

**Chapa: Um Agente de Interface para Ferramentas de
Bate-papo em Ambientes de Ensino à Distância na
Web**

Ricardo Luís Lachi

Dissertação de Mestrado

Instituto de Computação
Universidade Estadual de Campinas

Chapa: Um Agente de Interface para Ferramentas de Bate-papo em Ambientes de Ensino à Distância na Web

Ricardo Luís Lachi

Maio de 2003

Banca Examinadora:

- Prof^a. Dr^a. Heloísa Vieira da Rocha (**Orientadora**)
Instituto de Computação – UNICAMP
- Prof^a. Dr^a. Ariadne Maria Brito Rizzoni Carvalho
Instituto de Computação – UNICAMP
- Prof. Dr. Ricardo Ribeiro Gudwin
Faculdade de Engenharia Elétrica e Computação,
Departamento de Engenharia da Computação e Automação
Industrial – UNICAMP
- Prