dialogues Argumentés), um sistema que tem por objetivo auxiliar tutores a melhorar os resultados das discussões de grupos e aumentar a transferência de conhecimento entre os

participantes. Isto é feito integrando o diálogo coletivo como uma atividade bem coordenada e disciplinada em situações de aprendizado à distância. O diálogo é representado como uma árvore de argumentação, ou seja, um conjunto estruturado de questões, escolhas e argumentos que se desenvolvem no decorrer de ciclos de diálogo sequenciais. O comportamento inteligente do sistema é devido às ações de coordenação tomadas em resposta ao raciocínio sobre o diálogo.

Eleuterio em [ELE 01], afirma que os fóruns de discussão tradicionais falham muitas vezes em promover aprendizado de grupo, por duas razões principais:

- Falta de disciplina devido à fraca integração do processo de discussão nas atividades regulares do curso.
- Falta de articulação e coordenação da discussão.

Com o propósito de superar os problemas descritos acima, Eleuterio propôs um framework de diálogo que automaticamente coordena o diálogo e utiliza os participantes para gerar atividades de diálogo. Eleuterio identificou três diferenças principais entre AMANDA e os fóruns de discussão tradicionais:

- A presença de modelos de domínio na arquitetura de AMANDA permite um certo grau de raciocínio semântico sobre o diálogo.
- O mecanismo de coordenação alivia o tutor de tarefas de coordenação consumidoras de tempo, como encontrar relações entre as contribuições dos usuários, medindo o grau de comportamento dos participantes, detectando tópicos de desacordo e medindo a cobertura dos tópicos da discussão.
- O sistema gerencia o diálogo por meio da geração de ciclos de discussão, nos quais os participantes expressam suas idéias de apoio e oposição em relação à contribuição de outro participante, criando um contexto adequado para a articulação e confrontação de idéias e pontos de vista.

O mecanismo de coordenação proposto por Eleuterio permite vários graus de representação de conhecimento sem prejudicar o controle de diálogo, ou seja, se o sistema não possuir modelos de conhecimento, ele ainda pode coordenar o diálogo mesmo que com alguma degradação, considerando somente parâmetros estruturais. Isto é possível devido à separação entre os aspectos estruturais e semânticos, o que permite utilizar AMANDA em situações onde a modelagem do conhecimento não é possível, como discussões de domínio aberto.

Eleuterio almeja combinar duas abordagens no sistema AMANDA:

- A abordagem utilizada na arquitetura CoLLeGE descrita por Ravenscroft e Pilkington, citados em [ELE 01], que analisa movimentos de diálogo, mudanças conceituais e modelos do mundo como a base do processo de diálogo. Esta

arquitetura inspeciona profundamente a interação entre o tutor e o aprendiz, mas dá pouca ênfase ao aspecto coletivo do diálogo.

- A abordagem utilizada no Ambiente de Discurso Argumentativo, descrito por Karacapilidis citado em [ELE 01], a qual é totalmente devotada a formalizar discursos argumentativos.

AMANDA é um sistema autônomo de coordenação de diálogo inteligente independente de domínio aplicado a discussões coletivas. O Módulo de Controle de Diálogo é o mecanismo de coordenação central de AMANDA. Este módulo organiza os diálogos em períodos sequenciais chamados sessões, representando um intervalo de tempo no qual um certo número de discussões serão realizadas. Durante cada sessão o sistema dispara um certo número de ciclos de diálogo de forma a atualizar a árvore de diálogo com contribuições dos participantes.

O planejamento do diálogo é representado pelo sincronizador de diálogo (dialog schedule) e o sincronizador de sessão (session schedule). O sincronizador de diálogo é o planejamento geral do diálogo, ele especifica as sessões de diálogo, as datas correspondentes a início e fim e o respectivo domínio de discurso. O sincronizador de sessão é uma estrutura dinâmica produzida automaticamente e atualizada pelo sistema durante uma determinada sessão. Cada lançamento do sincronizador de sessão é um ciclo de diálogo que especifica uma tarefa de diálogo para cada participante. Uma tarefa de diálogo é o conjunto de todos os nós da árvore de diálogo que são atribuídos ao mesmo participante em um certo ciclo de diálogo.

Uma tarefa de diálogo é representada por uma “tarefa de folha de trabalho” (worksheet assignment), na qual uma lista de elementos de folhas de trabalho é atribuída a um participante particular.

A árvore de diálogo é a estrutura que representa o diálogo. Seus nós internos podem ser de cinco tipos:

- **Nó de Diálogo:** É o nó mais importante da árvore. Ele contém uma referência para um determinado número de sessões de diálogo. Quando um diálogo é criado, este nó é inicializado com a informação contida no sincronizador de diálogo.
- **Nó de Sessão:** É o nó mais importante de uma sessão de diálogo. As sessões de diálogo têm por objetivo organizar a discussão em períodos de tempo separados, cada uma atribuída a um certo domínio do discurso. O nó de Sessão contém uma referência a todos os elementos de discussão, que estão sincronizados para discussão dentro desta sessão.
- **Nó DE:** Este nó representa um elemento de discussão (Discussion Element – DE), uma questão em linguagem natural que irá originar uma discussão específica, como por exemplo, “Que tipos de elementos de conexão existem em

uma rede de computadores ?”. Um elemento de discussão pode ser classificado como um ato de fala interrogativo à espera de conteúdo (content-expected interrogative speech act), segundo Porayska-Pompa citado em [ELE 01], para o qual se espera uma resposta com um certo conteúdo. De acordo com o modelo de argumentação de Karacapilidis, citado em [ELE 01], um nó DE é uma questão a ser debatida.

- Nó ALT: O nó ALT (alternativa), é uma resposta a uma questão. É uma resposta alternativa a um certo elemento de discussão. A resposta contida em um nó ALT é o conteúdo esperado pelo seu nó DE correspondente. No modelo de Karacapilidis, citado em [ELE 01], um nó ALT é uma posição sobre uma questão.
- Nó ARG: O nó de Argumentação representa uma reação de apoio ou oposição de um determinado participante sobre um elemento de diálogo colocado por outro participante. Um nó ARG pode referir-se a um nó ALT ou a outro nó ARG. Os nós de argumentação são elementos chave do diálogo; quando analisados como um todo, representam o nível de concordância coletiva sobre uma determinada posição. Cada nó ARG conduz uma intenção de apoio ou oposição. Esta intenção é expressada por quatro níveis: concordância total (++), concordância parcial (+), discordância parcial (-) e discordância total (--). Um grande esforço de coordenação de AMANDA está concentrado na análise dos efeitos dos nós ARG sobre a árvore de diálogo.

O Módulo de Controle de Diálogo possui uma interface gráfica que permite observar a árvore de diálogo e realizar funções de edição e anotação sobre o diálogo.

O Módulo KB é responsável pelo gerenciamento do modelo de conhecimento e pelo fornecimento de parâmetros semânticos para o Módulo de Controle de Diálogo. A representação de conhecimento central é a ontologia de domínio; no entanto, podem ser acrescentadas outras estruturas, como a estrutura de tarefa de domínio. O módulo KB avalia o diálogo do ponto de vista semântico, calculando um certo número de parâmetros, tais como proximidade semântica entre mensagens baseadas em texto, a distância conceitual entre conceitos de ontologia ou a cobertura conceitual de uma determinada sessão de diálogo.

O sistema AMANDA tem necessidade de modelos de domínio para produzir raciocínio semântico sobre o diálogo. De maneira a permitir tipos diferentes de modelos de domínio a serem conectados no módulo KB, é utilizada uma abordagem centralizada em ontologias, o que permite construir diversos modelos, tais como mapas conceituais e estruturas de tarefa.

A ontologia de domínio é a representação de conhecimento central de AMANDA. Seu papel é organizar conceitos de domínio de forma que permitam raciocínio. Em AMANDA adotou-se uma estrutura simples que organiza conceitos por meio de ligações é-um e parte-de.

A estrutura de tarefa é utilizada como um modelo de conhecimento complementar. Ela representa a decomposição de uma tarefa por meio de dois tipos de ligações:

- Ligações de Seqüência, que decompõem uma tarefa complexa em uma seqüência de sub-tarefas seqüenciais mais detalhadas.
- Ligações de Tipo, que especificam métodos diferentes de desempenho de uma certa tarefa.

O Módulo Gerador de Elementos de Diálogo produz questões em linguagem natural baseadas nos modelos de conhecimento disponíveis. As questões são geradas pelo sistema de maneira a incluir um tópico de domínio específico para o diálogo.

O Módulo HTML é responsável pela interface entre AMANDA e os participantes do diálogo. Este módulo gera dinamicamente folhas de trabalho no formato HTML, que são acessados pelos participantes, preenchidas e enviadas de volta para o sistema, onde o módulo de controle de diálogo atualiza a árvore de diálogo.

O diálogo inicia com a criação de um sincronizador de sessão baseado no sincronizador de diálogo disponível. Quando a sessão de diálogo for estabelecida, o sistema pode disparar o primeiro ciclo de diálogo, identificado como o nível ALT na árvore de diálogo que almeja distribuir os elementos de discurso entre os participantes. O sistema AMANDA toma o conjunto de elementos de discurso e o conjunto de participantes e executa o algoritmo de atribuição de elementos de discurso, que gera atribuições do tipo (elemento de discussão, lista de identidades) e pode ser parametrizada de acordo com a carga desejada de elemento de discussão/participante e a presença/ausência do tutor na discussão.

Como resultado do primeiro ciclo, o sistema recebe um grande número de respostas para os elementos de discurso propostos, chamadas alternativas, representadas pelos nós ALT na árvore de diálogo. Estas alternativas serão o foco da análise no ciclo de argumentação. A partir deste momento, AMANDA gerará uma seqüência de ciclos de diálogo, de maneira a expandir a árvore em profundidade ou em amplitude (largura), até que se atinja um grau satisfatório de concordância. Neste ponto, são distinguidos dois conceitos chave:

- **Nível de Diálogo:** Representa o nível de profundidade da árvore de diálogo, ou seja, a distância de um determinado nó até a nó raiz (o primeiro nó da árvore). A presença de um grande número de níveis de diálogo implica que o diálogo cresceu em profundidade, o que leva a concluir que uma resposta original de um determinado elemento de discussão foi motivo de muitos ciclos de argumentação subsequentes. Níveis de diálogo altos indicam que a resposta foi sendo repetidamente repudiada ou progressivamente esclarecida. Certos comportamentos típicos no discurso argumentativo, tais como mudanças de crença, só podem ser detectados com níveis de diálogo altos.

- **Ciclo de Diálogo:** Representa um período de tempo no qual a árvore de diálogo se expande. A presença de um grande número de ciclos de diálogo implica que o diálogo desenvolveu-se por meio de um grande número de interações, mas não necessariamente que teve um crescimento em profundidade.

As ações de coordenação tomadas pelo sistema são baseadas em um certo grau de raciocínio sobre a árvore de diálogo. Eleuterio propôs dois tipos de diálogo:

- **Raciocínio Estrutural:** Preocupa-se com o aspecto estrutural da árvore de diálogo, especialmente com a distribuição dos nós ARG e suas intenções inerentes de apoio/oposição para decidir quais nós serão re-iniciados e para quais participantes serão atribuídos.
- **Raciocínio Semântico:** Analisa o conteúdo da informação textual de maneira a encontrar relações semânticas entre os nós.

A separação entre raciocínio estrutural e semântico permite que AMANDA coordene o diálogo mesmo na ausência de modelos de domínio.

O principal parâmetro estrutural do raciocínio estrutural é o nível de apoio de nó em relação ao seu nível mais baixo na sub-árvore. Antes de iniciar um novo ciclo de diálogo, AMANDA avalia o nível de concordância geral de cada elemento de discussão e decide sobre a criação de um novo ciclo ou o encerramento da árvore de discussão para o elemento de discussão correspondente. Esta decisão leva em consideração os seguintes níveis de conceito:

- **Nível de Apoio Local:** Cada nó do tipo ALT ou ARG pode ser atribuído a um nível de apoio local. Este nível representa o grau de consenso deste nó com respeito à sua sub-árvore de nível mais baixo. Os níveis de apoio são calculados pelo cruzamento da árvore de diálogo das folhas (nós terminais) até a raiz (nó inicial) e pela atribuição de níveis de suporte para cada nó ALT ou ARG. O nível de apoio local é um número real que varia de -1.0 (discordância total) a $+1.0$ (concordância total). Este número é o nível médio de apoio transmitido por todos os seus nós descendentes diretos. Se o nó é um nó folha (nó terminal), o nível de apoio local recebe o valor máximo de $+1.0$.
- **Nível de Apoio Transmitido:** O princípio é que cada nó ARG descendente transmite a seu nó pai direto um certo nível de apoio. Este nível depende do tipo de argumento ($++$, $+$, $-$, $--$) e o nível de apoio local do transmitido do próprio nó. O nível nominal que um nó do tipo $++/+/-/--$ transmite a seu nó pai é respectivamente $+1.0/+0.5/-0,5/-1.0$.

Uma hipótese importante do algoritmo é que os nós com apoio local negativo estão desabilitados a transmitir o nível de apoio transmitido a seu nó pai, sendo excluídos do conjunto de nós filhos no cálculo de apoio local. Isto é feito de maneira a evitar que nós

altamente contrários influenciem seus respectivos ascendentes. Isto é necessário também para anular situações indesejáveis nas quais a polaridade original de um nó (de apoio ou oposição) está invertida pelo seu apoio local negativo.

O algoritmo inicia a avaliação pela atribuição de valores de apoio local de +1.0 para todos os nós folha e então sobe a árvore calculando os valores de apoio local correspondentes para todos os nós até o nó do elemento em discussão.

Devido a natureza baseada em texto das contribuições de diálogo e a dependência de domínio do diálogo, são aplicadas técnicas de combinação semântica para melhorar o mecanismo de coordenação. Eleuterio identificou dois parâmetros semânticos com grande potencial para este propósito:

- Proximidade semântica entre Contribuições Textuais, tais como respostas ou argumentos diretos. Isto pode ser útil para descobrir relações escondidas entre as contribuições dos usuários, especialmente em árvores de diálogo extensas com grandes quantidades de informação textual. A disponibilidade de uma ontologia de domínio pode aumentar a combinação de palavras tradicional pelo acréscimo de combinações baseadas em conceito, de acordo com Honkela, citado em [ELE 01].
- Cobertura conceitual, que almeja detectar tópicos não discutidos ou tópicos insuficientemente discutidos nas sessões de diálogo. Tais tópicos podem ser identificados pela análise da ocorrência de certas palavras de domínio em uma determinada árvore de diálogo. Como uma resposta, elementos de discussão específicos podem ser gerados com o objetivo de trazer tais assuntos de volta ao diálogo.

Outras técnicas textuais, como por exemplo Recuperação de Informação Baseada em Ontologias, podem ser aplicadas para descobrir conceitos em meio à informação textual. Uma das principais dificuldades de aplicar raciocínio semântico é a necessidade de modelos de conhecimento compreensivos e bem construídos, que são muito difíceis de desenvolver, mesmo com o auxílio de especialistas em conhecimento. Além disso, a diversidade léxica pode apresentar dificuldades em relacionar conceitos semelhantes de contribuições de usuários diferentes, o que sugere que o raciocínio semântico pode fornecer resultados quando aplicado a domínios muito específicos e com baixa diversidade de terminologia.

Aleven em [ALE 2001] propôs criar um sistema de diálogo tutorial que auxilie alunos a aprender através de auto-explanação. Seu protótipo atual é capaz de analisar as explanações gerais dos alunos em suas etapas de resolução de problemas; reconhecer os tipos de omissões que são observadas nestas explanações e fornecer retorno (feedback). O sistema reúne os alunos em uma forma restrita de diálogo para ajudá-los a melhorar explanações que não são precisas o suficiente.

Aleven desenvolveu um Tutor de Explicações sobre Geometria, que fornece suporte a resolução de problemas. Ele monitora os alunos enquanto eles trabalham na resolução de problemas e fornece assistência na forma de sugestões de retorno sensíveis ao contexto. Este tutor necessita que os alunos expliquem seus passos e reúne os aprendizes em uma forma restrita de diálogo de maneira a auxiliar os alunos a expressar regras de geometria corretamente.

O Tutor de Explicações sobre Geometria está baseado no padrão da Arquitetura Tutor Cognitivo descrita por Anderson, citado em [ALE 2001], adicionado de um componente NLU (Natural Language Understanding – Compreensão da Linguagem Natural). Em cada ciclo de diálogo, este componente desenvolve uma representação semântica da explicação do aprendiz e a classifica de acordo com a hierarquia do sistema de categorias de explicações. O módulo de Tutor Cognitivo verifica então se a explicação do aluno se focaliza sobre uma regra de geometria correta e decide que retorno fornecer ao aprendiz.

Uma fonte de conhecimento importante é a hierarquia de categorias de explicações, que constitui o conhecimento de conteúdo pedagógico do sistema. As categorias de explicações nesta hierarquia representam maneiras de declarar cada regra de geometria corretamente, da mesma forma que freqüentemente ocorrem maneiras de declarar regras incorretamente. Cada nó da hierarquia representa uma classe de explicações que têm o mesmo significado, mas podem ter formas de superfície altamente diferentes.

Categorias de explicações na parte de baixo da hierarquia representam formas de declaração de regras de triângulos isósceles corretas e completas. Categorias de explicações mais altas representam formas de declarar estas regras de maneiras progressivamente mais incompletas. A hierarquia também contém informações sobre como reagir às explicações dos aprendizes. Anexado a cada categoria existe uma mensagem de retorno que é apropriada quando uma explicação feita por um aprendiz é classificada abaixo daquela categoria. Aleven identificou aproximadamente 140 categorias de explicações relacionadas a 25 regras de geometria que constituem a unidade de Ângulos do tutor. Um ponto chave é que estas categorias foram induzidas a partir de observações de dados de aprendizes reais, contidos em diversos arquivos de explicações de alunos. Assim esta rede captura categorias que ocorrem freqüentemente à medida em que os aprendizes progredem.

A hierarquia é implementada como uma base de conhecimento Loom (Tear). Esta base de conhecimento também contém uma ontologia do domínio que consiste de objetos de geometria, tais como ângulos e linhas, além de relações como congruência e adjacência. A ontologia cobre o material da unidade de ângulos do currículo do tutor. Atualmente, a base de conhecimento contém definições de cerca de 310 conceitos e 90 relações.

O componente NLU analisa as contribuições dos aprendizes por meio de uma abordagem baseada em unificação. Aleven empregou o analisador LCFLEX, um analisador gráfico ativo combinado com um unificador de estrutura de característica. Desenvolveu-se uma gramática de cerca de 200 regras. O analisador e o unificador construíram uma estrutura de característica codificando a sintaxe da frase, além de guiar a hierarquia Loom

para construir uma representação semântica. No processo, o classificador Loom testou a coerência da representação semântica com respeito a obstáculos semânticos expressados na ontologia de domínio do sistema. A hierarquia Loom classificou a representação semântica de acordo com as categorias das explicações.

O módulo de Tutor Cognitivo determina como reagir aos aprendizes. Primeiramente, o tutor estabelece qual regra de geometria o aprendiz deverá explicar. Esta regra deve ser uma que justifique a etapa de resolução do problema corrente. O tutor determina o conjunto de regras de geometria aplicáveis, por meio de seu modelo cognitivo de resolução de problemas de geometria. Este modelo captura o conhecimento típico e desejado de estudantes de geometria, representado na forma de regras de produção. O tutor seleciona então uma mensagem de retorno apropriada. Se a explicação do aprendiz formar um enunciado completo de uma regra de geometria relevante, o tutor aceitará a explicação. Se a explicação for somente uma declaração parcial, o tutor selecionará uma mensagem de retorno apropriada. Se a explicação for apenas o nome de uma regra de geometria, o tutor pedirá ao aprendiz para declarar a regra. Se a explicação se focaliza sobre uma regra de geometria errada, o tutor informará ao aprendiz este fato. Quando a base de conhecimento falha em produzir uma análise, o classificador estatístico é utilizado para determinar se a explicação do aprendiz está focalizada sobre a regra de geometria correta; em caso positivo, o tutor emitirá uma mensagem de retorno dizendo que o aluno parece estar no caminho certo.

Atualmente, as ações do sistema em relação à cada diálogo estão baseadas unicamente na classificação da última tentativa de explicação do aprendiz. O tutor pode responder a tipos de omissões muitas vezes vistas nas explicações dos alunos e pode, às vezes, produzir um senso de diálogo coerente; no entanto, a abordagem possui limitações, por exemplo, o sistema não tem como lembrar do que veio antes no diálogo, não sendo capaz de detectar quando os aprendizes retrocederam ou estagnaram; além disso, o tutor não é capaz de utilizar múltiplas estratégias ou guiar os aprendizes através de uma linha direta de raciocínio.

Rickel em [RIC 2001], declara ter por objetivo desenvolver tutores computacionais que colaborem com aprendizes na resolução de tarefas em ambientes simulados. Rickel tentou integrar Sistemas Tutores Inteligentes (ITS – Intelligent Tutoring Systems) com Sistemas de Diálogo Colaborativo (CDS – Collaborative Dialogue Systems). Para integrar essas duas áreas, foi desenvolvido um agente tutorial dentro do sistema Collagen, um sistema baseado em uma longa linha de pesquisa sobre discurso colaborativo. O sistema Collagen mantém um modelo do estado do discurso compartilhado pelo aprendiz e o agente. O estado do discurso inclui informações sobre o foco de atenção corrente e os planos mutuamente compartilhados dos colaboradores. Os agentes construídos usando Collagen utilizam o estado do discurso para gerar uma agenda de atos de discurso candidatos, incluindo tanto ações físicas como declarações, e então escolhem um ato para realizar ou declarar. O agente tutorial desenvolvido, PACO (Pedagogical Agent for Collagen – Agente Pedagógico para Collagen), ensina aos aprendizes tarefas procedurais em ambientes simulados.

A abordagem pedagógica deste agente baseia-se no modelo de aprendizagem de estudo de Collins, citado em [RIC 2001], que requer duas capacidades:

- O tutor deve ser capaz de realizar e explicar a tarefa.
- O tutor deve ser capaz de monitorar o aprendiz enquanto ele realiza a tarefa, fornecendo assistência quando necessário, além de retornos (feedback) críticos ou positivos. À medida em que o aluno ganhar proficiência, a assistência deve diminuir.

Se as tarefas e cenários não estiverem interligadas, o agente PACO pode ser implementado da seguinte forma: o tutor primeiro demonstrará a tarefa completa, e então repetidamente deixará o aluno praticar a tarefa fornecendo assistência, se necessário. No entanto, tarefas diferentes muitas vezes compartilham sub-tarefas ou ações e cenários diferentes e muitas vezes requerem variantes da mesma tarefa. Portanto, a qualquer momento um nível de habilidade do aluno pode diferir ao longo de diferentes partes da tarefa. Por este motivo PACO utiliza um modelo de aluno para intercalar dinamicamente a demonstração e a prática orientada. Como o aluno e PACO progredem por meio de uma tarefa, o sistema Collagen irá identificar repetidamente o conjunto dos próximos passos válidos no plano para resolver a tarefa corrente. PACO consulta o modelo do aluno para verificar se o aprendiz tem conhecimento suficiente para escolher a próxima etapa; se assim for, esperará que o aprendiz passe para a próxima tarefa e só fornecerá assistência ao aluno se este cometer um engano ou pedir ajuda; caso contrário, PACO intervirá e ensinará ao aluno o que fazer.

O agente PACO e o aprendiz trabalham juntos por meio de tarefas. A iniciativa passará de um para o outro, dependendo das experiências anteriores do aprendiz. Sempre que PACO decidir que a iniciativa deve ser trocada, ele informará isto ao aprendiz, através de comentários verbais. PACO representa os procedimentos que irá ensinar, usando a linguagem declarativa de Collagen para conhecimento procedimental específico de domínio. Este conhecimento serve como um modelo de como as tarefas de domínio devem ser realizadas. Cada tarefa está associada com uma ou mais fórmulas, ou seja, procedimentos para realização da tarefa. Cada fórmula consiste de diversos elementos tirados de uma representação relativa de plano padrão.

O principal valor de Collagen para construir sistemas tutores é que ele fornece um modelo geral de diálogo colaborativo baseado em duas partes principais:

- Uma representação do estado do discurso
- Um algoritmo de interpretação de discurso, que usa planos de reconhecimento para atualizar o estado do discurso, dadas as ações e declarações do usuário e do agente.

Collagen divide o estado do discurso em três componentes inter-relacionados:

- **Estrutura Lingüística:** É implementada como um histórico de interações segmentadas, Collagen agrupa o histórico do diálogo em uma hierarquia de segmentos de discurso. Cada segmento é uma seqüência contígua de ações e declarações que contribuem para algum propósito, como realizar uma tarefa ou sub-tarefa.
- **Estado Atencional:** Registra o que o aprendiz e o agente estão discutindo e/ou trabalhando no momento; é representado por uma pilha de propósitos de discurso chamada de pilha de foco. Quando um novo segmento de discurso é iniciado, seu propósito é colocado na pilha. Quando um segmento de discurso é completado ou descontinuado, seu propósito é retirado da pilha. O estado atencional mantido por Collagen mantém um registro de qual participante conserva a iniciativa conversacional, o que permite a PACO decidir quando passar a iniciativa para o aprendiz.
- **Estrutura Intencional:** Representa as decisões que devem ser tomadas como resultado das ações e declarações, independente de sua ordem. Collagen representa a estrutura intencional como árvores de plano, que são uma implementação parcial de Nós de Planos Compartilhados (SharedPlans), sugerido por Grosz em [RIC 2001], que representam concordâncias mútuas sobre intenções e a estrutura da árvore representa os relacionamentos de sub-objetivo entre estas intenções. As árvores de plano também registram outros tipos de decisões, tais como, se uma fórmula foi escolhida para uma tarefa, se quaisquer de seus parâmetros foram determinados e quem é responsável por realizar esta tarefa.

O núcleo de Collagen é o algoritmo de interpretação de discurso que especifica como atualizar o estado do discurso quando uma nova ação ou declaração é fornecida pelo aluno ou pelo agente. Seu objetivo é determinar como os atos correntes contribuem para a colaboração.

Collagen ampliou o algoritmo de interpretação de discurso de Lochbaum, citado em [RIC 2001], com reconhecimento de plano, que pode reconhecer quando um ato contribui para um propósito de segmento de discurso através de um ou mais atos implícitos. O algoritmo de interpretação de discurso de Collagen segue os seguintes procedimentos: Se o ato corrente contribui para o propósito de segmento de discurso corrente, este é adicionado ao segmento e a árvore de plano é atualizada de acordo. Caso contrário, Collagen pesquisa através da árvore de planos para verificar se um ato contribui para qualquer outra ação no plano; neste caso, e se o ato for uma próxima etapa válida, ele representa uma transferência de foco. Collagen retira todos os propósitos da pilha que não são pais da etapa e então insere qualquer propósito necessário até que o ato esteja em foco. Finalmente, se nada na árvore de plano combina com o ato presente, ele é tratado como uma interrupção e inserido na pilha sem retirar-se nada.

Collagen foi recentemente atualizado para realizar reconhecimento de plano de erro. Se ele não puder encontrar uma interpretação correta de um ato, ele sistematicamente

pesquisa por extensões da árvore de plano que explicariam o ato corrente, se algum obstáculo foi resolvido. O reconhecimento do plano de erro tenta encontrar interpretações plausíveis dos erros do aprendiz, fornecendo uma capacidade independente de domínio para diagnóstico do aluno. Collagen está sendo atualizado de maneira a poder utilizar informações causais nas fórmulas, para reparar planos depois da ocorrência de ações ou eventos externos incorretos.

A arquitetura de PACO é composta por um simulador, Collagen e o próprio agente. Collagen realiza muito poucas hipóteses sobre o simulador. Primeiramente, ele presume que o aprendiz e o agente possam realizar ações de domínio e possam observar as ações tomadas por cada um. Collagen fornece um API para tais mensagens de evento, de forma que ele seja capaz de interpretá-las. O simulador pode, opcionalmente, especificar uma localização de tela para ações de domínio, o que permite que o agente use um cursor em forma de mão apontando a fim de atrair a atenção do usuário para um objeto ou indicar que o agente está realizando uma ação.

Collagen representa declarações usando uma linguagem de discurso artificial derivada do trabalho de Sidner, citado em [RIC 2001]. A linguagem destina-se a incluir os tipos de declarações que as pessoas utilizam durante a colaboração de tarefas. Atualmente a linguagem de Collagen inclui tipos de declarações como:

- Concordância, como “Yes” e “OK”.
- Discordância, como “No”.
- Oferecimento de uma tarefa ou ação, como, “Let’s engage engine one” (Vamos ligar o motor número um).
- Indicação de quando uma tarefa foi completada, como, “We succeeded in stopping engine two” (Conseguimos desligar o motor número dois).
- Cancelamento de uma tarefa.
- Questionamento ou sugestão sobre o valor de um parâmetro para uma tarefa ou ação.
- Questionamento ou sugestão de como uma tarefa deveria ser realizada.
- Questionamento do que deveria ser feito em seguida, como “What next ?” (Qual o próximo passo ?).

O trabalho atual está atualizando a linguagem de Collagen para incluir elementos adicionais da linguagem de Sidner, especialmente para suportar negociação sobre decisões de tarefa.

Para evitar questões de compreensão de linguagem natural, Collagen fornece uma janela para possibilitar que o aprendiz construa declarações e para apresentar as declarações do agente. Collagen converte sua linguagem de discurso interna em strings na língua inglesa, utilizando uma combinação de modelos de texto de domínio independente e de domínio específico (opcional). Os aprendizes constroem declarações, selecionando-as a partir de um menu de tipos de declarações e de um menu de declarações. Os aprendizes podem modificar qualquer declaração escolhida, selecionando-a e escolhendo uma substituição de frase. Opcionalmente, Collagen pode também utilizar um software de reconhecimento de discurso para permitir ao aprendiz falar estas declarações ao invés de criá-las e pode também utilizar software de síntese de discurso para permitir ao agente fazer suas declarações.

PACO utiliza diversos elementos de estado do discurso para gerar seus atos de discurso, incluindo o foco de atenção, a iniciativa e as árvores de plano. O foco de atenção, é usado para evitar o ensino de uma etapa, a menos que seu propósito esteja em foco. A pilha de foco também demonstra quando o aprendiz interrompeu a tarefa atual, o que leva PACO a gerar um ato de discurso que finalizará a interrupção presente. Em adição ao foco compartilhado mantido por Collagen, PACO também mantém um foco privado porque ele prefere finalizar o ensino de uma ação antes de seguir em frente. Se o aluno começa a trabalhar em outra parte do plano, enquanto ainda existem etapas legais dentro do foco privado de PACO, então o agente adicionará uma ação de Foco Correto na agenda. Se o aluno não retornar à tarefa anterior por ele mesmo, PACO gerará novamente a ação de foco correto.

As várias condições para geração de atos de discurso são fáceis de processar devido aos dados mantidos por Collagen. Por exemplo, diversos dos atos operam sobre as próximas ações válidas, que se referem às etapas do plano que podem ser executadas em seguida, baseadas em pré-condições e ordenação de obstáculos. Collagen processa esta informação durante a interpretação do discurso. Além disso, o reconhecimento de erro de Collagen processa as condições necessárias para gerar os vários sub-casos de retorno negativo. Finalmente, quando o aprendiz pede ajuda, Collagen insere um propósito de discurso de ajuda ao aprendiz na pilha que permanecerá ali até que o agente forneça a ajuda.

A utilização das capacidades genéricas de Collagen para registrar informações sobre o aprendiz, permite a PACO manter um simples modelo de cobertura (overlay), que registra em uma fórmula cada etapa a que o aprendiz foi exposto. O modelo de aprendiz de PACO também registra o recebimento de notificações pelo aprendiz das ações por ele concluídas.

A representação genérica de Collagen é utilizada pelas fórmulas para armazenar conhecimento de domínio específico indicando porque determinadas ações precisam de ser realizadas, ou seja, o conhecimento de domínio de PACO inclui fórmulas que atingem o sub-objetivo de explicar o porquê de uma ação, ou mais especificamente, porque uma etapa de uma fórmula deveria ser realizada.

Normalmente, estas fórmulas são compostas de uma ou mais declarações de texto escritas por um especialista de domínio, mas fórmulas de explanação podem conter

quaisquer tipos de ações primitivas ou de resumo. Sempre que PACO gerar um ato de discurso candidato para ensinar uma etapa, ele também verifica se uma fórmula de explanação existe para aquela etapa. Em caso positivo, e se a etapa ainda não tenha sido explicada, PACO gera um ato de discurso candidato de execução da primeira etapa da fórmula.

As condições para geração de atos de discurso representam condições necessárias, mas não suficientes para PACO realizar um ato. Uma vantagem de tornar explícitas todas as condições necessárias para um ato de discurso é tornar mais fácil alimentar PACO com novos atos de discurso ou proporcionar a outros agentes a habilidade de realizar as ações tutoriais de PACO. No entanto, esta abordagem deixa em aberto a questão de como escolher qual ato realizar. PACO faz esta escolha baseado nas posições dos atos de discurso, por exemplo, PACO prefere ceder a iniciativa quando o aprendiz sabe o que fazer em seguida, ao invés de ensiná-lo ou lembrá-lo do que fazer.

O trabalho de [RIC 2001] focalizou-se sobre o pragmatismo da compreensão da linguagem natural, isto é, o uso de um algoritmo de interpretação de discurso e uma rica representação do estado do discurso. Rickel afirma que diálogo tutorial será mais natural para os aprendizes se o tutor seguir os princípios de diálogos colaborativos humanos, aos quais muitas pesquisas em lingüística computacional se tem dedicado.

3 Agentes

O termo “agente” foi utilizado para referenciar desde simples processos até software e/ou hardware altamente avançado, denotando uma entidade criada para realizar uma tarefa específica ou um conjunto de tarefas [GIR 99] [GIR 99a]. Dessa forma, não existe um consenso sobre a definição de agentes [GIR 99] [GIR 99a] [MOI 99].

Segundo [RUS 95], um agente pode ser definido como qualquer objeto que possa ser visto como percebendo seu ambiente através de sensores e agindo sobre ele através de causadores de efeito. Um agente humano possui olhos, orelhas e outros órgãos sensoriais, como braços e mãos, pernas e outras partes do corpo, que são todos causadores de efeito [RUS 95]. Demazeau, citado por [HÜB 95], apresenta uma definição semelhante, acrescentando ainda que um agente deve ser capaz de comunicar-se com outros agentes e de exibir um comportamento autônomo.

Ferber, citado em [CAZ 97], dá uma definição similar à descrita acima, mas talvez mais completa e concisa: "Chama-se agente a uma entidade abstrata que é capaz de agir sobre si mesma e sobre seu próprio ambiente, que dispõe de uma representação parcial deste ambiente, e que, em um universo multiagente, pode comunicar-se com outros agentes e cujo comportamento é consequência de suas observações, de seu conhecimento e das interações com outros agentes".

Agentes artificiais, assim como os seres humanos, possuem conhecimentos, crenças, intenções, compromissos, decisões e desejos [FRO 97] [OHA 96]. Estas características (com exceção do conhecimento), são chamadas de “Estados Mentais” e podem-se acrescentar ainda objetivos, capacidades e expectativas, de acordo com [VIC 89].

Um agente atua em um ambiente de acordo com suas próprias metas e conhecimentos. Os efeitos das ações do agente são percebidos pela produção de eventos, que correspondem a modificações do ambiente [FRO 97]. No entanto, eventos externos, não causados pelo próprio agente, podem fazer com que este reaja, produzindo novas ações [BIG 98]. Um evento é qualquer coisa que altere o ambiente ou qualquer coisa que o agente perceba [BIG 98]. Agentes são inteligentes se apresentarem a capacidade de serem flexíveis e adaptativos, compondo suas próprias metas e realizando-as por ações eficientes [FRO 97]. Bratman, citado por [HÜB 95], afirma que um agente planeja suas ações de acordo com suas crenças e desejos para satisfazer suas intenções.

Um agente pode ser implementado como um sistema especialista, um robô ou um sistema interagindo com o ser humano ou com outros sistemas [PER 97]. Cerveira, citado por [PER 97], considera que um agente se trata, normalmente, de um software atuando em nome de um usuário, podendo ainda, agir em função de outros agentes computacionais, com o propósito de auxiliar na realização de determinadas tarefas. Tais agentes podem realizar trabalhos rotineiros para os seus usuários, bem como auxiliá-los em tarefas complexas.

Segundo Colazzo, citado em [GIR 99a], do ponto de vista de software, um agente “é um programa que possui um plano específico de ação definido em um domínio limitado e um padrão de comportamento que lhe permite mudar, no momento certo, sua própria interação com o mundo, dependendo dos estímulos do ambiente”. Assim, todo o software agente é um programa; no entanto, nem todo programa é um agente [GIR 99a].

Um programa agente é uma função que implementa o mapeamento das percepções do agente em ações. Este programa deve ser executado em uma arquitetura computacional que pode necessitar de hardware de propósito especial para realizar tarefas como processamento de imagens, por exemplo. Assim, um agente pode ser descrito como a união do programa com sua arquitetura, sendo que esta arquitetura é uma das principais responsáveis por transmitir as percepções do ambiente ao programa [RUS 95].

3.1 Agentes Pedagógicos

Agentes podem também ser utilizados para auxiliar no processo de ensino-aprendizagem, quando passam a chamar-se agentes pedagógicos [GIR 99]. Os agentes são utilizados como softwares que auxiliam na atividade educacional, existindo ainda projetos mais ambiciosos que tentam utilizar os agentes e a Internet para auxiliar no ensino à distância [CAZ 97] [PER 97].

Giraffa, em [GIR 99a] cita diversos tipos de agentes pedagógicos, tais como tutores, mentores, assistentes, MOO (um ambiente virtual na Internet, em que pessoas se conhecem e se comunicam), agentes trabalhando em aplicações na Internet, agentes aprendizes e agentes mistos, capazes de ensinar e aprender.

Agentes pedagógicos têm um conjunto de objetivos de ensino e planos para atingi-los (estratégias de ensino), além de recursos associados ao ambiente de aprendizagem. Agentes tutores são entidades cujo propósito é comunicar-se com o aluno a fim de preencher eficientemente sua função de tutor como parte da missão pedagógica do sistema [GIR 99a].

As capacidades de comunicação e interação apresentadas pelos agentes, constituíram a razão fundamental para sua utilização ao auxiliar no processo de ensino-aprendizagem, tendo em vista que um agente pode-se adaptar e aprender durante uma sessão de aprendizagem. Além disso, e graças a essas capacidades, em um mundo com diversos agentes, pode haver colaboração entre eles para atingir objetivos comuns. [GIR 99a].

Agentes pedagógicos podem agir como tutores virtuais, alunos virtuais ou colegas de aprendizado virtual, que podem ajudar os alunos no processo de aprendizagem [GIR 99a].

Giraffa, em [GIR 99], classifica as estratégias básicas de aprendizagem em ambientes que utilizam agentes em:

- Aprendizagem com um colega, onde um agente artificial age como um colega no sistema e discute as tarefas com o aluno, auxiliando-o a adquirir conhecimento. Este colega possui objetivos e nível de conhecimento semelhantes ao aprendiz. O papel do tutor é alternar os problemas e criticar as soluções apresentadas.
- Aprendizagem através do ensino, onde um estudante humano ensina um agente artificial, que age como colega, forçando o aluno a organizar seu pensamento e ampliar seu conhecimento e sua maneira de manipulá-lo.
- Aprendizagem por perturbação, onde um tutor, um agente “encrenqueiro” (troublemaker) e um aluno utilizam o mesmo espaço de aprendizagem. Um problema é apresentado ao aluno e ao encrenqueiro e este pode ter diversos comportamentos, como dar respostas erradas, oferecer a solução certa e dar sugestões. O aluno explica sua opinião ao encrenqueiro sob o controle do tutor. Se o aluno demonstrar-se incapaz de solucionar o problema, o tutor lhe fornecerá a solução correta.

De acordo com Wang, citado em [GIR 99a], sistemas de aprendizagem social são ambientes de aprendizagem emergente que permitirão a múltiplos alunos e agentes trabalhar em um mesmo computador ou através de máquinas conectadas, sob diferentes protocolos de atividades de aprendizado.

Também pode ser considerada uma abordagem cooperativa entre o aprendiz e o sistema. Muitos grupos de pesquisa criaram simulações de uso de ambientes em que o processo de ensino-aprendizagem é simulado por um conjunto de agentes a fim de produzir interação entre eles e permitir observar as mudanças dinâmicas que ocorrem durante o processo de interação [GIR 99a].

Um ambiente de ensino-aprendizado, pode ser considerado como uma sociedade composta por agentes autônomos, podendo ser tanto humanos como artificiais. Alguns destes agentes representam o papel de tutores e outros os de alunos, sendo que estes papéis podem ser mudados. Nesta sociedade, todos os agentes estão preocupados em construir uma base de conhecimento comum sobre um domínio particular; assim, o comportamento de um agente pode afetar o comportamento da sociedade como um todo e o comportamento da sociedade pode alterar o comportamento de um ou mais agentes individuais [GIR 99a].

Mitsuru, citado em [GIR 99a], declara que ambientes sociais implicam em trabalho colaborativo e que a importância principal educacional do aprendizado colaborativo é melhorar a motivação dos participantes, estimulando reflexões maduras sobre sua própria compreensão e compartilhando seus resultados.

Giraffa, em [GIR 99a] afirma que aprendizado cooperativo significa que dois ou mais alunos tem a mesma tarefa, que é dividida em sub-tarefas e estas, por sua vez, atribuídas a cada agente, que não interfere no trabalho dos outros. A solução final será composta pela união dos resultados obtidos pelos agentes. Já em um aprendizado colaborativo, todos os agentes têm a mesma tarefa e cada agente sugere sua própria solução e o grupo discute sobre ela. Neste caso, a solução final é o produto da interação do grupo.

3.2 Sistemas Tutores Inteligentes

Segundo Burns, citado em [MOI 99], um sistema tutor inteligente deve apresentar três características:

- O especialista embutido (o tutor) deve ser capaz de realizar inferências ou resolver problemas do domínio ou dentro do domínio.
- Deve possuir a capacidade de deduzir o intervalo entre o aluno e o conhecimento almejado.
- Deve possuir a capacidade de implementar estratégias de ensino de forma inteligente a fim de melhorar a capacidade do especialista em transmitir seu conhecimento para o aluno.

[CAS 99] acrescenta ainda outras características, como:

- Flexibilidade para mudanças, evoluções e atualizações na base de conhecimento ou no conjunto de procedimentos de treinamento, de maneira a suportar reusabilidade e, caso as mudanças forem muito grandes, constituir um novo sistema tutor.
- Suporte a operações multi-usuário e a variedade de interfaces de hardware e software.
- Um ambiente para aprendizagem, baseado em simulação, como estratégia de aprendizagem.

Segundo Linton, citado em [CAS 99], um sistema tutor inteligente constitui-se em uma pesquisa de inteligência artificial no domínio da educação auxiliada por computador, que almeja descobrir maneiras de produzir tutores inteligentes sustentados por ambientes computacionais que representem as fases de conhecimento do aprendiz, estabelecer o aprendizado intermediário e final e prover guias instrucionais para auxiliar e acelerar a aprendizagem.

Já de acordo com Moussale, citado em [MOI 99], sistemas tutores inteligentes são programas que podem identificar as intervenções feitas pelos alunos, diagnosticar suas deficiências, dialogar com os mesmos e orientá-los.

Inicialmente, os sistemas tutores inteligentes tentavam reproduzir o comportamento de um tutor humano inteligente que pudesse adaptar seu método de ensino ao ritmo de aprendizagem do aprendiz, essa interação era controlada pelo tutor. Estes sistemas eram difíceis de controlar do ponto de vista pedagógico e não muito eficientes [GIR 99a].

Os avanços das capacidades de hardware, o grande aumento do uso de computadores pessoais e os avanços do software permitiram o uso de computadores por grandes grupos de aprendizes. Além disso, avanços em pesquisas educacionais contribuíram para o desenvolvimento de novos softwares educacionais. Estes softwares requeriam novas técnicas de projeto e desenvolvimento, sendo uma de suas principais características uma arquitetura modular, utilizada para suportar o aprendizado adequado [GIR 99a].

Apesar desses avanços, o modelo do aluno e o comportamento do tutor artificial permaneceram ainda um tanto insuficientes dentro do ambiente. O comportamento do tutor está relacionado com as estratégias selecionadas e suas intenções, desejos e expectativas sobre os alunos. Já o modelo do aluno está relacionado com as seqüências de estratégias tomadas pelo tutor durante o processo de interação [GIR 99a].

Recentemente, os ambiente educacionais passaram a ter mais do que um aluno interagindo com o mesmo ambiente, sob a supervisão de um tutor artificial. Isto foi possível graças ao desenvolvimento de hardware e software que passou a permitir a conexão de diversas pessoas através do uso de redes de computadores. Os pesquisadores passaram então a utilizar estas novas tecnologias para melhorar seus ambientes educacionais [GIR 99a].

No entanto, ainda não é possível criar um sistema tutor inteligente ideal, principalmente pela falta de conhecimentos completos do funcionamento do processo de aprendizagem humano e de limitações ainda existentes em hardware e software [GIR 99a] [GIR 99b].

Galvis, citado em [CAS 99], afirma que um sistema tutor inteligente deve possuir quatro grandes etapas:

- Introdutória, que gera a motivação, centra a atenção e favorece a percepção seletiva.
- Orientação inicial, onde é codificado, armazenado e retido o que foi aprendido.
- Aplicação, em que se evoca e transfere o que foi aprendido.

- Retroalimentação, onde é demonstrado o que foi aprendido e se recebe uma retroalimentação de reforço.

Giraffa, em [GIR 99] defende o uso de diversos tutores em um sistema de tutor inteligente, afirmando que um único tutor detém todo o conhecimento, que deve ser armazenado nele previamente; além disso o aluno não se sente confiante para questionar o tutor. Giraffa destaca algumas vantagens de se utilizar um sistema de tutor inteligente com diversos tutores, tais como:

- A distribuição do conhecimento entre os diversos tutores. Cada tutor possui seus próprios estados mentais e planos de ação, o que permite a utilização de técnicas pedagógicas diversas. Além disso, permite criar uma sociedade de agentes tutores que se comunicam entre si e determinam qual deles realizará uma determinada tarefa e auxiliará um determinado perfil do aluno.
- Maior flexibilidade na interação com o tutor.
- O aluno pode transmitir seus conhecimentos a um tutor que os repassará a outros alunos ou pode transmiti-los diretamente aos outros alunos.
- Aumento da autonomia do aluno, possibilitando-lhe questionar as proposições de seus tutores e seus colegas.

Uma das atividades de um professor é observar o comportamento dos alunos, a fim de avaliar seus aspectos motivacionais e afetivos, objetivando definir ou melhorar suas estratégias de ensino [MOI 99]. A qualidade pedagógica de um sistema tutor inteligente depende muito da estratégia de ensino selecionada. Vários fatores implicam na escolha da estratégia, tais como [GIR 99b]:

- O nível de conhecimento do aluno.
- O domínio do aluno.
- A motivação do aluno.
- As características afetivas do aluno.

A utilização da mesma estratégia não produz um efeito satisfatório para todos os alunos [GIR 99b]. Frasson, citado em [GIR 99b], declara que cada estratégia possui vantagens específicas, cada aluno responderá melhor a uma estratégia diferente. A melhor maneira de determinar qual a melhor estratégia de ensino para um determinado aluno é através dos resultados adquiridos durante as interações anteriores do aluno [GIR 99b]. Cronbach, citado em [CAS 99], afirma que bons instrutores são capazes de se adaptarem intuitivamente, baseados nas experiências e impressões adquiridas de cada aluno.

Murray e Keller, citados em [GIR 99b], sugerem que um tutor deve ter dois níveis de planejamento:

- Estratégias pedagógicas, que armazenam o conhecimento sobre como ensinar.
- Táticas pedagógicas, que contêm as ações para efetivar a estratégia selecionada.

Casas, em [CAS 99] declara que um sistema tutor inteligente, deve possuir, entre suas principais características, a capacidade de fornecer ensino adaptativo, que possibilite a desenvolvimento de atividades apropriadas para que o nível de habilidade do usuário evolua, quando este acumular experiência suficiente no ensino. Além disso, um projeto de sistema tutor inteligente deve permitir reusabilidade, ou seja, possibilitar que seu conhecimento, armazenado em sua base de dados, possa ser modificado, atualizado ou substituído, sem qualquer modificação de seu código, de maneira a simplificar atualizações na área de ensino do sistema ou a criação de outros sistemas tutores em áreas de ensino diferentes.

A utilização de agentes e sistemas multiagentes em sistemas tutores inteligentes começou a tornar-se forte a partir do momento em que a participação do estudante no processo de ensino-aprendizagem aumentou e foi um dos fatores que permitiu o surgimento dos Sistemas de Aprendizagem Cooperativos ou Sistemas de Aprendizagem Sociais [GIR 99b].

Segundo Bercht e Moussale, citados em [MOI 99], a arquitetura clássica de um sistema tutor possui os seguintes componentes funcionais:

- A base de conhecimento do domínio ou especialista, que armazena o conteúdo a ser ensinado, que abrange conhecimento, fatos e regras do domínio.
- O modelo do aluno, que armazena o modelo cognitivo de cada aluno. Este modelo contém informações tais como as crenças do aluno, o que o aluno sabe a respeito do domínio e de outros domínios afins, o conhecimento do aluno (correto, incerto ou incorreto) e o conhecimento que o sistema possui sobre o aluno. Viccari, citada em [MOI 99], afirma que este modelo pode ainda conter a especificação dos objetivos do aluno, seu modelo de intenções, planos, atitudes e procedimentos de inferência. Este módulo necessita ser constantemente atualizado, pois é através dele que o aluno é avaliado pelo tutor, para definir uma estratégia de ensino adequada e personalizada para o mesmo.
- O módulo pedagógico ou estratégia de ensino, que mantém estratégias para a aplicação do material instrucional ao aluno, determina as diversas etapas no processo de ensino, representa o conhecimento de como se deve apresentar o conteúdo ao aluno e as táticas referentes ao ensino.
- A interface com o usuário, que estabelece a comunicação entre o aluno e o ambiente e tem como principais funções o monitoramento de desempenho, o

comportamento do aluno e a apresentação do material instrucional. Esta interface deve ser adaptável a diferentes usuários, amigável, rápida e objetiva, em suma, procurar apresentar características que auxiliem a aprendizagem do aluno.

Viccari, citada em [MOI 99], acrescenta ainda um módulo de controle a esta arquitetura, que é responsável pelo controle dos módulos descritos anteriormente.

Moissa, em [MOI 99] descreve diversos tipos de arquiteturas possíveis para a construção de um sistema tutor inteligente, tais como:

- Arquitetura aberta, proposta por Oliveira e Viccari, citados em [MOI 99]. Esta arquitetura é estruturada nos módulos de teoria do ambiente, de ensino, de aprendizagem, do domínio e do modelo do aluno.
- Arquitetura por dimensões de relação, proposta por Burns, citado em [MOI 99]. Esta arquitetura enfatiza os relacionamentos de suas funções, que abrangem três dimensões: dimensão de comunicação homem-máquina, que trata da integração, dos módulos de interface e base de domínio; dimensão instrucional, que trata da comunicação, individualização e integração das funções do especialista instrucional, além da interface e modelo do aluno; dimensão do conhecimento, que trata dos relacionamentos entre os módulos, tais como interfaces inteligentes, simulações e domínios.
- Arquitetura orientada a objetos, proposta por Warren, citado em [MOI 99]. Os componentes desta arquitetura são entidades independentes e se comunicam por meio de mensagens. O componente executivo controla o processo do sistema, existindo também os componentes: modelo do especialista, gerador de explicações, controle de apresentação, interface multimídia, gerenciador curricular, modelo do aluno, ambiente instrucional, administrador de treinamento e simulador.
- Arquitetura baseada em sociedades de agentes, proposta por Bica, citado em [MOI 99]. Esta arquitetura está baseada em uma sociedade de agentes autônomos, persistentes e que se comunicam através de mensagens KQML. Estes agentes tem como tarefa: recuperar o conhecimento do domínio sobre cada conteúdo a ser ensinado ao aluno, propor exercícios, avaliar respostas, elaborar e controlar a apresentação dos exemplos ao aluno e tratar da comunicação entre a interface de frames.

[MOI 99] declara que os estados motivacionais do aluno afetam seu desempenho durante o processo de ensino-aprendizagem para melhor ou para pior. Assim, deve-se procurar identificar os estados motivacionais e tentar alterá-los, para melhorar o desempenho do aluno. Keller, citado em [MOI 99], destaca os seguintes estados motivacionais:

- **Confiança:** o nível de confiança de um aluno é medido pela convicção que este possui de sua capacidade em executar uma determinada tarefa. Alunos com pouca confiança evitam as tarefas difíceis ou desistem sem tentar solucioná-las.
- **Esforço:** o nível de esforço pode ser medido pela insistência do aluno em resolver um problema. Esforço implica em motivação.
- **Independência:** o nível de independência pode ser medido pelo número de vezes em que o aluno pede auxílio para resolver um problema. Um aluno possui pouca independência quando pede auxílio frequentemente e muita independência quando raramente ou nunca pede auxílio.
- **Atenção:** o nível de atenção pode ser medido pela capacidade de concentração que um aluno pode dedicar ao problema. É um fator dinâmico que pode ser facilmente alterado pelas ações do tutor ou eventos no meio ambiente.

Soldato, citado em [MOI 99] destaca quatro métodos para identificar os estados motivacionais de um aluno:

- Questionários no início das sessões, que tentam definir a autoconfiança, a afinidade com situações desafiadoras e motivação para estudar um determinado domínio. No entanto, os resultados destes questionários não são dinâmicos, ao contrário das motivações dos alunos.
- Comunicação com o estudante durante a interação, que é um método mais dinâmico para capturar os estados motivacionais dos alunos.
- Solicitações de ajuda e perseverança para completar a tarefa, que determinam o grau de confiança, esforço e independência do aluno.
- Auto-avaliação do estudante de seus estados motivacionais durante a interação, um método que depende exclusivamente do aluno e de sua vontade de expressar seus estados motivacionais.

O ambiente AME-A, que será descrito no quinto capítulo desta dissertação, é um ambiente de ensino-aprendizagem composto por agentes tanto humanos como artificiais, que procuram auxiliar o agente humano a ensinar e/ou aprender. Este ambiente pode ser operado na forma tutorial ou na forma de livre navegação, como se fosse um hiperdocumento.

4 KDT - Knowledge Discovery in Text – Descoberta de conhecimento em texto

O objetivo deste capítulo é possibilitar ao leitor uma introdução à área de KDT, definindo suas principais características e objetivos e detalhando algumas de suas técnicas mais conhecidas. Este capítulo se justifica devido ao fato da ferramenta desenvolvida e descrita neste trabalho ter se baseado em características de algumas técnicas de KDT.

KDT (Knowledge Discovery in Text – Descoberta de conhecimento em texto), também conhecido como Text Mining (Mineração de Texto) ou Document Mining (Mineração de Documentos), é um ramo recente de pesquisa da área de Data Mining [GOT 97], mais especificamente da área de KDD (Knowledge Discovery in Databases) [KOD 99] [LOH 2000]. Data Mining é um processo que busca extrair informações compreensivas, válidas e anteriormente desconhecidas de bancos de dados, de outras fontes de dados da empresa e através da rede para auxiliar na tomada de decisões empresariais [FEL 95] [GOT 97] [RAJ 98].

As aplicações de data mining costumam descobrir de forma semi-automatizada tendências e padrões dentro de enormes grupos de dados [HEA 99]. Nos últimos anos, o uso da mineração de dados tem crescido muito no meio empresarial; no entanto, data mining costuma trabalhar somente com dados estruturados, deixando de lado uma enorme gama de informações textuais que poderiam ser extremamente úteis [GOT 97] [FEL 98] [RAJ 98] [LOH 2000].

Além disso, com o advento e imensa utilização da Internet, enormes quantidades de informações foram colocadas à disposição de qualquer pessoa que saiba encontrá-las (como e-mails, chats, home-pages, etc. [WIV 99]). Informações preciosas estão agora disponíveis em centenas de milhares de documentos sem nenhuma ou com muito pouca estruturação [GOT 97] [HEA 99] [WIV 99]. No entanto, devido à falta de estruturação destas informações, torna-se muito difícil decifrar as mesmas de forma automática [HEA 99] [LOH 2000]. Wives em [WIV 99] refere-se à essa imensa quantidade desorganizada de informação como “Sobrecarga de Informação”. Recentemente, alguns pesquisadores têm procurado desenvolver ferramentas que possibilitem a extração dessas informações de uma forma ordenada, de maneira que elas possam ser unidas e utilizadas para gerar novos conhecimentos e, assim auxiliarem na tomada de decisões e na chegada de conclusões [GOT 97].

Dixon, em [DIX 97], afirma que KDT combina técnicas de extração de informações, recuperação de informações, processamento de linguagem natural e sumarização de documentos com métodos de mineração de dados (data mining), procurando descobrir conhecimentos anteriormente desconhecidos em grandes volumes de texto.

Segundo [GOT 97], alguns dos objetivos que as ferramentas de KDT buscam atingir são:

- Extrair informações chaves de textos.
- Organizar documentos por assuntos.
- Encontrar temas predominantes em conjuntos de documentos.
- Pesquisar por documentos relevantes através do uso de queries flexíveis e poderosas.

Rajman em [RAJ 98] declara também que as técnicas de mineração de texto preocupam-se em extrair informações implicitamente contidas em coleções de documentos, estruturação baseada em similaridade e visualização de grandes conjuntos de textos. Kodratoff em [KOD 99] acrescenta ainda que KDT procura determinar as semelhanças e/ou diferenças entre os diversos documentos analisados.

Kodratoff em [KOD 99], discute as diferenças entre KDT e NLP (Natural Language Processing – Processamento da Linguagem Natural), destacando que KDT utiliza processos de indução, o que não é do interesse da NLP, além do que KDT trabalha com grandes quantidades de textos, enquanto NLP, trabalha somente com documentos individuais ou com pequenos grupos de documentos e que KDT não está interessado em melhorar as técnicas de compreensão de textos, mas sim em descobrir relações insuspeitas, contidas nos corpos dos mesmos. No entanto, [DIX 97], afirma que métodos de pré-processamento linguístico seriam úteis em fases como extração de informação e [LAR 2000] declara que mineração de texto é uma área de pesquisa que surge na intersecção de diversas áreas de pesquisa, incluindo mineração de dados, processamento da linguagem natural e recuperação de informação.

Além disso, segundo [FEL 98], os sistemas de mineração de texto padrão, normalmente, não operam sobre documentos não preparados, mas sim sobre documentos categorizados, classificados, de forma manual ou automática, com termos identificando seu conteúdo. Assim, a exploração de texto total de documentos requer pré-processamento linguístico adicional para permitir a extração automática de elementos linguísticos mais complexos do que simples palavras e para solucionar possíveis ambigüidades.

A maioria das ferramentas desenvolvidas para mineração de texto estão dedicadas à tarefa de descoberta e extração de informação ou conhecimento de documentos de texto. As peças básicas de informação em textos, tais como a linguagem do texto, nomes de companhias, datas mencionadas, são chamadas características (features), que são utilizadas para classificar os documentos em categorias [GOT 97].

Gotthard em [GOT 97] identifica alguns dos processos usados na mineração de texto para coletar informações ou extrair características (features):

- Identificação da linguagem: Este processo procura identificar a linguagem do texto utilizando pistas, tais como uma alta frequência de palavras e estatísticas de como certas seqüências de caracteres estão distribuídas através do documento.
- Extração de Nomes: Nomes são pistas valiosas para classificar documentos por assuntos. Por meio de heurística, módulos de extração de nomes localizam ocorrências de nomes em textos e determinam a que tipo de entidade o nome se refere, como pessoas, lugares, organizações, etc. Estes algoritmos normalmente trabalham com “nomes canônicos”, que tentam distinguir entidades com nomes parecidos mas com significados completamente diferentes.
- Extração de termos de domínio: Os termos de domínio que o documento contém também são utilizados para classificar documentos por assunto. Um módulo de extração de termos usa um conjunto de heurísticas simples para identificar termos técnicos compostos de diversas palavras em documentos.
- Reconhecimento de abreviaturas: Estes módulos tentam identificar as formas abreviadas e combiná-las com suas formas completas. Quando uma forma abreviada é identificada com uma forma completa, ela é adicionada ao conjunto de palavras relacionadas a um determinado “nome canônico”.

Após a extração das características e estatísticas registradas de um conjunto de documentos, este pode ser organizado por meio da utilização de tecnologias de clusterização e categorização [GOT 97]. Alguns pesquisadores afirmam que a simples categorização de documentos poderia ser considerada KDT [HEA 99].

Algumas técnicas de KDT aplicam ferramentas de KDD sobre palavras-chave designadas a textos como atributos (características) e utilizam análise estatística para descobrir regras de associação e padrões, de acordo com as distribuições e associações das palavras-chaves nos textos. Outras técnicas extraem termos dos textos para categorizá-los e para encontrar associações; estas técnicas, no entanto, têm de lidar com enganos de semântica, tais como sinônimos, polissemia (palavras com muitos significados), lemas (palavras com o mesmo radical) e quase-sinônimos. Outras abordagens ainda tentam aplicar técnicas de KDD após a aplicação de técnicas de extração de textos que transformam a informação contida nos textos em banco de dados estruturados. Em alguns casos, a simples extração de informações tem produzido resultados; no entanto, sistemas de extração de informações são muito dependentes de domínio e úteis somente em aplicações específicas [LOH 2000].

Dixon em [DIX 97] afirma que a maioria dos sistemas de KDT seguem os seguintes passos para produzir conhecimento:

- Recuperação de informações, onde são localizados e recuperados os documentos considerados relevantes.

- Extração de informações, onde se procura extrair informações dos documentos selecionados, com o objetivo de preencher os modelos especificados pelo usuário com a informação esperada.
- Mineração de informações, onde se procura descobrir padrões dentro das informações recuperadas.
- Interpretação, onde os padrões descobertos são interpretados.

Wives e Loh em [WIV 99] são mais detalhistas, definindo os passos que devem ser seguidos como:

- Definição de objetivos.
- Seleção de um subconjunto de dados.
- Limpeza dos dados.
- Redução dos dados (escolha de características relevantes para a análise).
- Escolha da técnica de mineração.
- Mineração.
- Interpretação dos resultados.
- Consolidação do conhecimento descoberto.

Wives e Loh em [WIV 99] afirmam que a descoberta é uma atividade iterativa e interativa, com aplicação repetida de métodos de mineração e interpretação dos resultados pelos usuários.

Hearst em [HEA 99] afirma que KDT não necessita compreender os textos pesquisados, sendo suficiente uma mistura de processamento computacional e análise guiada pelo usuário.

Segundo [WIV 99], um dos objetivos de KDT não é simplesmente retornar documentos contendo textos de interesse do usuário, mas tratá-los de forma a trazer algum tipo de conhecimento novo e útil. Hearst em [HEA 99] afirma que a maioria das ferramentas de KDT, são utilizadas para auxiliar na extração de conhecimentos já conhecidos (pelo menos pelo autor), de grandes volumes de informações textuais, como a WEB. Hearst em [HEA 99] acredita que KDT poderia ser utilizada para gerar realmente novos conhecimentos ainda desconhecidos de todos, através de consultas e estatísticas desses grandes volumes. KDT poderia ser utilizada para análise exploratória de dados, partindo do princípio de que novas informações podem ser derivadas de conjuntos de textos. Obviamente isso é uma das principais características da pesquisa; no entanto, o

número de textos têm crescido constantemente e é impossível para qualquer pesquisador ler mais do que uma pequena parcela dos textos publicados sobre suas áreas de pesquisa, isso sem falar nas áreas relacionadas. Hearst em [HEA 99] afirma ainda que KDT seria útil para comprovação de hipóteses, por meio da descoberta de palavras-chave relacionadas em textos.

Segundo [YEA 2000] muitas operações de mineração de texto podem ser reformuladas como problemas de inserção de identificadores (tags), principalmente em funções de recuperação (identificação) e extração de informações. Yeates em [YEA 2000] utiliza como exemplos, entre outros, o problema de identificar os limites das palavras em textos em chinês, que, diferente das línguas ocidentais, não utiliza espaços para separar as palavras; e a identificação de partes do discurso, que também é um problema de inserção de identificadores, requerendo que identificadores sejam colocados entre cada palavra para determinar seu papel sintático no contexto.

Feldman em [FEL 95] propôs o uso de paradigmas de categorização de tópicos dos textos para anotar artigos de texto com conceitos significativos, que estejam organizados em uma estrutura hierárquica, permitindo sumarização de dados, exploração de padrões interessantes e análise de tendências, combinando paradigmas de categorização e técnicas de KDD. Apesar de simples, Feldman afirma que esta técnica é robusta e fácil de implementar. Esta abordagem baseia-se em três componentes:

- Definição de uma hierarquia de conceitos;
- Categorização dos textos pelos conceitos da hierarquia;
- Comparação das distribuições dos conceitos para encontrar padrões inesperados.

Rajman e Besançon em [RAJ 98], discorrem sobre duas técnicas de mineração de textos: Extração automática de associação de palavras-chave e mineração de documentos prototípicos (originais). A primeira técnica utiliza normalmente conjuntos de palavras-chave para criar estruturas de indexação que são utilizadas como base para extração de informação. Esta técnica procura extrair associações significativas de palavras-chave e possui duas etapas básicas: geração de conjuntos de palavras-chave, chamados conjuntos freqüentes e geração de todas as regras de associação que possam ser derivadas dos conjuntos freqüentes produzidos.

Já a segunda técnica, é baseada na “abordagem de texto completo”, que aplica técnicas de descoberta de conhecimento ao conteúdo textual completo dos documentos. No entanto, os experimentos de Rajman, citado em [RAJ 98], demonstram que o processo de extração não produz bons resultados quando as técnicas padronizadas de extração de associações são aplicadas diretamente sobre as palavras contidas nos documentos ao invés de operar sobre conceitos abstratos já representados pelas palavras-chave. Rajman em [RAJ 98] sugere então a extração de documentos prototípicos (um documento correspondendo a uma informação que ocorre em um estilo repetitivo na coleção de documentos, ou seja, um

documento representando uma classe de documentos similares na base textual). Esta técnica se divide também em duas etapas básicas:

- Identificação de partes do discurso, que identifica as categorias morfo-sintáticas (substantivos, verbos, adjetivos) das palavras nos documentos, filtrando palavras não-significativas, com base em sua categoria morfo-sintática.
- Extração de termos, que procura identificar as combinações dependentes de domínio, permitindo ao processo de mineração visualizar as co-ocorrências mais significativas. Este processo passa pelas etapas de identificação de termos candidatos e filtragem dos mesmos.

Feldman em [FEL 98] declara que a maioria das abordagens de mineração de texto anteriores utilizam identificadores associados a documentos ou palavras contidas nos mesmos. Feldman em [FEL 98], utiliza “termos normalizados”, seqüências de uma ou mais formas de palavras “lematisadas”, associadas com identificadores de parte do discurso. Feldman em [FEL 98] sugere uma abordagem baseada na extração de termos significativos de documentos, iniciando com uma coleção de documentos sem rótulos ou identificadores. Os documentos são rotulados por meio de termos extraídos dos próprios documentos. Estes termos e entidades adicionais de nível mais alto, são utilizados para operações de KDD sobre os documentos. O sistema de mineração de texto baseado em termos, descrito por [FEL 98], utiliza um módulo de extração de termos, responsável pela rotulação de cada documento com um conjunto de termos extraídos do próprio documento. Este módulo passa por três estágios principais:

- Pré-processamento lingüístico, que engloba simbolização, identificação de parte do discurso (para associar categorias morfo-sintáticas, como substantivos, verbos e adjetivos) e lematisações (uma versão lingüisticamente mais fundamentada de busca do processo de desenvolvimento, da origem da palavra);
- Geração de termos, onde seqüências identificadas de lemas são selecionadas como termos candidatos potenciais baseando-se em padrões morfo-sintáticos relevantes; em seguida, tenta-se combinar os termos adjacentes e calcula-se o coeficiente de associação entre cada par, decidindo-se se eles devem ser combinados; os documentos são então atualizados pela conversão de todos os termos combinados em termos atômicos e o procedimento é repetido até que não se possa mais gerar novos termos;
- Filtragem dos termos, que, baseada em algum esquema de classificação de relevância estatística, seleciona somente os termos de maior pontuação.

Feldman em [FEL 98] destaca ainda a necessidade de uma taxonomia (classificação) de termos, para a realização de mineração de texto a nível de termos, permitindo a produção de regras de associação de alto nível, que capturam relacionamentos entre grupos de termos ao invés de termos individuais, além de permitir ao usuário especificar tarefas de mineração de uma forma concisa. Feldman em [FEL 98] descreve um

conjunto de ferramentas para construção semi-automática de uma taxonomia; sua principal ferramenta é um Editor de Taxonomia, que permite adicionar conjuntos de termos à taxonomia de termos do sistema ou adicionar taxonomias externas ao mesmo.

Loh em [LOH 2000] sugere o uso de descoberta baseada na extração de conceitos dos textos. Segundo Loh, conceitos representam atributos do mundo real, como eventos, objetos, sentimentos e ações; e podem ser utilizados para auxiliar a compreender idéias e ideologias presentes nos textos. Esta abordagem combina uma tarefa de categorização automática para identificar os conceitos presentes nos textos e uma tarefa de mineração para descobrir padrões por meio da análise e relacionamento das distribuições de conceitos em uma coleção. Uma classificação prévia é necessária para criar as definições de conceitos. Os principais objetivos desta abordagem são:

- Permitir a descoberta sobre conceitos, ao invés de palavras ou valores de atributo, procurando encontrar idéias, ideologias, tendências e intenções presentes nos textos.
- Diminuir o esforço necessário para identificar conceitos nos textos.
- Encontrar padrões interessantes em coleções textuais usando técnicas estatísticas simples.
- Permitir descobertas temporárias com objetivos mal definidos sem ter que gastar tempo e esforço criando modelos formais.

A abordagem sugerida por [LOH 2000] inclui ainda:

- Análise de Discurso, procura por intenções e estruturas em expressões textuais.
- Sociologia, procura por temas e idéias presentes em um estudo textual.
- Analogia, procura por conceitos iguais presentes em discursos diferentes.
- Pesquisa de Saúde, procura por relações entre sintomas presentes em registros textuais.
- Inteligência Competitiva, procura por estratégias usadas por diferentes companhias.

Wives e Loh em [WIV 99] detalham um grande número de técnicas de KDT, tais como:

- Descoberta tradicional após extração: Os dados são extraídos dos textos, formatados em bases de dados estruturadas e depois são examinados por algoritmos de KDD, com o objetivo de gerar conhecimento útil para o usuário.

- **Descoberta por Extração de Passagens:** Esta técnica procura por determinadas informações através da utilização de regras que busquem passagens do interesse do usuário no texto. Nesta técnica, o usuário não precisa ler todo o texto, mas deve interpretar os trechos retornados pelo algoritmo para extrair a informação desejada.
- **Descoberta por Análise Lingüística:** Esta técnica tenta descobrir informações e regras através de análises a nível léxico, morfológico, sintático e semântico dos textos.
- **Descoberta por Análise de Conteúdo:** É semelhante às duas técnicas anteriores, no entanto demonstra maior empenho no tratamento semântico dos textos e tenta compreender o significado do texto, ao invés de retornar trechos contendo passagens específicas.
- **Descoberta por Sumarização:** Utiliza técnicas semelhantes às anteriores, diferenciando-se pela tentativa de gerar sumários dos textos analisados em linguagem natural.
- **Descoberta por Associação entre Passagens:** Esta técnica combina recuperação de passagens com recuperação contextual, pois tenta recuperar automaticamente informações relacionadas no mesmo texto ou em vários, normalmente através de links. Apresenta ao usuário partes de texto que tratam de um mesmo assunto.
- **Descoberta por Listas de Conceitos-Chave:** Esta técnica apresenta uma lista com os conceitos principais de um texto. Baseia-se na idéia de que o significado de um texto não é determinado por sua leitura linear, mas pela análise do conjunto de elementos léxicos (palavras-chave) mais importantes.
- **Descoberta de Estruturas de Textos:** Esta técnica procura determinar a estrutura do texto analisado, através do relacionamento de cadeias de termos, baseando-se na afirmação de que conceitos relacionados costumam aparecer próximos uns dos outros.
- **Descoberta por Clusterização:** Esta técnica classifica elementos em classes, facilitando a identificação de padrões nestas classes e portanto sua compreensão por usuários que possuam conhecimento dentro do domínio.
- **Descoberta por Descrição de Classes de Textos:** Esta técnica tenta determinar as principais características de uma classe de documentos já agrupada.
- **Descoberta por Recuperação de Informações:** Esta é uma técnica que auxilia várias outras, encontrando documentos com novas informações, normalmente através do simples retorno de informações contendo determinados trechos de texto, com filtros ou não.

- **Descoberta por Associação entre Textos:** Esta técnica tenta relacionar diversos textos com idéias afins, unindo diversos fragmentos para gerar um conhecimento.
- **Descoberta por Associação entre Características:** Esta abordagem procura relacionar tipos de informação (atributos) presentes em textos, aplicando técnicas de correlação estatísticas ou associação tradicional em KDD diretamente sobre partes do texto. Esta técnica extrai trechos automaticamente do texto, quando da presença de palavras-chave e procura encontrar padrões por meio de análises e estatísticas da distribuição das palavras-chave.
- **Descoberta por Hipertextos:** Esta é uma técnica basicamente de recuperação de informações, utilizada principalmente por quem tem pouco conhecimento sobre um domínio e deseja adquirir um conhecimento geral sobre o mesmo.
- **Descoberta por Manipulação de Formalismos:** Os textos são recuperados e representados em formalismos internos e sofrem transformações por meio de regras que geram novas representações, que são utilizadas, por sua vez, como hipóteses para novos conhecimentos.
- **Descoberta por Combinação de Representações:** Diferencia-se da associação entre textos, porque os textos passam por um processo de representação interna antes de serem combinados.
- **Descoberta por Comparação de Modelos Mentais:** Esta técnica procura representar documentos e o estado de conhecimento do usuário (modelo mental das informações) em um formalismo padrão para depois compará-los.

Dixon, em [DIX 97] também discorre, além de outras já citadas, sobre a técnica de Regras Episódicas: Esta é uma técnica de Data Mining adaptada para KDD, que busca atribuir valores temporais aos itens de dados. Ahonen, citado em [DIX 97], descreve uma seqüência de tuplas consistindo de um vetor de características e um índice descrevendo sua localização temporal. Ahonen utiliza tuplas para representar cada ocorrência de palavra (podendo ser também frases, pontuações, etc.), no documento e sua localização. Esta técnica procura por padrões, tais como palavras ou frases co-ocorrentes que possam ser usadas na construção de listas de concordância ou regras gramaticais de aprendizagem.

Moscarola, citado em [FRE 98] distinguiu sete níveis de análise textual, implementados no software Sphinx Lexica:

- **A Abordagem do Léxico Resumido:** Afirma que ao examinar só as palavras que ocorrem mais freqüentemente, pode-se ter uma idéia de seu conteúdo. A palavra léxico refere-se à lista de todas as palavras do texto usado com o número de vezes em que cada uma ocorre.

- A abordagem de Léxico Controlado: Reduz o texto ao seu léxico (cada palavra do texto e sua frequência), e controla, através da navegação do léxico, a validade e o fundamento das interpretações elaboradas a partir desse mesmo léxico.
- A Abordagem de Léxico Seletivo: Trabalha com um léxico reduzido, depois de ter eliminado as palavras “ferramentas” (tools), palavras sem um significado útil, como preposições ou artigos, também conhecidas como stop words, concentrando-se em substantivos, verbos, adjetivos.
- Estatísticas Léxicas e Quantificação de Texto: Pesquisa as características do léxico para estabelecer estatísticas das palavras do texto de acordo com uma variável não-textual externa, como gênero (masculino / feminino) idade (acima ou abaixo de 20), nível de educação (graduado, não-graduado), etc.
- Codificação do Léxico e Geração de Medidas: Descreve o texto por uma variável nominal criada por uma codificação baseada na presença nas respostas de um grupo de palavras selecionadas do léxico.
- Quantificação do texto: Calcula o número de vezes que certas palavras ocorrem, começando do texto “lematisado” (texto com seus substantivos, verbos, adjetivos, etc. já identificados), e sem as palavras ferramentas. Este nível permite saber se certa opinião é exclusiva ao texto ou se é regular ou comum.
- A Análise Multivariada dos Dados Textuais: A análise dos dados é feita sobre as variáveis extraídas do léxico. Os métodos de análise multivariada são aplicados utilizando as novas variáveis, tais como análise fatorial ou classificação automática.

Freitas em [FRE 98] descreve os resultados de diversas análises de conteúdo utilizando as técnicas acima descritas, por exemplo, sobre controle de qualidade em uma empresa no sul do Brasil e um debate sobre a eleição presidencial americana.

Larocca em [LAR 2000] descreve uma ferramenta de mineração de texto que utiliza duas das técnicas já descritas anteriormente: clusterização de documentos (descoberta por clusterização) e sumarização de texto (descoberta por sumarização).

Larocca em [LAR 2000], afirma que um dos principais problemas na mineração de texto é que um documento pode conter um número de palavras muito grande e se cada uma destas palavras for representada como uma coordenada em um vetor, farão com que este se torne grande demais para o algoritmo. Por este motivo, Larocca afirma ser imperativo aplicar métodos de pré-processamento que reduzam de forma substancial o número de palavras a serem processadas pelo algoritmo. Larocca afirma ainda que o método de pré-processamento deve ser robusto de forma a ser capaz de lidar com textos contendo erros gramaticais e tipográficos.

O sistema de Larocca pode aplicar quatro métodos de pré-processamento nos documentos originais, descritos em seguida:

- **Case folding:** Este método consiste em converter todos os caracteres de um documento no mesmo formato, ou seja, converte todos os caracteres em maiúsculas ou minúsculas.
- **Radiciação (Stemming):** Consiste na conversão de cada palavra à sua raiz, ou seja, uma forma neutra com a respeito a delimitadores de texto (tags-of-speech) e inflexões verbais ou plurais. Para isolar a raiz de uma palavra é necessário eliminar seus sufixos; algoritmos de derivação normalmente incorporam uma grande quantidade de conhecimento lingüístico.
- **Remoção de Palavras Ferramentas (Stop Words):** Stop Words são palavras que ocorrem muito freqüentemente em um documento e por serem tão comuns, elas geram muito pouca informação sobre o conteúdo do documento no qual elas aparecem. Dessa forma, geralmente retiram-se as stop words, como artigos, conetivos, etc. da representação do documento.
- **Representação N-grama (N-gram):** Este método é uma alternativa aos métodos de derivação e remoção de stop words. Um N-grama é uma fatia de N-caracteres de uma string mais longa, segundo Cavnar citado em [LAR 2000]. Larocca utiliza como exemplo a palavra DATA, que pode ser representada pelos trigramas _DA, DAT, ATA, TA_ ou pelos quadrigramas _DAT, DATA, ATA_, nos quais o caráter sublinhado representa um espaço importante ou de segmento. A representação N-grama, possui a vantagem de ser mais robusta em relação aos métodos de derivação e remoção de stop words, sendo mais sensível a erros gramaticais e tipográficos e não requerendo preparações lingüísticas, o que torna este método mais independente da linguagem; no entanto, o método de representação N-grama não é tão bem sucedido em reduzir o número de palavras como os outros dois métodos.

O sistema de Larocca realiza o pré-processamento, primeiramente aplicando o método de case folding e em seguida escolhe entre aplicar derivação e remoção de stop words ou representação N-grama.

O algoritmo de clusterização utiliza o algoritmo Autoclass de Cheeseman citado em [LAR 2000]; a saída deste algoritmo é apresentada ao usuário junto com uma lista de palavras-chave, caracterizando cada cluster. Estas palavras-chave auxiliam o usuário a compreender rapidamente os tópicos principais discutidos nos documentos pertencentes a um determinado cluster. Estas palavras-chave podem ser vistas como um tipo de ultra-sumário dos conteúdos de todos os documentos no cluster.

O sistema de Larocca pode também sumarizar documentos individuais. Neste caso, ele executa um algoritmo de sumarização que extrai as frases mais relevantes do documento. A relevância de cada frase é determinada pelo cálculo da relevância média de

todas as palavras na frase pré-processada. O sistema de Larocca permite o uso dos dois algoritmos em uma mesma análise, de maneira que um dos métodos complementa o outro.

5 Ambiente AME-A (Ambiente Multiagente de Ensino-Aprendizagem)

Este ambiente é composto por uma sociedade de agentes, em que cada agente é responsável por parte do processo de ensino ou por parte do processo de aprendizagem. Este ambiente almeja fornecer um acompanhamento sobre o assunto desejado ao aprendiz, de maneira que seu desempenho seja o melhor possível dentro de suas capacidades [DAM 99].

Os diversos agentes desta sociedade comunicam-se por meio de mensagens e cooperam entre si com o objetivo de ensinar e/ou aprender. Os agentes podem ser tanto humanos como artificiais (software) e atuam como processos concorrentes. As regras de cooperação e compartilhamento de recursos estão inseridas na sincronização e priorização dos processos. Os agentes possuem objetivos individuais, mas compartilham o objetivo comum de auxiliar no aprendizado. O sistema é composto por nove agentes: um humano, que pode ser tanto o aprendiz como o professor, e oito programas, além de um banco de dados [DAM 99]. A fig. 5.1, abaixo, demonstra a interação entre os agentes, sendo que o agente humano é representado por uma elipse, os agentes restantes são representados por retângulos e as comunicações através de mensagens indicadas por setas simples. O agente Promove_Interação, aparece aqui em destaque, por ter servido como motivação e embasamento para o desenvolvimento da ferramenta aqui descrita. É importante notar que o agente Promove_Interação relaciona-se exclusivamente com o agente humano, não se relacionando com os outros agentes.

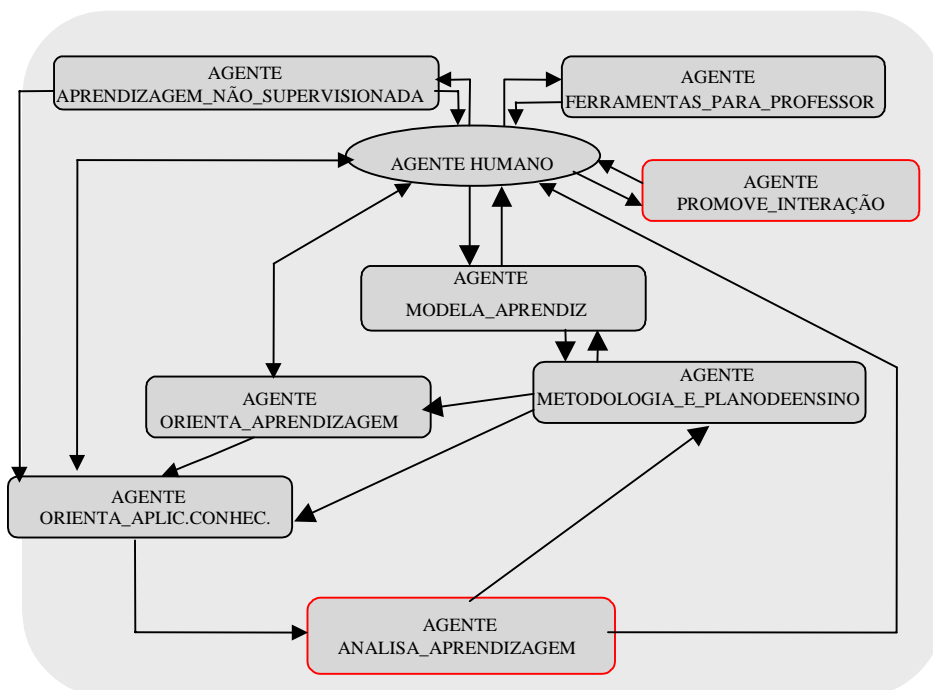


FIGURA 5.1 – Arquitetura do ambiente AME-A

No início da interação com este ambiente, o aprendiz é examinado por um agente que determina seu perfil. Baseado neste perfil, todo agente que participar do processo de aprendizagem analisa as ações do aluno e propõe mudanças nos valores que determinam as características do aprendiz [DAM 99].

Este ambiente também fornece algumas ferramentas, tais como [DAM 99]:

- O agente “PROMOVE_INTERAÇÃO”, que auxilia nas interações entre aprendizes; professores; e aprendizes e professores. A ferramenta desenvolvida neste trabalho está parcialmente baseada no conceito deste agente. Tomou-se um software de “chat”, já previamente desenvolvido e realizaram-se algumas alterações no mesmo, para representar o agente Promove_Interação, embora este software ainda não implemente todas as características do mesmo; em seguida, desenvolveu-se uma ferramenta que auxiliasse o professor a determinar o nível de desempenho dos aprendizes durante as sessões em que estes discutiam os conteúdos propostos pelo professor. Esta ferramenta seria posteriormente utilizada pelo agente Analisa_Aprendizagem para fornecer um feedback do desempenho do aprendiz quando do desenvolvimento de textos construídos em colaboração com outros aprendizes.
- O agente “FORNECE FERRAMENTAS PARA O PROFESSOR”, por intermédio do qual é possível acrescentar novos conteúdos ou atualizar conteúdos mais antigos, além de novos paradigmas de aprendizagem, novas áreas de conhecimento ou novos objetivos.

Cada agente artificial do ambiente é composto de [DAM 99]:

- Uma interface de rede.
- Uma interface de comunicação, que provê tratadores de notificações de I/O ou de exceções.
- Modelos de conhecimento sobre os outros agentes e dos métodos para utilizá-los.
- Um modelo de seus próprios conhecimentos, dependente da aplicação.
- Conhecimento a respeito dos procedimentos necessários para executar suas próprias tarefas.

Além disso, cada agente é definido por uma sêxtupla, por exemplo, um agente A possuiria uma sêxtupla tal como $(W_A, A_A, G_A, P_A, S_A, F_A)$, onde [DAM 99]:

- W_A (World ou mundo de A), é o conjunto de possíveis estados do mundo do agente A; o estado do mundo corrente é denotado por $W_{A,C}$.

- A_A (Actions ou ações de A), é o conjunto de ações que A pode realizar. Estas ações alteram o estado do mundo corrente.
- G_A (Goals ou objetivos de A), é o conjunto de objetivos ativos do agente A.
- P_A (Plans ou planos de A), é o conjunto de conjuntos de planos disponíveis e que quando executados viabilizam os objetivos do agente A.
- S_A (Selected ou selecionado), é o conjunto de planos selecionados pelo agente A.
- F_A (Functions ou funções de A), é o conjunto de funções usadas pelo agente A, para ativar objetivos, determinar o melhor plano, executar ações, por exemplo.

Cada ação de um agente é composta de [DAM 99]:

- Pré-condições, que devem ocorrer no estado do mundo corrente do agente, antes que a ação seja executada.
- Procedimentos, que são as funções que um ou mais componentes do agente (corpo, cabeça e comunicador), executa na ação.
- Efeitos, ou seja, as alterações causadas no estado do mundo corrente do agente pela execução da ação.

Um novo objetivo pode ser acrescentado aos objetivos de um agente através da função Novo_Objeto. Novos objetivos podem ser ativados pelo próprio agente, por um evento externo, pelo recebimento de uma mensagem de outro agente ou como consequência da execução de uma ação por outro agente [DAM 99].

Os planos possuem pré-condições e efeitos semelhantes aos das ações, mas os procedimentos não, porque podem ser compostos de ações e/ou outros planos. Um plano é um conjunto de eventos. Um evento é definido por uma ação instanciada, uma ordenação parcial, um agente executante, um conjunto de eventos que o precedem diretamente, um conjunto de eventos que o sucedem diretamente, o estado do evento e ainda um conjunto de restrições nos recursos necessários [DAM 99].

A sociedade de agentes do AME-A gira em torno dos interesses do Agente Humano, que pode ser tanto o professor/administrador como o aprendiz. Como professor, ele pode criar atividades de ensino e avaliar seus aprendizes através do Agente Ferramentas-para-Professor. Como aprendiz ele pode optar por uma sessão de aprendizado não dirigida ou uma sessão de aprendizagem supervisionada. A aprendizagem supervisionada modela dinamicamente o aprendiz e seleciona a metodologia e o plano de ensino mais adequado. Além disso, existe a facilidade de interação entre aprendizes, professores ou professores e aprendizes através do agente Promove-Interação [DAM 99]. Atualmente, o aprendiz pode também ser avaliado quando interagindo com o agente Promove_Interação por meio da ferramenta descrita no capítulo seguinte.

Uma sessão de aprendizagem não dirigida envolve apenas o agente Aprendizagem_não_Supervisionada e, eventualmente, o agente Orienta_AplicConhec. Nesse tipo de aprendizagem o agente não segue um caminho pré-definido. Já uma sessão de aprendizagem supervisionada envolve os agentes Modela_Aprendiz, Orienta_Aprendizagem, Metodologia_e_PlanodeEnsino, Orienta_AplicConhec e Analisa_Aprendizagem. As características psico-pedagógicas e preferências do aprendiz são analisadas e o aluno é classificado em um modelo, utilizado para definir a metodologia de ensino a ser utilizada. A análise das características e preferências do aluno é dinâmica, permitindo que diferentes metodologias possam ser aplicadas durante o processo de ensino [DAM 99].

A modelagem ocorre no primeiro contato do aprendiz com o ambiente e durante suas interações subsequentes com os agentes do AME-A. Os dados modelados do aprendiz constituem-se em características psico-pedagógicas, nível de conhecimento, preferências e histórico das interações anteriores do aluno [DAM 99].

Na primeira etapa da modelagem, é aplicado um questionário ao aprendiz, para fornecer dados ao sistema de maneira a classificá-lo corretamente. Este questionário é composto por dezoito perguntas que avaliam características como adaptabilidade, agradabilidade, competitividade, comunicabilidade, concentrabilidade, decibilidade, impessoabilidade, intuitividade, objetividade, persistência e racionalidade. A partir deste questionário, o aprendiz é identificado com um dos dezesseis tipos de personalidade derivados dos trabalhos de Jung e Myers-Briggs citados em [DAM 99].

A seleção das estratégias de ensino depende de fatores como nível de conhecimento, do domínio, da motivação, auto-confiança, estilo de aprendizagem preferido e estado emocional do aprendiz. A escolha das metodologias a serem utilizadas é feita pelo agente Metodologia_e_PlanodeEnsino através de uma rede neural. Após selecionar uma ou mais metodologias de ensino para um determinado aprendiz, o agente Metodologia_e_PlanodeEnsino transmite essa informação para o agente Orienta_Aprendizagem [DAM 99].

No decorrer da aprendizagem, um aprendiz pode mudar seu comportamento. Caso isto ocorra, o agente Modela_Aprendiz deve transmitir essa informação ao agente Metodologia_e_PlanodeEnsino de maneira que este selecione, se necessário, novas metodologias para o aprendiz [DAM 99].

O agente Orienta_Aprendizagem seleciona o material a ser ensinado e o apresenta ao aprendiz, sob diversas formas possíveis (texto, áudio, vídeo, etc.), de acordo com a metodologia utilizada, transmitindo ao agente Orienta_AplicConhec informações sobre o material fornecido ao aluno. Este agente apresentará, em seguida, um material de avaliação ao aprendiz [DAM 99].

O agente Analisa_Aprendizagem é responsável por verificar se o nível de conhecimento adquirido pelo aprendiz é adequado, ou seja, se ele está realmente aprendendo; e repassa essa informação ao agente Metodologia_e_PlanodeEnsino, que

verifica se o agente está atingindo o objetivo proposto; caso contrário, a metodologia é mudada. O agente Analisa_Aprendizagem pode também enviar mensagens de incentivo ou para sugerir novas atitudes ao aluno [DAM 99].

O agente Ferramentas-para-Professor foi projetado para auxiliar o professor e o professor-administrador. Possui como objetivos identificar o professor já cadastrado; cadastrar professores novos; atualizar dados do professor; ajudar e orientar o professor a atualizar ou criar atividades de ensino, inserir ou alterar planos de ensino e conteúdos; conduzir o professor na análise do processo de aprendizagem de um determinado aprendiz [DAM 99].

O agente Ferramentas-para-Professor pode realizar as seguintes ações [DAM 99]:

- Incluir, alterar ou excluir os dados de um professor.
- Modificar o processo de ensino, por meio da inclusão, alteração ou exclusão de uma atividade de ensino, uma área de conhecimento (ou um objetivo dentro da área), uma metodologia, um plano de ensino, o nível de conhecimento associado ao plano de ensino ou atualizando o conjunto de aprendizes que atingiram um objetivo.
- Incluir, alterar ou excluir uma sessão de professor. A exclusão de uma sessão só pode ser autorizada pelo administrador do sistema.
- Incluir, alterar ou excluir um aprendiz. Um aprendiz só pode ser excluído por meio da autorização do próprio aluno, do professor ou do administrador.
- Incluir, alterar ou excluir uma sessão de aprendiz. A exclusão de uma sessão de aprendiz já analisada só pode ser autorizada pelo administrador do sistema.
- Disponibilizar as perguntas mais freqüentes feitas ao agente Ferramentas_para_Professor.
- Fornecer ajuda ao professor.
- Repassar as ações, que não sejam de sua competência, para os outros agentes do ambiente.

O agente Promove_Interação é responsável por promover o contato entre os agentes humanos, tanto entre aprendizes, professores ou aprendizes e professores. Este contato pode ser através de agendamento de reuniões, correio eletrônico, listas de discussão, "chat" e teleconferência. Este agente deve utilizar inteligência para analisar as interações dos alunos no grupo em relação ao tema proposto e servirá como uma ferramenta de apoio à avaliação pelo professor [DAM 99].

O agente Promove_Interação pode realizar as seguintes ações [DAM 99]:

- Fornecer endereços eletrônicos para troca de mensagens.
- Alterar a agenda de reuniões aprendiz-aprendiz, incluindo, alterando ou excluindo o registro.
- Alterar a agenda de reuniões professor-aprendiz, incluindo, alterando ou excluindo o registro.
- Alterar a agenda de reuniões professor-professor, incluindo, alterando ou excluindo o registro.
- Alterar a agenda de teleconferência, incluindo, alterando ou excluindo o registro.
- Alterar a agenda de chat aprendiz-aprendiz, incluindo, alterando ou excluindo o registro.
- Disponibilizar as perguntas mais freqüentes feitas ao agente Promove_Interação.
- Auxiliar o agente humano a interagir com os demais usuários do ambiente.
- Repassar as ações, que não sejam de sua competência, para os outros agentes do ambiente.

Neste trabalho, o agente Promove_Interação foi representado por um software de “chat” que ainda não apresenta muitas das características descritas aqui, no entanto, este software permite a interação de diversos aprendizes e registra os resultados de suas discussões. O arquivo produzido pela discussão é então avaliado pela ferramenta descrita no capítulo seguinte.

O agente Aprendizagem_não_Supervisionada possibilita ao aprendiz realizar uma sessão de aprendizagem sem auxílio, sem orientação e sem um caminho pré-determinado. As ações permitidas a este agente são descritas abaixo [DAM 99]:

- Modificar o processo de ensino, por meio da inclusão, alteração ou exclusão de uma atividade de ensino, uma área de conhecimento (ou um objetivo dentro da área), uma metodologia, um plano de ensino ou o nível de conhecimento associado ao plano de ensino.
- Incluir, alterar ou excluir um aprendiz.
- Incluir ou alterar uma sessão de aprendiz.
- Enviar mensagens ao agente Orienta_AplicConhec, para avaliar a aprendizagem do aluno, caso isto tenha sido pedido pelo mesmo.
- Fornecer ajuda ao aprendiz.

- Disponibilizar as perguntas mais freqüentes feitas ao agente Aprendizagem_não_Supervisionada.
- Repassar as ações, que não sejam de sua competência, para os outros agentes do ambiente.

O agente Modela_Aprendiz mantém atualizadas as preferências e características psico-pedagógicas do aprendiz, bem como seus dados pessoais, sugestões apresentadas ao administrador, sessões em que participou, área analisada, sub-objetivos atingidos, nível de conhecimento, plano de ensino utilizado e mensagens recebidas durante uma sessão [DAM 99].

O agente Modela_Aprendiz, pode executar as seguintes ações [DAM 99]:

- Incluir, alterar ou excluir um aprendiz. Um aprendiz só pode ser excluído mediante autorização do aluno, do professor ou do administrador.
- Incluir, alterar ou excluir uma sessão de aprendiz. Uma sessão de aprendiz já analisada só pode ser excluída com autorização do administrador.
- Fornecer ajuda ao aprendiz.
- Disponibilizar as perguntas mais freqüentes feitas ao agente Modela_Aprendiz.
- Repassar as ações, que não sejam de sua competência, para os outros agentes do ambiente.

O agente Orienta_Aprendizagem estimula e acompanha o aprendiz no decorrer do aprendizado, por meio da apresentação do material selecionado de várias formas. Este agente pode realizar as seguintes ações [DAM 99]:

- Modificar o processo de ensino, por meio da inclusão, alteração ou exclusão de uma atividade de ensino, uma área de conhecimento (ou um objetivo dentro da área), uma metodologia, um plano de ensino, o nível de conhecimento associado ao plano de ensino ou atualizando o conjunto de aprendizes que atingiram um objetivo.
- Incluir, alterar ou excluir um aprendiz. A exclusão de um aprendiz só pode ser realizada por meio da autorização do administrador.
- Incluir, alterar ou excluir uma sessão de aprendiz. A exclusão de uma sessão já analisada só pode ser feita através da autorização do administrador.
- Fornecer ajuda ao aprendiz.
- Disponibilizar as perguntas mais freqüentes feitas ao agente Modela_Aprendiz.

- Repassar as ações, que não sejam de sua competência, para os outros agentes do ambiente.

O agente *Metodologia_e_PlanodeEnsino*, estabelece o plano de ensino e a metodologia aplicada ao agente no decorrer do processo de ensino-aprendizagem. A escolha da metodologia compreende a seleção da estratégia mais apropriada ao conteúdo e às características do aprendiz em questão. As estratégias que podem ser utilizadas englobam a descrição, a explanação, a exemplificação, a exercitação e os comentários. [DAM 99]. O conjunto de ações que este agente tem o poder de realizar é idêntico ao do agente anterior.

O agente *Orienta_AplicConhec*, possui diferentes estados do mundo que são constituídos pela parte do processo de ensino que faz a aplicação dos conhecimentos para validar o ensino realizado pelas informações contidas nos modelos dos aprendizes e nas suas sessões. O conjunto de ações que este agente é capaz de realizar é idêntico ao do agente anterior [DAM 99].

O agente *Analisa_Aprendizagem*, fornece um retorno ao aluno de seu desempenho no processo de aprendizagem em uma sessão. O conjunto de ações que este agente é capaz de realizar é idêntico ao do agente anterior [DAM 99].

Baseado nas metodologias ou estratégias de ensino definidas em [DAM 99] [GIR 99] e [GIR 99b], citadas neste trabalho, o agente *Metodologia_e_PlanodeEnsino*, monta uma metodologia híbrida, referenciada nas características do aprendiz e nas metodologias de ensino adequadas ao perfil do mesmo. Assim, foram definidas, para este ambiente, as seguintes táticas de ensino [DAM 99]:

- Mostrar exemplos de situações semelhantes.
- Mostrar uma mensagem com a melhor opção.
- Mostrar exemplos relacionados sem nenhuma explicação.
- Destacar os temas mais importantes.
- Mostrar o conteúdo de cada tópico.
- Avisar, por meio de mensagens, quando o aprendiz não está demonstrando um rendimento satisfatório.
- Mostrar mensagens explicando as conseqüências de suas ações.
- Mostrar ao aprendiz sucessivas questões, permitindo-lhe analisar hipóteses, descobrir contradições e realizar inferências corretas.

- Com base nos erros do aprendiz, apresentar exemplos semelhantes para compreensão da solução e determinação dos erros através de outros casos.
- Simular situações para levar o aprendiz a construir total ou parcialmente os conceitos que estão sendo ensinados.

6 A Ferramenta Desenvolvida

A ferramenta descrita neste capítulo está associada à idéia do agente *Promove_Interação*, descrito no ambiente AME-A em [DAM 99]; este agente é representado por um módulo de “chat” independente, ativado em períodos determinados pelo professor. As interações produzidas pelo módulo de “chat” são gravadas em um diretório específico. Tomou-se um software de “chat” já pronto, desenvolvido em java e realizaram-se algumas modificações, como identificação dos aprendizes e gravação das interações produzidas. A figura 6.1 apresenta a tela do módulo de “chat”.

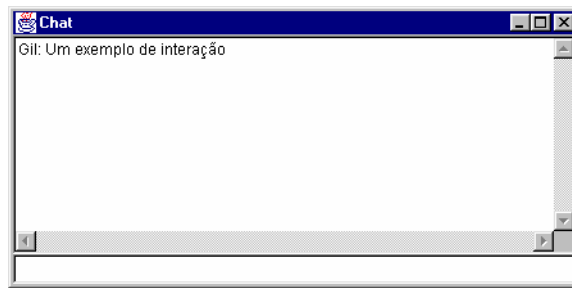


FIGURA 6.1 – Tela do módulo de Chat

Esta ferramenta tem por objetivo auxiliar o professor a determinar se seus aprendizes estão conseguindo produzir conhecimento por meio de textos construídos colaborativamente durante discussões sobre um determinado tema. A ferramenta analisa os comentários dos aprendizes, produzidos durante uma ou mais reuniões através do módulo de “chat”, realizando tarefas como classificação das palavras e frases utilizadas durante a discussão, além de prover dois módulos de avaliação baseados em conjuntos de palavras/frases-chave consideradas relevantes dentro do tema escolhido pelo professor. Os algoritmos utilizados por este software baseiam-se em algumas características de técnicas de KDT. A ferramenta em questão foi desenvolvida na linguagem de programação Visual Basic. A figura 6.2 demonstra como interagem o módulo de “chat” e a ferramenta para avaliação das interações produzidas.

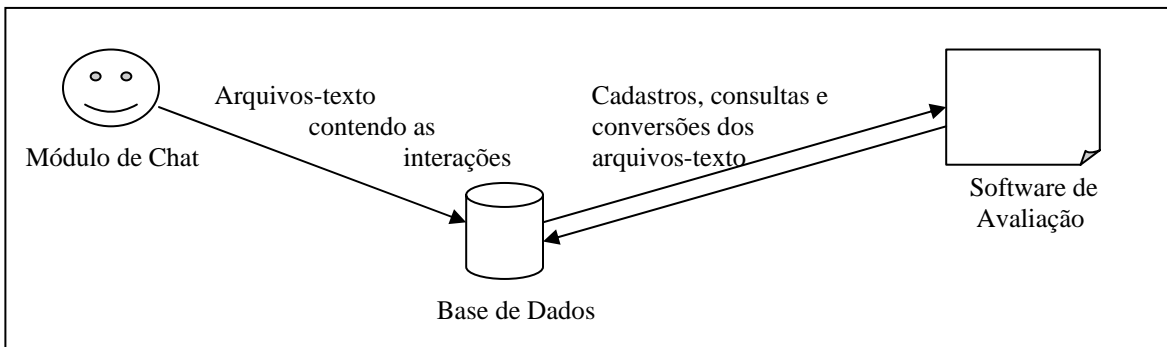


FIGURA 6.2 – Interação entre o módulo de Chat e a Ferramenta

6.1 O Modelo Entidade-Relacionamento

A figura 6.3 ilustra, por meio de um modelo entidade-relacionamento, como as tabelas que compõem o banco de dados utilizado pelo software estão relacionadas:

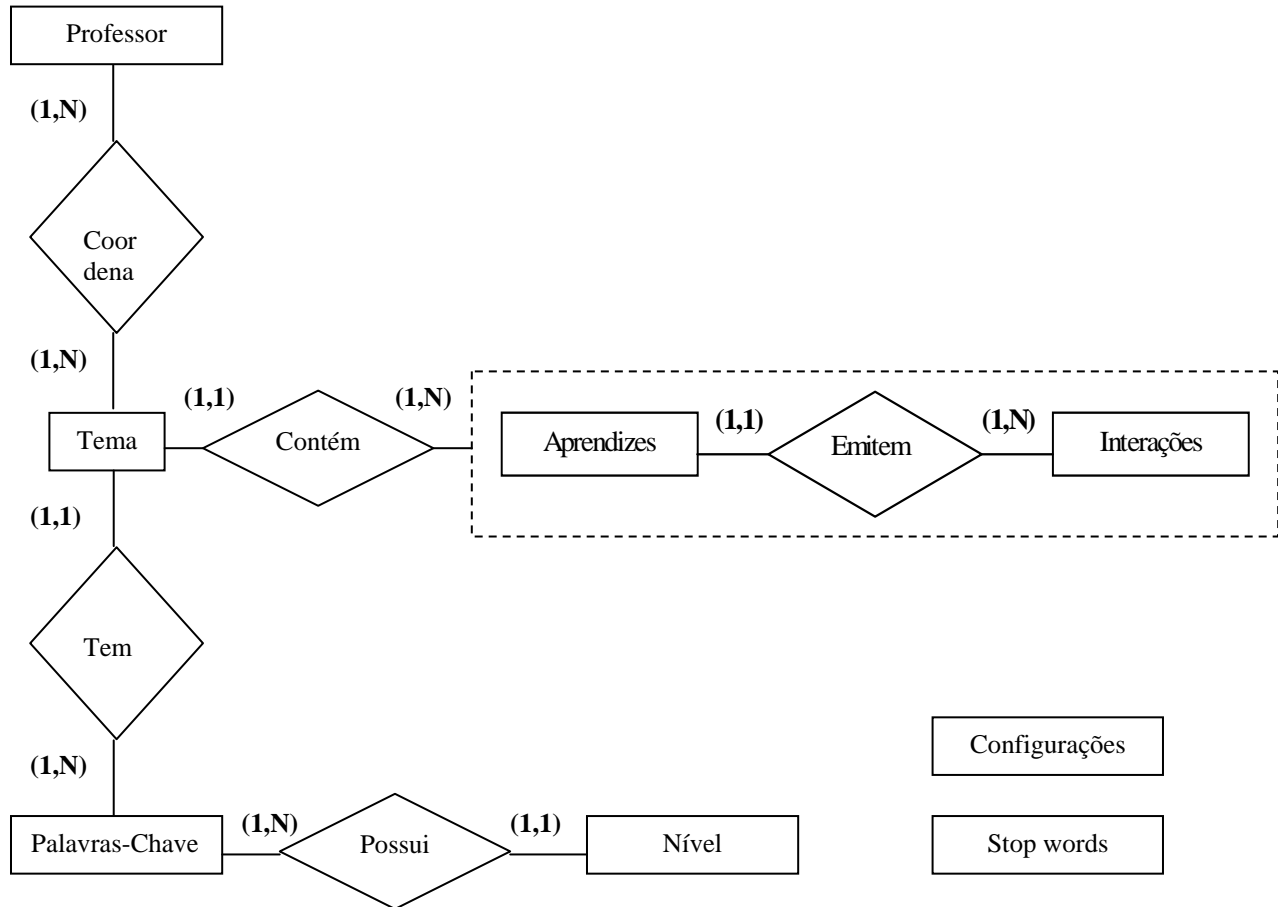


FIGURA 6.3 – Modelo Entidade-Relacionamento

6.2 Dicionário de Dados

A seguir, será descrito como estão estruturadas as tabelas do banco de dados utilizado pela ferramenta:

TABELA 1 – Tabela de Professores

Tabela Professor			
Campo	Tipo	Descrição	Chave
Logpro	Char (10)	Armazena o nome-login do professor	Primária
Nompro	Char (50)	Armazena o nome do professor	Secundária
Senpro	Char (50)	Armazena a senha do professor	

TABELA 2 – Tabela de Temas Coordenados por Professor

Tabela Temas Coordenados por Professor			
Campo	Tipo	Descrição	Chave
Logpro	Char (10)	Armazena o nome-login do professor	Primária
Codtem	Integer	Armazena o código do tema	Primária

TABELA 3 – Tabela de Temas

Tabela Tema			
Campo	Tipo	Descrição	Chave
Codtem	Integer	Armazena o código do tema	Primária
Destem	Char (50)	Armazena a descrição do tema	Secundária

TABELA 4 – Tabela de Palavras-Chave

Tabela Palavras-Chave			
Campo	Tipo	Descrição	Chave
Codcha	Long	Armazena o código da palavra-chave	Primária
Codtem	Integer	Armazena o código do tema ao qual a palavra-chave se refere	Estrangeira
Codniv	Integer	Armazena o código da categoria da palavra-chave	Estrangeira
Palcha	Char (250)	Armazena a string da palavra-chave	Secundária

TABELA 5 – Tabela de Níveis

Tabela Nível			
Campo	Tipo	Descrição	Chave
Codniv	Integer	Armazena o código do nível	Primária
Desniv	Char (20)	Armazena a descrição do nível	Secundária
Valniv	Double	Armazena o valor relativo à nível	

TABELA 6 – Tabela de Stop Words

Tabela Stop words			
Campo	Tipo	Descrição	Chave
Codbas	Long	Armazena o código da palavra-básica	Primária
Desbas	Char (50)	Armazena a string da palavra-básica	Secundária

TABELA 7 – Tabela de Aprendizes

Tabela Aprendizes			
Campo	Tipo	Descrição	Chave
Codusu	Integer	Armazena o código do aprendiz	Primária
Codtem	Integer	Armazena o código relativo ao tema que o aprendiz está discutindo	Primária
Nomusu	Char (50)	Armazena o nome do aprendiz	Secundária

TABELA 8 – Tabela de Interações

Tabela Interações			
Campo	Tipo	Descrição	Chave
Codtem	Integer	Armazena o código do tema relativo a interação	Primária
Nrint	Long	Armazena o número da interação	Primária
Codusu	Integer	Armazena o código do aprendiz	Primária
Interacao	Memo	Armazena a descrição da interação	

TABELA 9 – Tabela de Configurações

Tabela Configurações			
Campo	Tipo	Descrição	Chave
Maxpal	Integer	Armazena o número máximo de palavras que podem compor uma frase	
Letrai	Integer	Armazena o número máximo de letras que podem ser retiradas de uma palavra	
Tamrai	Integer	Armazena o tamanho mínimo que uma palavra deve possuir	
Ultrdat	Char (10)	Armazena o último arquivo analisado automaticamente	

6.3 O Funcionamento da Ferramenta

Quando ativada, a ferramenta pede que o professor se identifique, informando seu nome-login e senha. A tela de abertura da ferramenta pode ser vista a seguir, na figura 6.4.

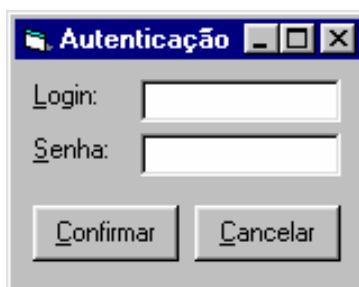


FIGURA 6.4 – Tela de Abertura da Ferramenta

Nesta tela o professor deve informar seu nome-login e sua senha e, se estes estiverem corretos, terá acesso à tela principal do software, conforme pode ser vista na figura 6.5.

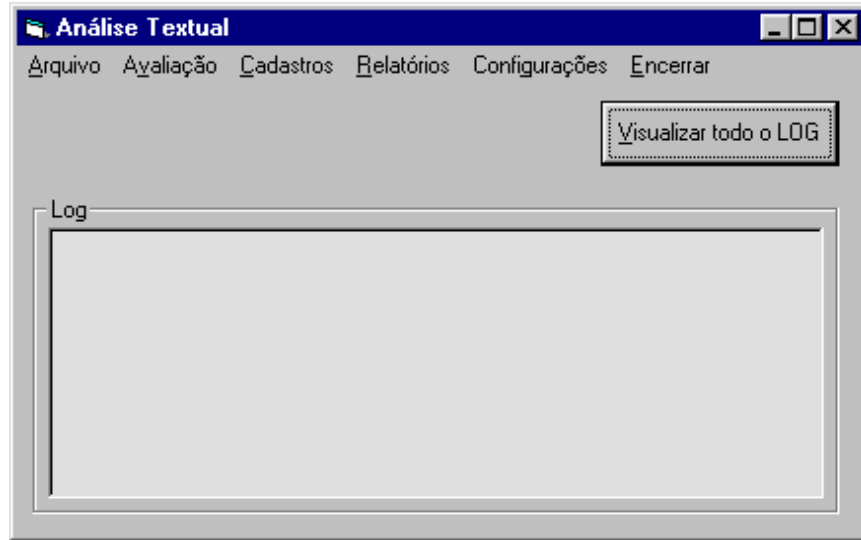


FIGURA 6.5 – Tela Principal da Ferramenta

A opção Arquivo fornece as opções de seleção de arquivos de chat a serem analisados, classificação dos aprendizes participantes da discussão com seu respectivo número de interações e classificação das palavras/frases utilizadas durante a discussão.

A opção Avaliação apresenta duas alternativas de avaliação ao professor, avaliação padrão e por queries.

A opção Cadastros permite o cadastramento de níveis, stop words, substantivos e temas com suas respectivas palavras-chave.

A opção Relatórios emite listagens das palavras-chave cadastradas, classificadas pelos temas aos quais se referem; bem como das stop words, substantivos e dos níveis.

A opção Configurações permite ao usuário configurar algumas características utilizadas durante os módulos de avaliação e classificação de palavras.

O botão “Visualizar Todo o Log”, contido na tela principal, apresenta na tela todos os arquivos de chat já analisados automaticamente, discriminando a data e o nome do arquivo analisado. Conforme já foi dito, esta ferramenta está associada a um módulo de chat, ativado em períodos determinados pelo professor e representa o agente Promove_Interação descrito no Ambiente AME-A em [DAM 99]. Este módulo de chat grava automaticamente todas as interações produzidas durante a discussão virtual em um diretório específico, no formato “diamesano.txt”. A ferramenta de análise textual verifica este diretório de tempos em tempos e, caso encontre um arquivo ainda não analisado, classifica seus usuários e interações automaticamente, emitindo no vídeo as operações que foram realizadas, tais como:

- 12/08/2001 – Novo arquivo encontrado – 12aug2001.txt

- 12/08/2001 – Iniciando identificação dos usuários/interações
- 12/08/2001 – Usuários/interações identificados

Estas mensagens permanecem na tela enquanto o programa não for encerrado, assim, no momento em que algum professor for trabalhar com a ferramenta, saberá que um novo arquivo de chat foi classificado automaticamente e já pode ser avaliado. Caso o professor queira verificar todos os arquivos já analisados automaticamente, basta pressionar o botão “Visualizar Todo o Log”.

6.4 A Opção Arquivo

Nesta opção o professor pode selecionar manualmente o arquivo de chat que ele deseja analisar. Após isso, o professor pode escolher classificar os aprendizes que participaram do chat ou classificar as palavras utilizadas durante a discussão. A figura 6.6 apresenta a tela com as alternativas da opção Arquivo:

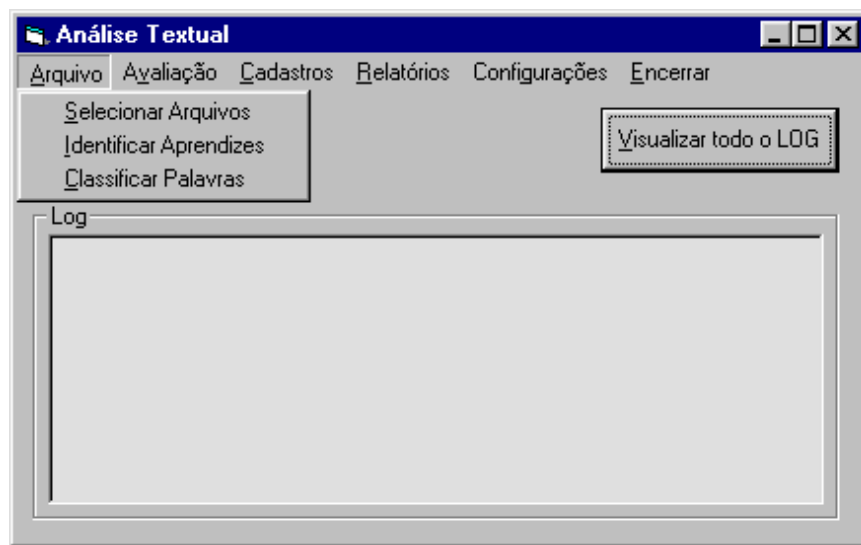


FIGURA 6.6 – Alternativas da Opção Arquivo

A alternativa Selecionar Arquivos permite que o professor selecione manualmente um arquivo de chat que ele queira analisar, embora o software faça isso automaticamente, esta opção também está disponível. Ao selecionar esta alternativa surgirá a tela para seleção de arquivo, conforme mostra a figura 6.7 a seguir:



FIGURA 6.7 – Tela de Seleção de Arquivo

Após selecionar o arquivo desejado, o professor pode escolher identificar os aprendizes que participaram da discussão ou classificar as palavras utilizadas durante a mesma. Caso queira identificar os aprendizes que participaram da discussão, será apresentada ao professor uma tela semelhante à apresentada na figura 6.8, abaixo:

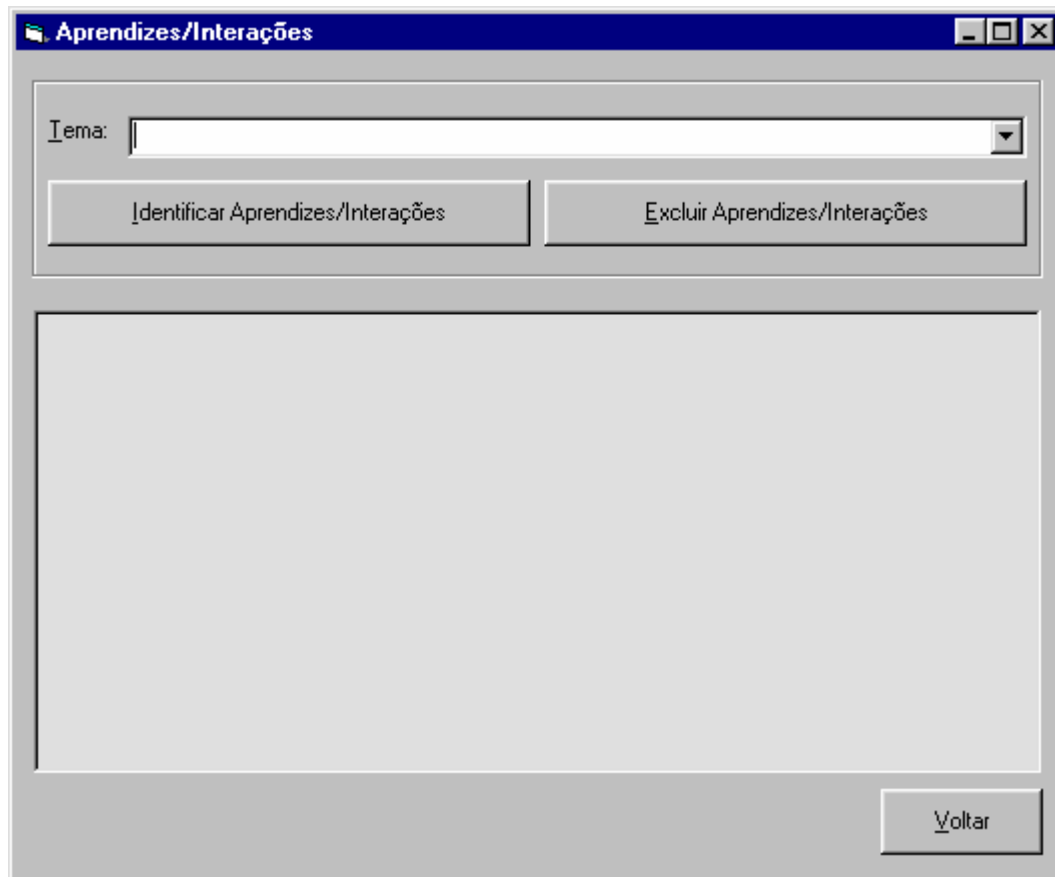


FIGURA 6.8 – Tela de Identificação dos Aprendizes

Nesta tela, o professor deve selecionar o tema que foi discutido no chat; somente serão apresentados os temas relacionados ao professor identificado durante a autenticação do sistema. Em seguida, ele pode escolher entre identificar os aprendizes que participaram da discussão ou excluir aprendizes porventura já identificados anteriormente. Nesta etapa os aprendizes, bem como suas interações são identificados e gravados nas tabelas “usuarios” e “interac”, utilizadas posteriormente durante a etapa de classificação e nas etapas de avaliação. Este módulo apresenta ao professor os nomes de todos os aprendizes que participaram da reunião, bem como o número de interações produzidas por cada um deles. A etapa de identificação de aprendizes/interações também “limpa” as interações dos aprendizes, procurando excluir caracteres que atrapalhem a avaliação das interações dos aprendizes, como pontos de exclamação, interrogação, etc. A etapa de limpeza dos dados é descrita em [WIV 99] como parte do processo de descoberta, além disso, [FEL 98] afirma que a maioria dos sistemas de text mining operam sobre documentos previamente categorizados e classificados e, segundo [LOH 2000], diversas técnicas de KDT, primeiro extraem as informações dos seus repositórios textuais e em seguida as gravam em depósitos de dados estruturados, para depois aplicarem técnicas de KDD sobre os mesmos. Isso também está de acordo com a Descoberta Tradicional após Extração descrita em [WIV 99], onde os dados não-estruturados são primeiro convertidos em bases estruturadas para depois serem minerados por técnicas convencionais. Um exemplo de identificação de aprendizes/interações pode ser visto na fig. 6.9, a seguir.

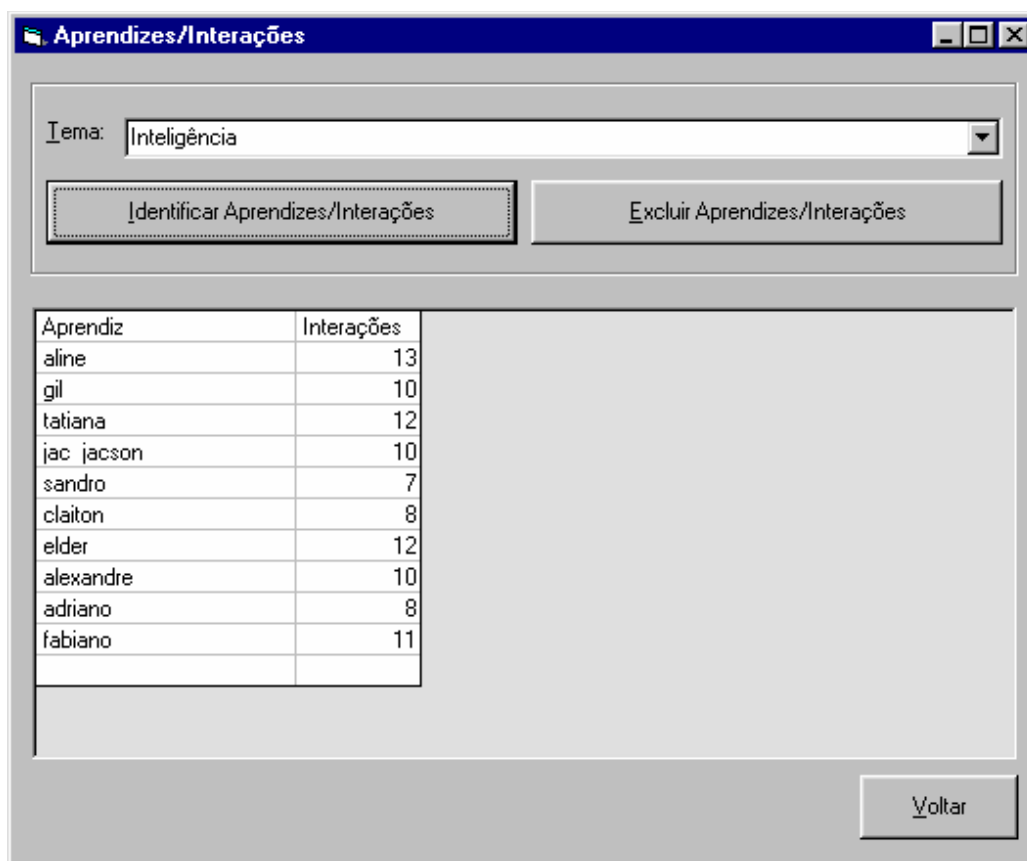


FIGURA 6.9 – Identificação de Aprendizes/Interações

Dentro da opção Arquivo, o professor também pode selecionar a alternativa Classificar Palavras. Esta alternativa já pode ser identificada como uma forma de avaliação. Ela permite ao professor observar as palavras utilizadas por um aprendiz específico ou por todos os aprendizes, além de poder fazer esta visualização dividida por cada interação individual ou por todas as interações em conjunto do aprendiz. O professor pode também escolher visualizar somente palavras-chave utilizadas durante a discussão, bem como somente stop words ou desconhecidas. Esta operação está de acordo com [GOT 97] que afirma que um dos objetivos das ferramentas de KDT é encontrar temas predominantes em conjuntos de documentos, além do que pode auxiliar o professor na etapa de Identificação da linguagem, por meio das palavras mais frequentemente utilizadas e nas etapas de Extração de nomes e Extração de termos de domínio, também descritas em [GOT 97]. Os termos mais comumente utilizados podem ser usados como características [GOT 97] [LOH 2000], para dar uma idéia sobre o que os aprendizes estão falando. Este módulo está de acordo com a técnica de Descoberta por Listas de Conceitos-Chave, descrita em [WIV 99], que apresenta uma lista dos principais conceitos do texto, afirmando que o significado do texto pode ser determinado pela análise dos elementos léxicos mais frequentes. Feldman em [FEL 98] destaca também a necessidade de uma taxonomia de termos, para realização de mineração de textos a este nível. [LOH 2000] também afirma que uma classificação prévia é necessária para criar definições de conceitos. Além disso, este módulo também identifica os substantivos utilizados durante a discussão, essa característica é semelhante a

um dos algoritmos descritos no trabalho de [JAQ 99]. Um exemplo da tela deste módulo pode ser visto na figura 6.10 a seguir.

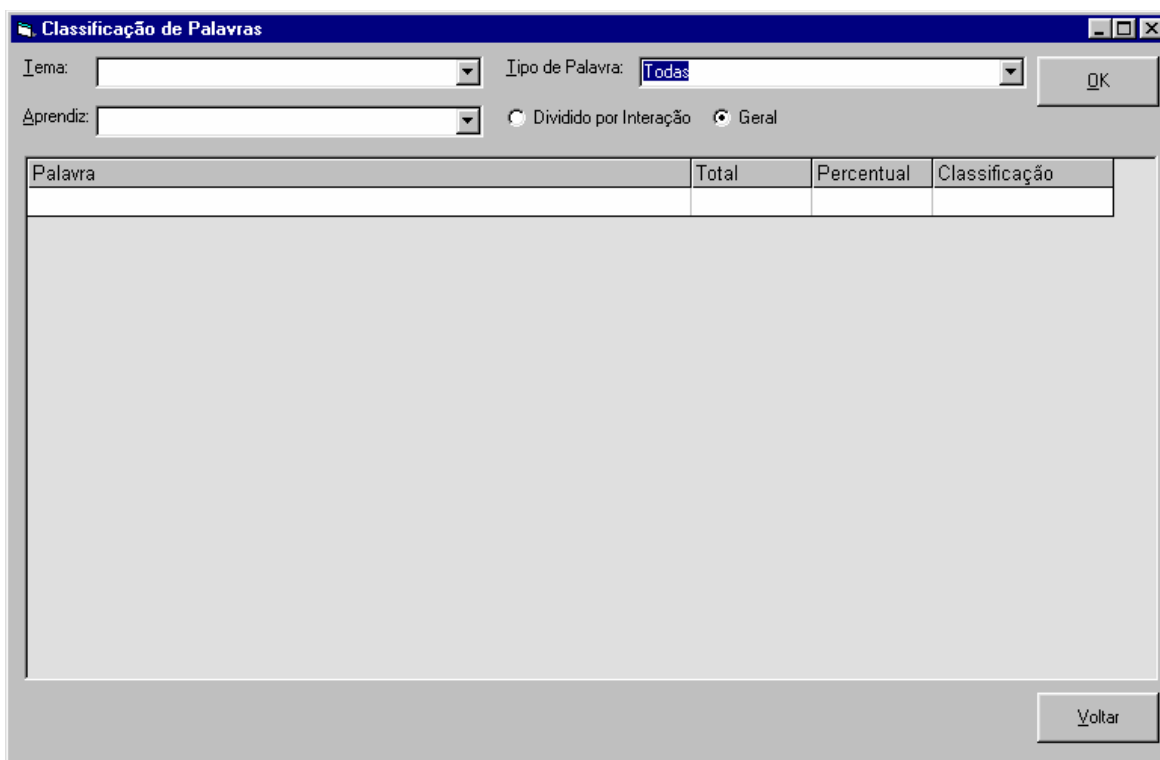


FIGURA 6.10 – Classificação de Palavras

Neste módulo, o professor deve selecionar o tema discutido durante o chat (necessário para identificar as palavras-chave). Em seguida o professor deve escolher se deseja classificar as palavras contidas nas interações de todos os aprendizes ou de um aprendiz específico. O professor deve também determinar se quer identificar todas as palavras contidas no chat ou somente palavras-chave, stop words, substantivos ou palavras desconhecidas. Finalmente, o professor pode definir se deseja uma classificação dividida por interação ou geral, que classificará as palavras contidas em todas as interações do aprendiz. Este módulo classifica todas as palavras contidas no chat: elas são apresentadas por ordem de maior utilização. O módulo informa o número de vezes que a palavra foi utilizada, seu percentual em relação ao total de palavras utilizadas pelo aprendiz e sua classificação (palavra-chave, stop word, substantivo ou palavra desconhecida). Isto está de acordo com a abordagem de Léxico Resumido descrito por Moscarola citado em [FRE 98], que afirma que se pode ter uma idéia do conteúdo de um texto identificando as palavras que ocorrem mais frequentemente. Este módulo também apresenta características da abordagem de Léxico Seletivo, descrita por Moscarola, já que permite ao professor identificar somente um tipo de palavra (palavras-chave, stop words, substantivos ou palavras desconhecidas). É preciso destacar que o termo “palavra-chave” não se refere unicamente a palavras individuais, podendo se referir a um conjunto de palavras. Um exemplo de Classificação pode ser observado na próxima figura.

Palavra	Total	Percentual	Classificação
com o	2	0,296%	Desconhecida
mas	2	0,296%	Stop Word
no	2	0,296%	Stop Word
sucesso	2	0,296%	Substantivo
pessoas	2	0,296%	Desconhecida
conhecimento	2	0,296%	Desconhecida
técnico	2	0,296%	Desconhecida
entendimento	2	0,296%	Palavra-Chave
conhecimento técnico	2	0,296%	Palavra-Chave
muito	2	0,296%	Desconhecida
os seus	2	0,296%	Desconhecida
dom	2	0,296%	Desconhecida
até	2	0,296%	Stop Word
pelo	2	0,296%	Stop Word
determinada	2	0,296%	Desconhecida
o dom	2	0,296%	Desconhecida
conhecimentos	2	0,296%	Desconhecida
técnicos	2	0,296%	Desconhecida
teste	2	0,296%	Palavra-Chave
somente as	2	0,296%	Desconhecida

FIGURA 6.11 – Classificação de Palavras

Ao final da listagem são apresentados alguns totalizadores, como pode ser observado na figura 6.12, informando o total de palavras encontradas no texto, o total de palavras-chave, substantivos, stop words e desconhecidas. Palavras desconhecidas são aquelas que não foram identificadas como palavras-chave ou stop words, sendo importante destacá-las, porque elas podem constituir palavras-chave ainda não identificadas pelo professor, sinônimos de alguma palavra-chave já classificada ou palavras-chave mal digitadas. Pode-se observar através da figura seguinte que a pesquisa retornou frases com até quatro palavras; o algoritmo de classificação pesquisa também por frases e o número máximo de palavras que as compõem é definido no módulo de configurações. Deve-se notar que as frases são montadas primeiramente em grupos de dois, em seguida em grupos de três até atingir o máximo permitido. Após montar a primeira frase com um determinado número de palavras, a segunda frase será montada com todas as palavras que compunham a primeira, exceto a primeira palavra, e acrescida da próxima palavra a ser lida, e assim sucessivamente.

Palavra	Total	Percentual	Classificação
são mais pelas	1	0,095%	Desconhecida
mais pelas definições	1	0,095%	Desconhecida
definições dos autores	1	0,095%	Desconhecida
dos autores do	1	0,095%	Desconhecida
do que realmente	1	0,095%	Desconhecida
que realmente é	1	0,095%	Desconhecida
é levado em	1	0,095%	Desconhecida
levado em conta	1	0,095%	Desconhecida
é o que digonum	1	0,095%	Desconhecida
o que digonum entendimento	1	0,095%	Desconhecida
entendimento mais geral as	1	0,095%	Desconhecida
mais geral as diferenças	1	0,095%	Desconhecida
diferenças são mais pelas	1	0,095%	Desconhecida
são mais pelas definições	1	0,095%	Desconhecida
definições dos autores do	1	0,095%	Desconhecida
dos autores do que	1	0,095%	Desconhecida
que realmente é levado	1	0,095%	Desconhecida
realmente é levado em	1	0,095%	Desconhecida
Totais: 879	Chaves: 14	Stop Words: 36	Desconhecidas: 826 Substantivos: 3

FIGURA 6.12 – Totalizadores do Módulo de Classificação

Através desta listagem, o professor pode modificar a classificação de qualquer palavra ou frase encontrada, bastando para isso um clique com o mouse sobre a palavra/frase escolhida, o que acarretará o surgimento de uma nova tela que permitirá a modificação da classificação da palavra ou frase em questão. No momento em que o professor alterar a classificação de qualquer palavra/frase, esta será excluída da tabela em que se encontrava anteriormente (exceto se sua classificação for Desconhecida) e adicionada na tabela escolhida. Neste caso, selecionou-se a frase “fator genético”, que havia, inicialmente, sido classificada como desconhecida, como pode ser observado na figura 6.13:

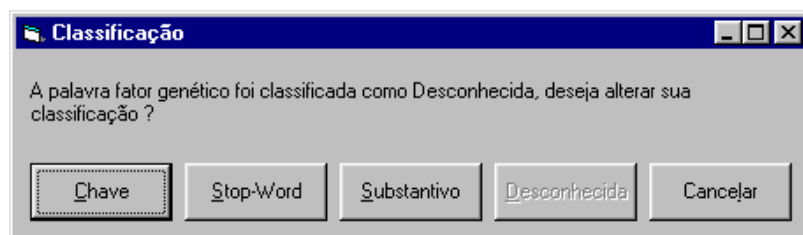


FIGURA 6.13 – Alteração da Classificação de uma Palavra ou Frase

Se o professor, resolver mudar a classificação da palavra para Chave, como neste caso, deverá informar ainda o tema ao qual se refere a palavra ou frase em questão, bem como seu respectivo nível, como pode ser visto na próxima figura:

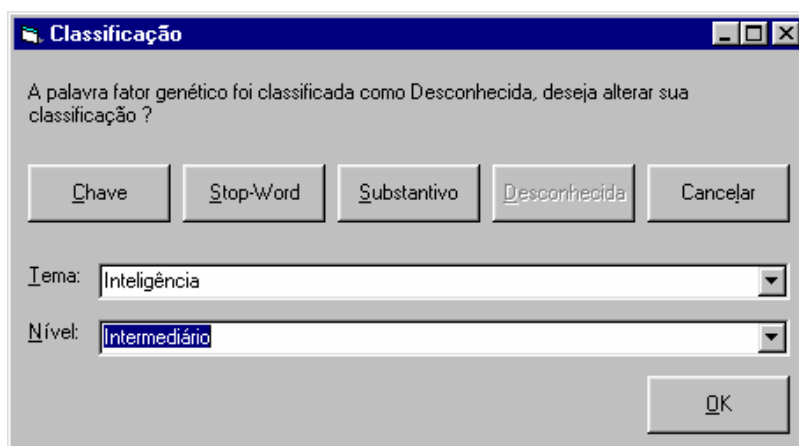


FIGURA 6.14 – Alteração de uma Palavra ou Frase Desconhecida para Chave

Após isso bastará ao professor selecionar o botão OK e a palavra escolhida será imediatamente adicionada a tabela de palavras/frases-chave.

Caso o professor escolha realizar uma classificação dividida por interação, o módulo de classificação apresentará um totalizador sempre que terminar de classificar uma interação, como se pode observar na tela seguinte:

Palavra	Total	Percentual	Classificação
inteligência seria a capacidade	1	1,563%	Desconhecida
seria a capacidade lógica	1	1,563%	Desconhecida
lógica de cada um	1	1,563%	Desconhecida
de cada um sendo	1	1,563%	Desconhecida
sendo diferenciado em níveis	1	1,563%	Desconhecida
diferenciado em níveis aliado	1	1,563%	Desconhecida
aliado as qualidades de	1	1,563%	Desconhecida
as qualidades de criatividade	1	1,563%	Desconhecida
criatividade intuição e outras	1	1,563%	Desconhecida
intuição e outras virtudes	1	1,563%	Desconhecida
alexandre - Totais: 63	Chaves: 2	Básicas: 10	Desconhecidas: 51
não	1	2,564%	Palavra-Básica
há	1	2,564%	Palavra-Básica
diversos	1	2,564%	Desconhecida
tipos	1	2,564%	Palavra-Básica
criatividade	1	2,564%	Palavra-Chave
como	1	2,564%	Palavra-Básica

FIGURA 6.15 – Classificação Dividida por Interação

Como se pode observar na tela anterior, o professor pode obter uma visualização gráfica do resultado da classificação por meio da seleção do botão gráficos. A figura 6.16

mostra um gráfico comparativo das palavras/frases-chave mais utilizadas em uma classificação:

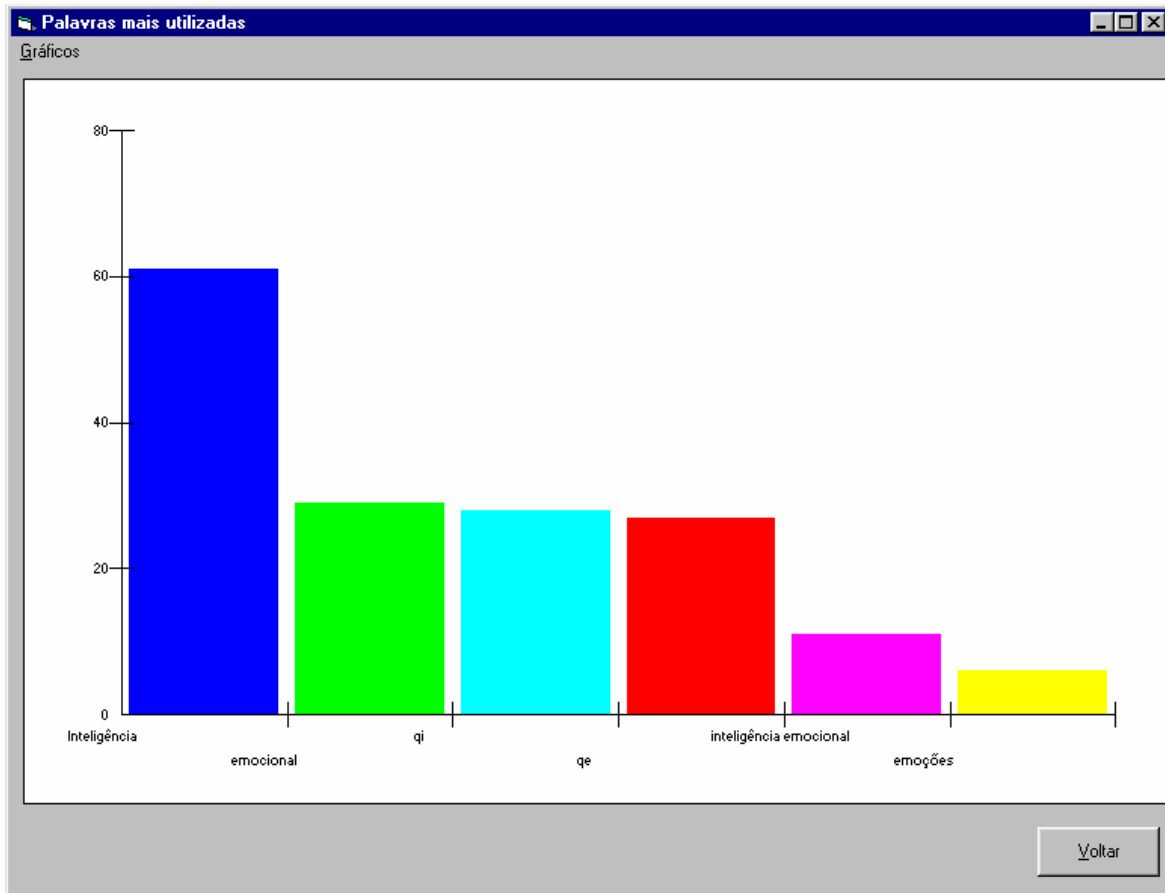


FIGURA 6.16 – Gráfico de Palavras/Frases-Chave Mais Utilizadas

O menu suspenso gráficos ainda permite ao professor escolher entre quatro tipos de gráfico: barras 2D, barras 3D, pizza 2D e pizza 3D.

6.5 A Opção Avaliação

A opção Avaliação apresenta duas alternativas de avaliação ao professor: avaliação padrão e avaliação por queries. A figura 6.17 apresenta as alternativas da Opção Avaliação.

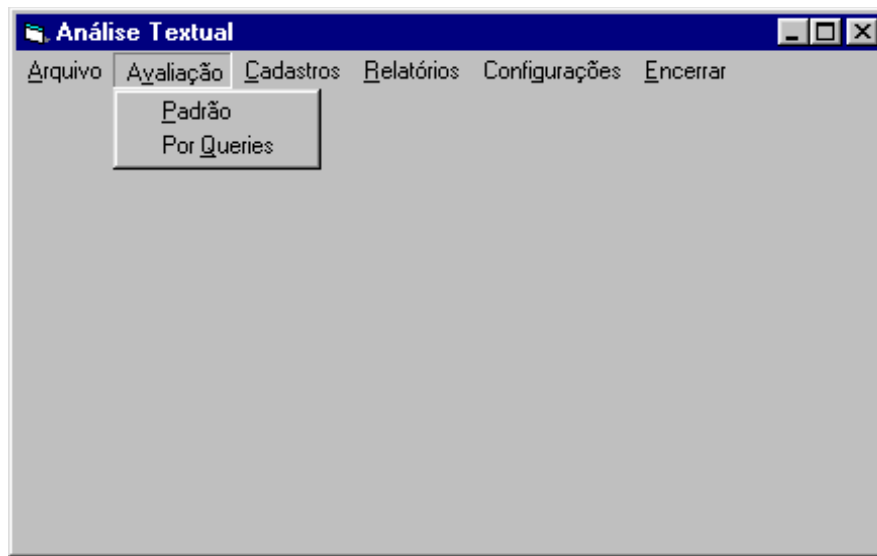


FIGURA 6.17 – Alternativas da Opção Avaliação

6.5.1 Avaliação Padrão

Caso o professor queira realizar uma avaliação padrão, será apresentada uma tela semelhante à mostrada na fig. 6.18:

The screenshot shows a window titled "Avaliação" with the following elements:

- Input Fields:**
 - Tema: [Dropdown menu]
 - Percentual Mínimo de Interações Válidas por Aluno: [Text box]
 - Aprendiz: [Dropdown menu]
 - Mínimo de Palavras-chave por interação: [Text box]
 - Nível: [Dropdown menu]
 - Percentual Máximo de palavras desconhecidas por interação: [Text box]
- Checkboxes:**
 - Levar em conta a raiz das palavras
 - Levar em conta o valor do nível
- Buttons (Right Side):**
 - OK
 - Gráficos
 - Limpar
 - Voltar
- Tabs (Bottom):**
 - Interações Válidas (Active)
 - Utilização de Palavras/Frases-Chave
 - Palavras Desconhecidas mais Utilizadas

FIGURA 6.18 – Tela de Avaliação Padrão

Como se pode observar na figura 6.18, neste módulo de avaliação o professor deve informar:

- O tema ao qual se refere o arquivo de chat que vai ser avaliado;
- Se deseja avaliar um único aprendiz ou todos;
- Se deseja pesquisar por palavras/frase-chaves de um nível específico ou de todos;
- Qual o percentual mínimo de interações válidas por aluno em relação ao total de interações geradas pelo mesmo (interações válidas são as interações que contêm um número mínimo de palavras/frases-chave, especificado no campo seguinte da tela e que não ultrapassem o máximo de palavras desconhecidas contidas na interação);
- O número mínimo de palavras/frases-chave por interação e o percentual máximo de palavras desconhecidas na interação.

Além disso, o professor pode ainda determinar se deseja levar em conta a raiz das palavras analisadas (o número máximo de letras a ser retirado do final ou do início da palavra é determinado no módulo de configurações; esta opção é útil no caso do aluno utilizar palavras/frases consideradas chave, mas mal escritas ou em tempos de conjugação diferentes; no entanto, esta opção pode incidir no erro de lemas, ou seja, palavras com significados diferentes mas com raízes idênticas [LOH 2000]), e se deseja mostrar uma pontuação relacionada aos níveis das palavras encontradas (o valor da pontuação deve ser informado pelo professor). Este módulo de avaliação retorna todas as interações consideradas válidas organizadas alfabeticamente por aluno, trazendo o percentual de interações válidas em relação ao total de interações que o aluno gerou, e, se o professor escolheu esta opção, a pontuação total do aluno. Este módulo retorna também a utilização das palavras/frases-chave pelos alunos, em ordem alfabética, e as palavras desconhecidas mais utilizadas. As figuras 6.19, 6.20 e 6.21 ilustram o desempenho deste módulo.

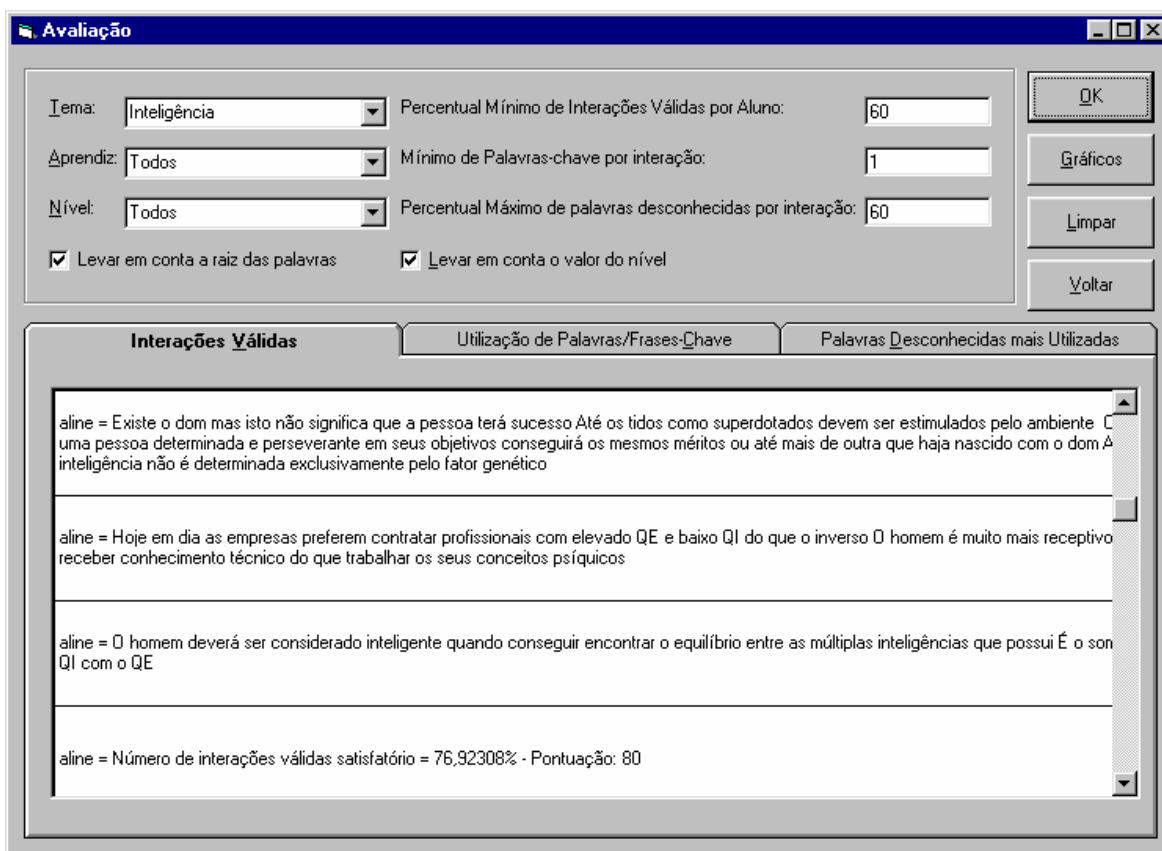


FIGURA 6.19 – Avaliação Padrão de um Arquivo de Chat

A tela anterior apresenta um trecho contendo algumas interações consideradas válidas da aprendiz Aline, apresentando no final o percentual de interações válidas em relação ao total de interações e sua pontuação; deve-se destacar que esta pontuação é definida por meio das palavras/frases-chave encontradas e o seu respectivo peso, definido pelo professor no momento de seu cadastramento. Este método de avaliação está de acordo com a técnica de Descoberta por Extração de Passagens, descrito em [WIV 99], que tenta

descobrir informações através da utilização de regras que procurem por passagens do interesse do usuário no texto; as passagens no caso são representadas pelas palavras-chave cadastradas pelo professor (o usuário). Segundo [WIV 99] nesta técnica o usuário não precisa ler todo o texto, mas deve interpretar os trechos retornados. Este método de avaliação também está de acordo com a técnica de Descoberta por Recuperação de Informações, descrita em [WIV 99], que retorna informações contendo determinados trechos do texto.

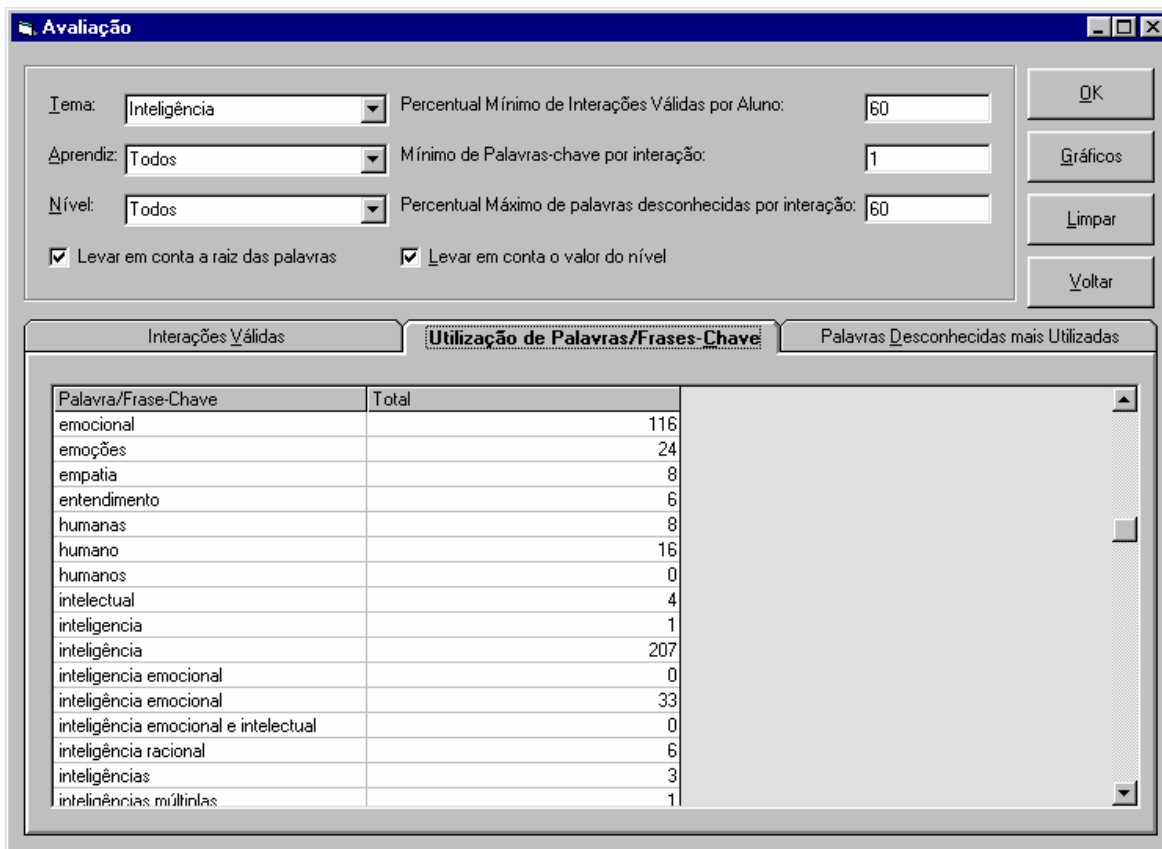


FIGURA 6.20 – Utilização das Palavras/Frases-Chave

Esta tela ilustra a utilização das palavras/frases-chave utilizadas pelos alunos; pode-se notar que algumas são muito utilizadas, enquanto outras simplesmente não foram citadas. Este tópico também está de acordo com a técnica de Descoberta por Listas de Conceitos-Chave, descrita em [WIV 99], que afirma que o significado do texto pode ser determinado através dos termos mais freqüentemente utilizados.

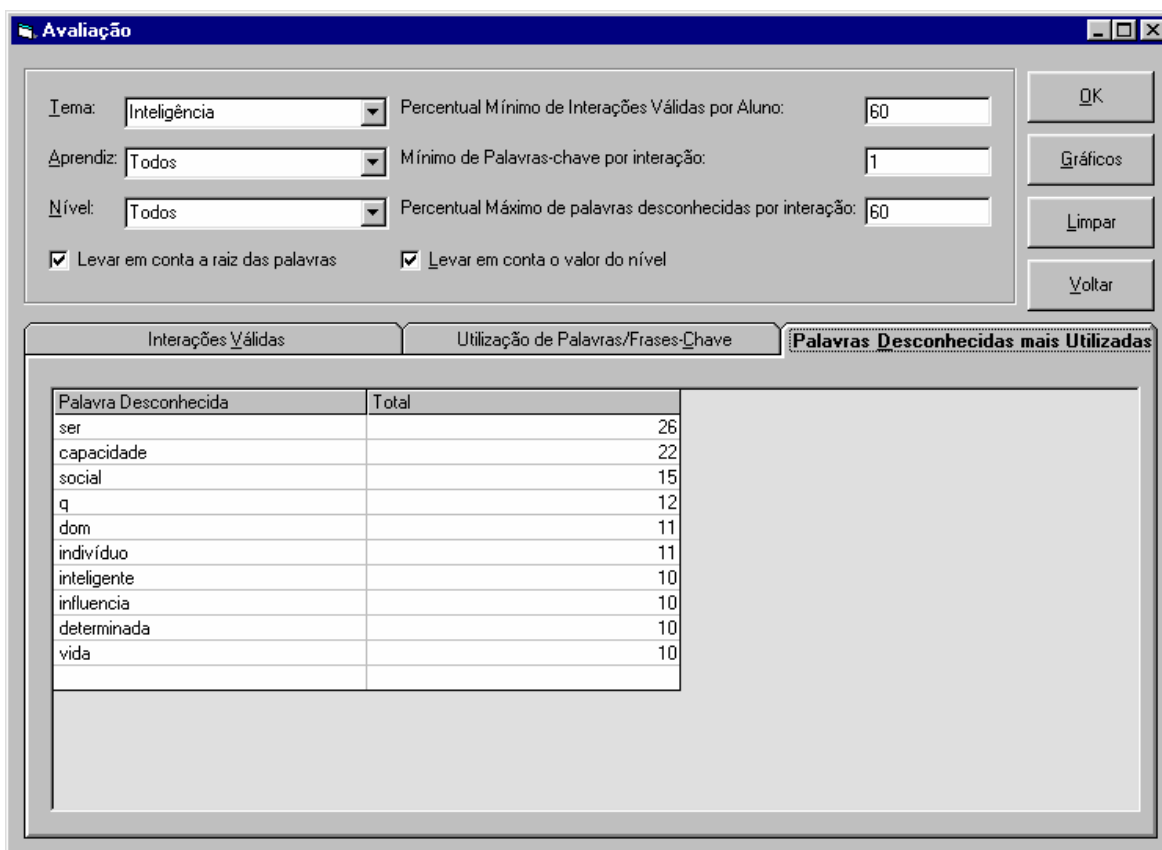


FIGURA 6.21 – Palavras Desconhecidas Mais Utilizadas

Esta última figura demonstra que o módulo de avaliação em questão também retorna os termos desconhecidos mais citados, com o intuito de destacar alguma palavra-chave ainda não identificada, porém relevante para a avaliação. Além disso, este tópico também apresenta características da técnica de Descoberta por Listas de Conceitos-Chave, descrita em [WIV 99].

Este módulo de avaliação também permite uma visualização gráfica, por meio da seleção do botão gráficos. A figura 6.22 apresenta um gráfico comparativo entre o total de interações produzidas por cada aluno e o total de interações consideradas válidas. O professor tem ainda a opção de escolher entre um gráfico com barras 2D ou 3D, através do menu suspenso gráficos. Deve-se observar que o resultado da avaliação deste módulo depende muito das escolhas do professor, como número mínimo de palavras-chave, percentual mínimo de interações consideradas válidas e percentual máximo de palavras desconhecidas, sendo por vezes necessário que o professor realize de 3 a 4 avaliações alterando suas opções até atingir um resultado satisfatório, conforme [WIV 99], KDT é uma atividade iterativa e interativa com aplicação repetida de métodos de mineração e interpretação dos resultados pelos usuários.

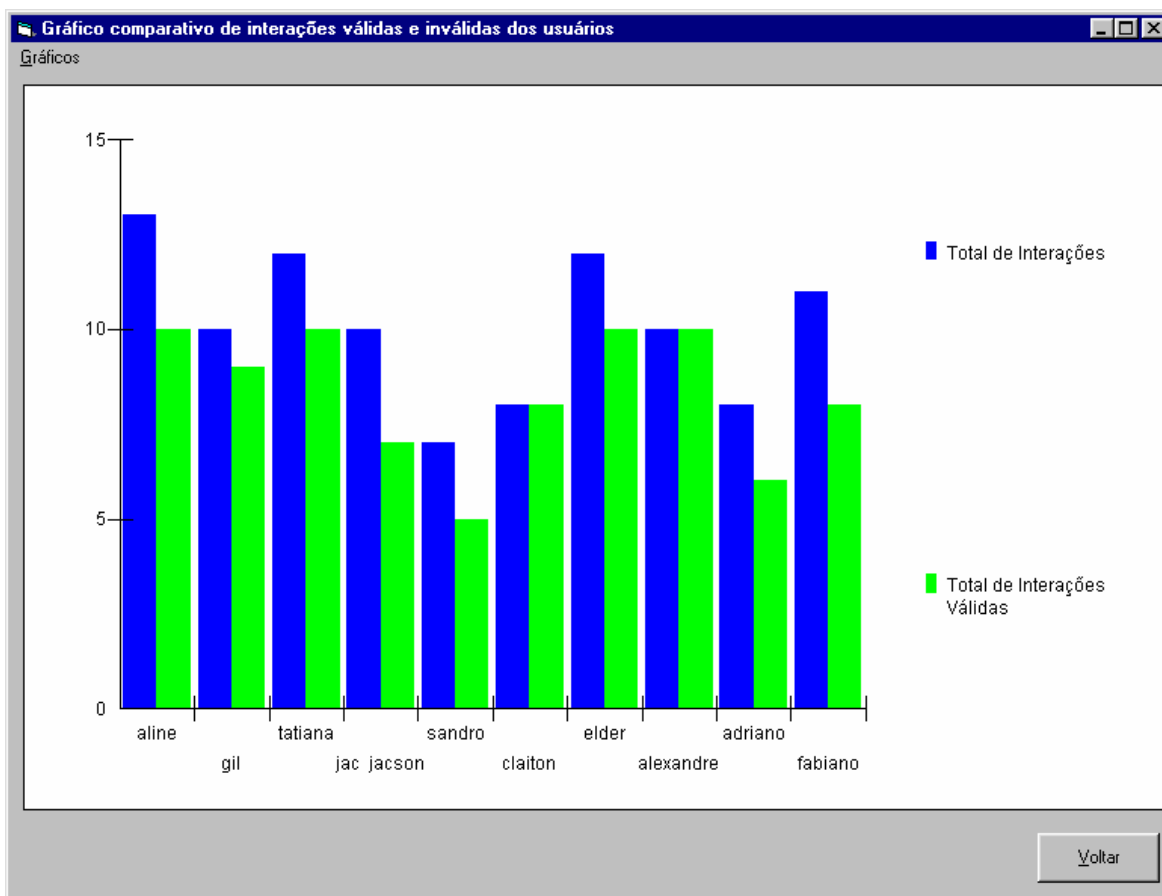


FIGURA 6.22 – Gráfico Comparativo entre o Total de Interações e o Total de Interações Consideradas Válidas de Cada Aluno

Pode-se notar por meio da figura anterior, que alguns aprendizes tiveram um aproveitamento excelente, já que o total de interações válidas é idêntico ao total de interações emitidas. Este módulo também serve para ilustrar o grau de participação dos aprendizes.

6.5.2 Avaliação por Questões (Queries)

Este módulo de avaliação permite ao professor selecionar quais palavras/frases-chave deseja pesquisar dentro do conjunto de interações de um aprendiz ou de todo o grupo. O professor pode escolher as palavras/frases-chave em um combo contendo todas as palavras-chave relacionadas ao tema escolhido ou, se preferir, pode digitar qualquer palavra ou frase dentro do combo e pressionar a tecla <ENTER>. Todas as palavras/frases-chave selecionadas ou digitadas são adicionadas a uma lista de pesquisa e, caso o professor queira, podem ser excluídas da mesma, bastando para isso selecioná-las com o mouse e pressionar a tecla <DELETE>. Este módulo está baseado na afirmação de [GOT 97] que

afirma ser um dos objetivos de KDT pesquisar por documentos relevantes através do uso de queries flexíveis e poderosas. A figura 6.23 apresenta a tela de avaliação por questões (queries):

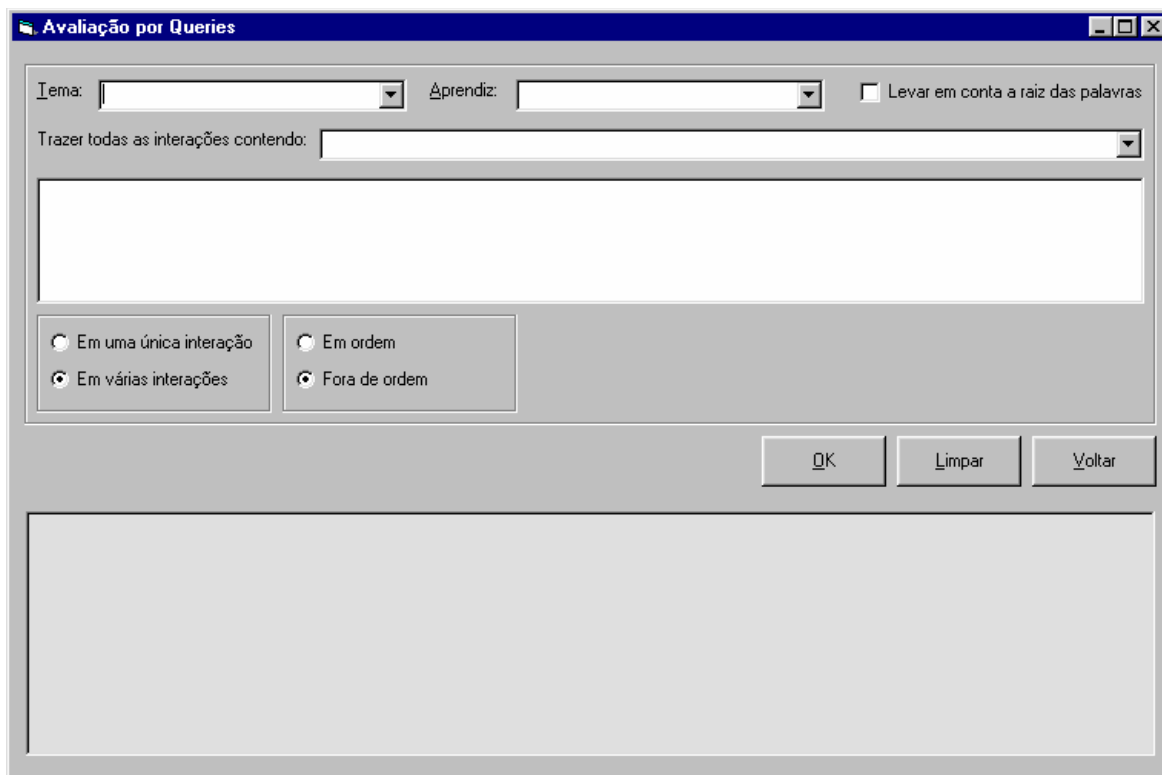


FIGURA 6.23 – Módulo de Avaliação por Questões (Queries)

O professor primeiramente deve selecionar o tema ao qual a discussão está relacionada (isso acarretará o preenchimento do combo com as palavras-chave relacionadas ao tema); em seguida deve informar se deseja fazer a avaliação sobre um aluno específico ou se deseja avaliar todos os alunos; após isso deve selecionar ou digitar as palavras/frases-chave que comporão a query de pesquisa e informar se deseja pesquisar somente interações contendo todas as palavras/frases-chave informadas ou se deseja pesquisar qualquer interação contendo alguma das palavras selecionadas, caso em que deverá informar ainda se quer que só sejam selecionadas interações contendo as palavras selecionadas em ordem ou se aceita qualquer interação que contenha alguma das palavras informadas em ordem ou não. O professor pode ainda informar se deseja pesquisar palavras/frases-chave por meio de sua raiz; o número de letras a serem retiradas de cada palavra para isolar sua raiz é definido no módulo de configurações. Este módulo de avaliação também está de acordo com a técnica de Descoberta por Extração de Passagens descrita em [WIV 99], onde, como já foi descrito, procura-se por passagens do interesse do usuário; e na técnica de Descoberta por Recuperação de Informações, também descrita em [WIV 99]. Neste caso, as passagens são representados pelas palavras/frases escolhidas ou digitadas pelo professor. A figura 6.24 ilustra uma avaliação por meio de uma questão (query), onde foram selecionadas as palavras “autocontrole emocional” e “inteligência emocional”.

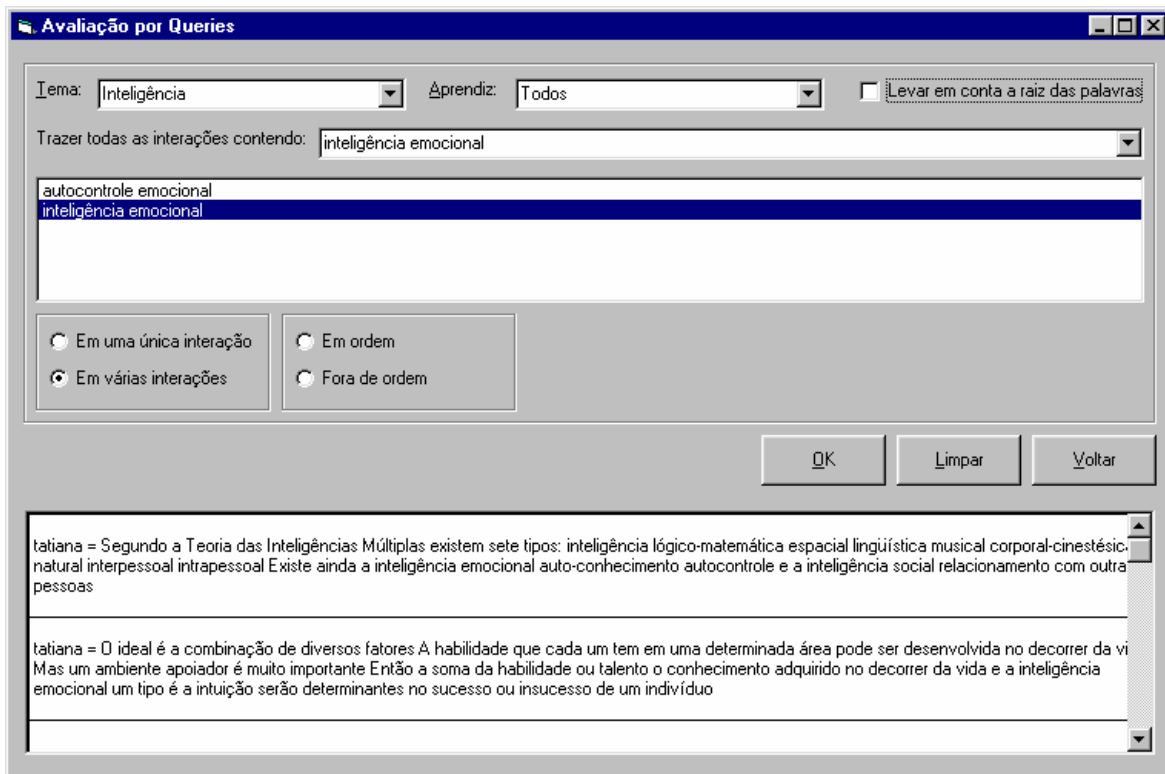


FIGURA 6.24 – Avaliação por meio de uma Questão (Query)

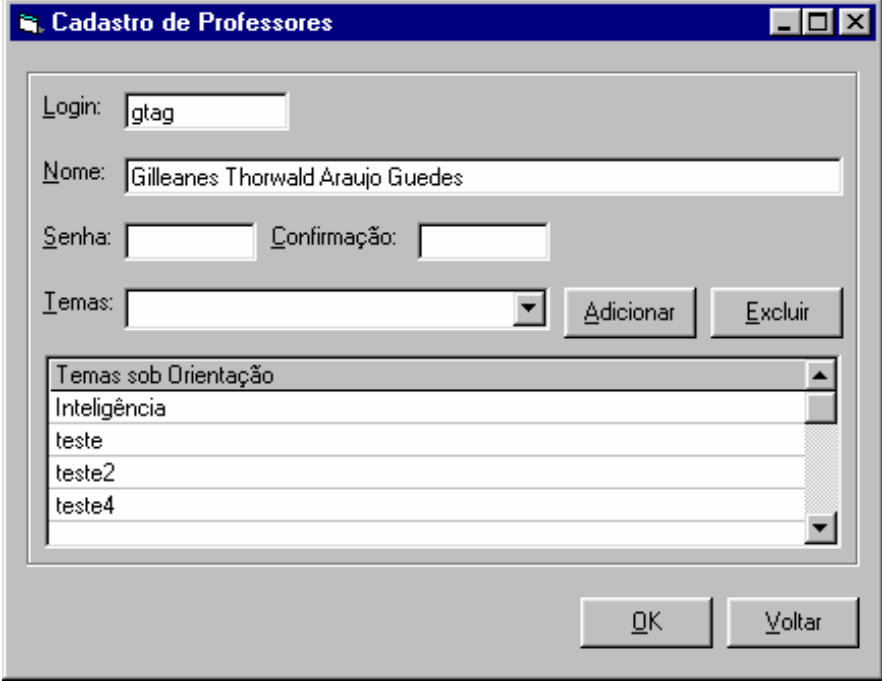
6.6 A Opção Cadastros

A opção de cadastro serve simplesmente para realizar os cadastros de professores, níveis, substantivos, stop words e temas com suas respectivas palavras-chave.

6.6.1 Cadastro de Professores

Este módulo permite o cadastramento dos professores que irão interagir com a ferramenta: o professor deve informar o nome-login pelo qual deseja entrar no sistema, bem como seu nome e a senha que utilizará para se autenticar. O professor deve também selecionar no combo “Temas”, todos os temas que deseja coordenar. Para adicionar um tema à lista dos temas que coordena, o professor deve selecionar o tema desejado no combo e em seguida pressionar o botão Adicionar. Caso o professor queira excluir algum dos temas que coordena, basta selecioná-lo e pressionar o botão excluir. O botão OK grava ou altera os dados do professor. Todos os campos são obrigatórios, inclusive o campo confirmação de senha, que serve para conter a senha digitada pelo professor. Além disso, o

professor precisa, obrigatoriamente, selecionar ao menos um tema. A fig. 6.25 apresenta a tela de cadastro de professores:



A imagem mostra uma janela de software intitulada "Cadastro de Professores". O formulário contém os seguintes campos e elementos:

- Login: gtag
- Nome: Gilleanes Thorwald Araujo Guedes
- Senha: [campo vazio] Confirmação: [campo vazio]
- Temas: [menu suspenso vazio]
- Botões: Adicionar, Excluir
- Lista de temas sob orientação:
 - Inteligência
 - teste
 - teste2
 - teste4
- Botões de controle: OK, Voltar

FIGURA 6.25 – Cadastro de professores

6.6.2 Cadastro de Níveis

Este módulo permite o cadastramento dos níveis que serão associados às palavras-chave. A tabela de níveis é composta somente de três campos: Código, Descrição e Pontuação. Este cadastro é opcional, caso o professor deseje dividir as palavras/frases-chave por nível de dificuldade e determinar um valor para o mesmo. Um exemplo da tela de cadastro de níveis pode ser vista na figura 6.26:

Código	Descrição	Pontuação
4	Avançado	10,00
1	Geral	1,00
3	Intermediário	5,00
2	Introdutório	1,00

FIGURA 6.26 – Cadastro de Níveis

Os botões deste módulo possuem as seguintes funções:

- O botão OK tem a função de gravar ou alterar um determinado registro.
- O botão Excluir exclui um registro selecionado.
- Os botões “<<” e “>>” têm a função de navegar entre os registros. O primeiro traz os registros anteriores e o segundo os seguintes.
- O botão Pesquisar preenche um “grid” com todos os campos cadastrados na tabela de níveis, caso o professor queira editar algum desses registros basta clicar sobre a linha do “grid” onde ele se encontra.

A pesquisa individual de um determinado nível por seu código, é feita quando o campo código do nível perde o foco.

6.6.3 Cadastro de Palavras Ferramentas (Stop Words)

Neste módulo o professor pode realizar a manutenção do cadastro de Palavras ferramentas (Stop words). Palavras ferramentas (Stop words) são palavras que são ignoradas nos módulos de avaliação, como artigos, conetivos, etc. Isto está de acordo com [LAR 2000], que declara que um dos métodos de pré-processamento antes da aplicação de técnicas de KDT, é a remoção de palavras ferramentas (stop words). É preciso destacar que o professor não precisa cadastrar uma a uma todas as palavras consideradas básicas por ele, pois o módulo de Classificação permite ao professor determinar quais palavras são consideradas básicas, bastando para isso selecionar através do mouse as palavras listadas e alterar sua classificação para stop word e estas serão automaticamente adicionadas à tabela de stop words, conforme já foi descrito anteriormente.

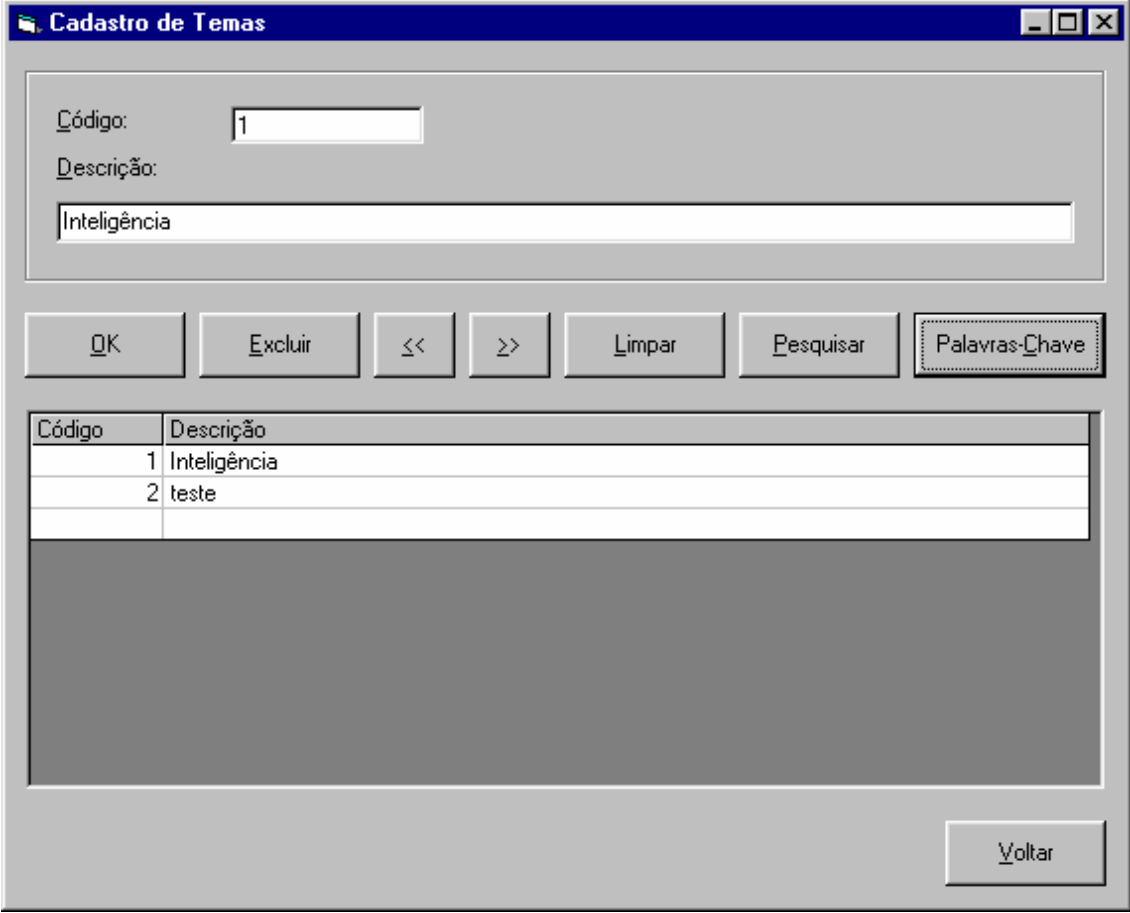
O cadastro de Stop words possui apenas dois campos: Código e Descrição; e sua interface é semelhante ao cadastro de níveis. A figura 6.27 apresenta a tela de cadastro de Palavras Ferramentas (Stop words):

Código	Palavra Básica
3	a
77	à
53	alguns
37	ao
41	aos
61	aquela
9	as
50	assim
32	até
5	com

FIGURA 6.27 – Cadastro de Palavras Ferramentas (Stop words)

6.6.4 Cadastro de Temas

O cadastro de temas possui uma interface semelhante aos cadastros anteriores, sendo única exceção o botão Palavras-Chave, que permite acessar o módulo de Cadastro de Palavras-Chave; no entanto, para acessá-lo, é necessário primeiro selecionar um tema e todas as palavras/frases-chave cadastradas estarão relacionadas ao tema escolhido. O cadastro de Temas possui dois campos: Código e Descrição, como pode ser observado na figura 6.28:



Código	Descrição
1	Inteligência
2	teste

FIGURA 6.28 – Cadastro de Temas

6.6.5 Cadastro de Palavras-Chave

Este módulo permite o cadastro de palavras ou frases-chave relacionadas a um determinado tema. Este módulo só pode ser acessado através do módulo de cadastro de temas. O módulo de cadastro de palavras-chave possui os campos Código, Palavra-Chave e Nível; este último campo deve ser selecionado em um combo que é preenchido com todos os níveis cadastrados quando a tela de cadastro de palavras-chave é carregada. O módulo de

cadastro de palavras-chave possui uma interface idêntica às descritas anteriormente. É preciso destacar a importância deste cadastro para os módulos de avaliação: o professor deve estar bem inteirado do assunto que virá a ser discutido pelos alunos e determinar que tópicos ele considerará importantes para serem citados e discutidos durante o chat. Obviamente, após as primeiras avaliações, é provável que o professor tenha que fazer certas alterações nos registros para permitir uma avaliação mais correta. A figura 6.29 apresenta a tela de Cadastro de Palavras/Frases-Chave.

Código	Palavra-Chave	Nível
8	altamente desenvolvida	Avançado
7	altamente desenvolvido	Avançado
14	ambiente de trabalho	Geral
39	autoconhecimento	Geral
24	autoconsciência	Geral
53	autocontrole emocional	Intermediário
21	capacidade intelectual	Geral
16	características sociais	Geral
48	controle emocional	Geral
31	cooperação	Geral
51	criatividade	Geral

FIGURA 6.29 – Cadastro de Palavras-Chave

6.7 A Opção Relatórios

Esta opção provê um conjunto de listagens dos cadastros descritos anteriormente: relatório de níveis, substantivos, stop words e temas com suas respectivas palavras-chave.

6.8 A Opção Configurações

Esta opção permite ao professor estabelecer algumas regras a serem utilizadas nos módulos de avaliação e classificação de palavras e serve também para informar qual o último arquivo de chat classificado automaticamente. Ao selecionar esta opção, surgirá uma tela semelhante a apresentada na figura 6.30:

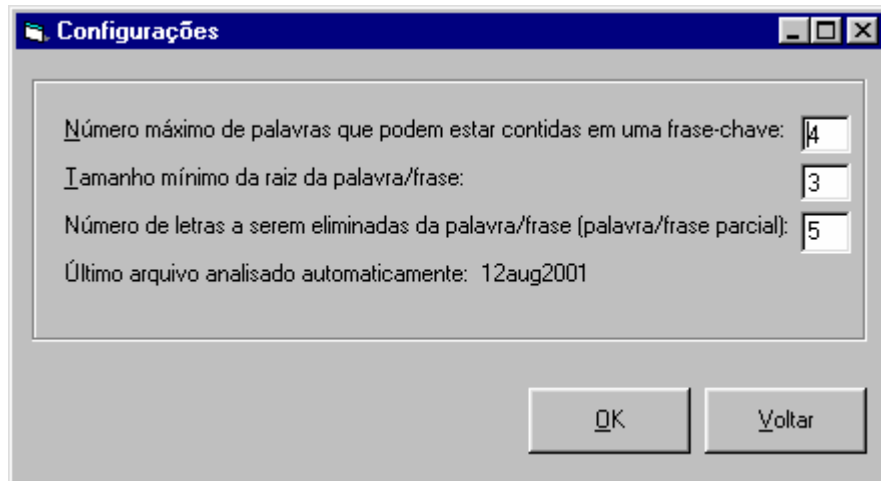


FIGURA 6.30 – Módulo de Configurações

O campo “Número Máximo de Palavras que podem estar Contidas em uma Frase-Chave” determina quantas palavras podem compor uma frase. Este valor é utilizado nos módulos de avaliação e classificação.

O campo “Tamanho Mínimo da Raiz da Palavra/Frase” determina o tamanho mínimo que uma palavra deve possuir, caso o professor queira comparar as raízes das palavras, para impedir que a palavra fique pequena demais e possa se igualar a palavras que possuam uma ou duas letras iniciais iguais, mas que tenham significados totalmente diferentes (erro de lemas [LOH 2000]). O campo seguinte também é utilizado na comparação de raízes: ele determina o máximo de letras que podem ser retiradas do início ou do fim da palavra para comparação. Finalmente o campo “Último arquivo analisado automaticamente” informa qual foi o último arquivo de chat que foi analisado automaticamente pelo software.

O botão OK permite gravar as alterações realizadas neste módulo. Todos os campos são obrigatórios, não podendo ser deixados em branco.

7 Conclusão e Trabalhos Futuros

A avaliação de um texto produzido de maneira colaborativa por meio de uma discussão ocorrida durante uma sessão de “chat” apresenta algumas dificuldades particulares, devido principalmente ao fato de que a conversação é muito coloquial e a discussão é muito dinâmica, sendo muito semelhantes às “tempestades cerebrais” utilizadas em algumas escolas. Além disso, as discussões são por vezes competitivas, onde alguns alunos querem se sobrepôr a outros, o que acarreta uma digitação rápida, causando freqüentes erros de português.

A ferramenta desenvolvida, descrita neste trabalho, procura auxiliar o professor a determinar se os aprendizes estão realmente discutindo os temas em questão, se não estão perdendo tempo com trivialidades e se estão realmente aprendendo e produzindo conhecimento.

Esta ferramenta então realiza algumas tarefas para este fim, como identificar os aprendizes, bem como suas respectivas interações, apresentar e classificar as palavras/frases utilizadas com seu nível de utilização e disponibilizar algumas formas de avaliação já descritas anteriormente, que basicamente procuram por ocorrências de palavras/frases, consideradas relevantes dentro do contexto discutido, nas interações produzidas pelos aprendizes. Obviamente, é preciso que o professor examine os resultados produzidos pela ferramenta, pois é possível que esta verifique a utilização de termos coerentes, mas não considerados relevantes inicialmente e deva fazer ajustes nas suas opções de avaliação, após os exames preliminares.

Com relação aos trabalhos relacionados descritos no início desta dissertação podemos identificar algumas semelhanças e diferenças como:

- A ferramenta procura e classifica substantivos como um dos algoritmos utilizados no trabalho de Jaques em [JAQ 99a], além de selecionar as palavras utilizadas mais freqüentemente, no entanto carece de um dicionário próprio tendo este que ser construído pelo próprio usuário.
- O software de “chat” utilizado não apresenta uma semi-estruturação como apresentada, por exemplo, nos sistemas de Barros em [BAR 2000] e Soller em [SOL 2001], seja como contribuições ou introdutores de sentença, o que dificulta intuir qual as intenções dos aprendizes ao inserirem suas interações. No entanto como no sistema de Barros, a ferramenta permite a elaboração de questões com o intuito de verificar o rendimento dos alunos e também o retorna o total de interações produzidas pelos alunos.
- A ferramenta, em sua versão atual, ainda não apresenta um modelo de domínio, nem estrutura os diálogos dos aprendizes em nós em uma árvore e não aplica nenhuma análise semântica como no sistema de Eleutério em [ELE 01]. No

entanto o sistema de Eleutério obriga os aprendizes a utilizarem o módulo HTML, que permite uma estruturação dos diálogos dos mesmos.

Acreditamos ter ajudado a enriquecer o ambiente AME-A, com o desenvolvimento desta ferramenta, associada ao software de “chat” já descrito, fornecendo-lhe, ainda que muitas das características do agente Promove_Interação não tenham ainda sido implementadas no software de “chat”, um ambiente em que os diversos aprendizes que utilizam o ambiente AME-A possam virtualmente se reunir, trocar idéias através da discussão de determinados assuntos e produzir textos de maneira colaborativa. Além disso, a ferramenta descrita neste trabalho fornece uma maneira de auxiliar o professor a determinar o avanço de seus aprendizes na aquisição de conhecimentos por meio das discussões, provendo maneiras do professor verificar os termos mais freqüentemente utilizados, bem como determinar se os aprendizes estão discutindo todos os tópicos propostos e não se estão desviando do assunto em questão.

Como um trabalho futuro, pretende-se melhorar o software de “chat” como acrescentar introdutores de sentença ao mesmo, para facilitar a identificação das intenções dos aprendizes durante as discussões.

Seria útil também a adição de um corretor ortográfico que eliminasse erros gramaticais e de digitação dos arquivos de chat, antes destes serem analisados, o que evitaria que muitos termos promissores fossem desprezados pela ferramenta. Outra adição útil seria o acréscimo de um léxico mais completo de português, onde estivessem discriminadas as principais palavras do idioma, identificando sua classificação como substantivo, verbo, etc.

Pretende-se também melhorar os métodos de avaliação da ferramenta, como por exemplo, possibilitar que a ferramenta retorne a média de freqüência de uso de uma palavra-chave, o que não é exatamente a quantidade de vezes em que ela foi citada, mas sim, o intervalo de tempo que demora para ela ser citada e o tamanho médio das interações dos aprendizes. Além disso, pretende-se que a ferramenta possa classificar as interações dos aprendizes de acordo com os introdutores de sentença utilizados por eles durante as discussões.

Bibliografia

- [ALE 2001] ALEVEN, Vincent; POPESCU, Octav; KOEDINGER, Kenneth. Pedagogical Content Knowledge in a Tutorial Dialogue System to Support Self-Explanation. In: AI-ED, 2001. **Workshop Papers: Multi-agent Architectures for Distributed Learning Environments**. [S.l.:s.n.], 2001. p. 67-70.
- [BAR 2000] BARROS, Beatriz; VERDEJO, Felisa. Analyzing Student Interaction Process in Order to Improve Collaboration. The DEGREE Approach. **International Journal of Artificial Intelligence in Education, IJAIED**, 2000. Disponível em: <www.cbl.leeds.ac.uk/ijaied>. Acesso em set. 2001.
- [BIG 98] BIGUS, Joseph P.; BIGUS, Jennifer. **Constructing Intelligent Agents with JAVA**. New York: Wiley Computer Publishing, 1998.
- [CAS 99] CASAS, Luís Alberto Alfaro. **Contribuições para a Modelagem de um Ambiente Inteligente de Educação Baseado em Realidade Virtual**. 1999. Tese (Doutorado em Ciência da Computação) - Universidade Federal de Santa Catarina, Florianópolis. Disponível em: <<http://www.eps.ufsc.br/teses99/casas/index.html>>. Acesso em fev. 2000.
- [CAZ 97] CAZELLA, Sílvio César. **Uma Arquitetura para Coordenar a Interação de Agentes na Internet**. 1997. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [DAM 99] D'AMICO, Carmem B. **Aprendizagem Estática e Dinâmica em Ambientes Multiagentes de Ensino-Aprendizagem**. 1999. Tese (Doutorado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [DIX 97] DIXON, Mark. **An Overview of Document Mining Technology**. 1997. Disponível em: <<http://citeseer.nj.nec.com/feldman98text.html>>. Acesso em: jun. 2001.
- [ELE 2001] ELEUTERIO, Marco et al. AMANDA – An Intelligent Dialog Coordination Environment. In: AI-ED, 2001. **Workshop Papers: Tutorial Dialogue Systems**. [S.l.:s.n.], 2001. p.13-22.
- [FEL 95] FELDMAN, Ronen; DAGAN, Ido. Knowledge Discovery in Textual Databases (KDT). In: INTERNATIONAL CONFERENCE ON KNOWLEDGE, DISCOVERY, KDD, 1995. **Proceedings...** Disponível

- em: <<http://citeseer.nj.nec.com/feldman98text.html>>. Acesso em: jun. 2001.
- [FEL 98] FELDMAN, Ronen et al. Text Mining at the Term Level. In: EUROPEAN SYMPOSIUM ON PRINCIPLES OF DATA MINING AND KNOWLEDGE DISCOVERY, PKDD, 2., 1998. **Proceedings ...** Disponível em: <<http://citeseer.nj.nec.com/feldman98text.html>>. Acesso em: jun. 2001.
- [FRE 98] FREITAS, Henrique; MOSCAROLA, Jean; JENKIN, Milton. **Content and Lexical Analysis: A Qualitative Practical Application**. Baltimore: Information Systems Research Group (ISRC), University of Baltimore, 1998. (WP ISRC, n. 070498).
- [FRO 97] FROZZA, Rejane. **SIMULA – Ambiente para Desenvolvimento de Sistemas Multiagentes Reativos**. 1997. Dissertação (Mestrado em Ciência da Computação) - Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [GIR 99] GIRAFFA, Lúcia Maria Martins. **Uma Arquitetura de Tutor Utilizando Estados Mentais**. 1999. Tese (Doutorado em Ciência da Computação) - Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [GIR 99a] GIRAFFA, Lúcia Maria Martins; VICCARI, Rosa Maria. Intelligent Tutoring Systems Built Using Agent Techniques. **Revista de Educação, Ciência e Cultura**, Canoas, v. 4, n. 1, p. 23-39, 1999.
- [GIR 99b] GIRAFFA, Lúcia Maria Martins; VICCARI, Rosa Maria. Estratégias de Ensino em Sistemas Tutores Inteligentes Modelados Através da Tecnologia de Agentes. **Revista Brasileira de Informática na Educação**, [S. i.], n. 5, p. 9-17, 1999.
- [GOT 97] GOTTHARD, Willi; MARWICK, Alan; SEIFFERT, Roland. Mining Text Data. **DB2 Magazine**, [S. i.], v. 2, n. 3, 1997. Disponível em: <http://www.db2mag.com/db_area/archives/1997/q4/>. Acesso em: jun. 2001.
- [HEA 99] HEARST, Marti A. **Untangling Text Data Mining**. Berkeley: School of Information Management & Systems, University of California, 1999. Disponível em: <<http://www.sims.berkeley.edu/~hearst/papers/ac199/ac199-tdm.html>>. Acesso em: jul. 2000.
- [HÜB 95] HÜBNER, Jomi Fred. **Migração de Agentes em Sistemas Multiagentes Abertos**. 1995. Dissertação (Mestrado em Ciência da Computação) –

Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.

- [JAQ 99a] JAQUES, Patrícia Augustin. **Agentes de Software para Monitoramento da Colaboração em Ambientes Telemáticos de Ensino**. 1999. Dissertação (Mestrado em Ciência da Computação) – Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre.
- [JAQ 99b] JAQUES, Patricia Augustin; OLIVEIRA, Flávio Moreira de. Software Agents for Collaboration Analysis in a Virtual Classroom. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, AI-ED, 1999. **Proceedings ...** Disponível em: <www.inf.ufrgs.br/~pjaques>. Acesso em: abr. 2001.
- [KOD 99] KODRATOFF, Yves. About Knowledge Discovery in Texts: A Definition and a Example. In: INTERNATIONAL SYMPOSIUM ON METHODOLOGIES FOR INTELLIGENT SYSTEMS, ISMIS, 11., 1999, Warsaw, Poland. **Foundations of Intelligent Systems: proceedings...** Berlin: Springer, 1999. Disponível em: <<http://citeseer.nj.nec.com/feldman98text.html>>. Acesso em: jun. 2001.
- [LAR 2000] LAROCCA, Joel et al. Document Clustering and Text Summarization. In: INTERNATIONAL CONFERENCE ON PRACTICAL APPLICATIONS OF KNOWLEDGE DISCOVERY AND DATA MINING, PADD, 4., 2000. **Proceedings ...** London: The Practical Application Company, 2000. p. 41-55. Disponível em: <<http://citeseer.nj.nec.com/297292.html>>. Acesso em: set. 2001.
- [LIM 84] LIMA, Lauro de Oliveira. **Construção do Homem Segundo Piaget - Uma Teoria da Educação**. São Paulo: Summus, 1984.
- [LOH 2000] LOH, Stanley; WIVES, Leandro Krug; OLIVEIRA, José Palazzo M. de. Concept-Based Knowledge Discovery in Texts Extracted from the Web. **SIGKDD Explorations**, [S. l.], v. 2, n. 1, p. 29-39, 2000. Disponível em: <www.acm.org/sigs/sigkdd/explorations/issue21/contents.htm#Loh>. Acesso em: jun. 2001.
- [MOI 99] MOISSA, Harry Erwin. **Identificação de Fatores Motivacionais e Afetivos em um Ambiente de Ensino e Aprendizagem**. 1999. Trabalho Individual (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [OHA 96] O'HARE, G. M. P.; JENNINGS, N. R. **Foundations of Distributed Artificial Intelligence**. New York: John Wiley & Sons, 1996.
- [PER 97] PEREIRA, Adriana Soares. **Um Estudo de Aplicações de Ensino na Internet Orientadas a Agentes**. 1997. Trabalho Individual (Mestrado em

Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.

- [RAJ 98] RAJMAN, Martin; BESANÇON, Romaric. Text Mining – Knowledge Extraction from Unstructured Textual Data. In: EUROPEAN SYMPOSIUM ON PRINCIPLES OF DATA MINING AND KNOWLEDGE DISCOVERY, PKDD, 1998, Nantes. [**Proceedings ...**]. [S.l.: s.n.], 1998. p. 23-26. Disponível em: <<http://liawww.epfl.ch/~inmain/old/tln-gb-public.html>>. Acesso em: jun. 2001.
- [RIC 2001] RICKEL, Jeff et al. Using a Model of Collaborative Dialogue to Teach Procedural Tasks. In: AI-ED, 2001. **Workshop Papers** [S.l.: s.n.], 2001. p. 9-12.
- [RUS 95] RUSSEL, Stuart; NORVIG, Peter. **Artificial Intelligence - A Modern Approach**. New Jersey: Prentice Hall, 1995.
- [SOL 2001] SOLLER, Amy L. Supporting Social Interaction in an Intelligent Collaborative Learning System. **International Journal of Artificial Intelligence in Education**, IJAIED, 2001. Disponível em: <www.cbl.leeds.ac.uk/ijaied>. Acesso em: set. 2001.
- [VIC 89] VICCARI, R. **Um Tutor Inteligente para a Programação em Lógica: Idealização, Projeto e Desenvolvimento**, 1989. Tese (Doutorado) – Faculdade de Ciência e Tecnologia, Universidade de Coimbra, Coimbra.
- [WIV 99] WIVES, Leandro Krug; LOH, Stanley. **Tecnologias de Descoberta de Conhecimento em Informações Textuais**. Porto Alegre: PPGC-UFRGS, 1999.
- [YEA 2000] YEATES, Stuart; WITTEN, Ian. On tag insertion and its complexity. In: INTERNATIONAL WORKSHOP ON TEXT AND DATA MINING, PRICAI, 2000. [**Proceedings ...**]. [S.l.: s.n.], 2000. p. 52-63. Disponível em: <<http://www.cs.waikato.ac.nz/~say1/publications.html>>. Acesso em: jun. 2001.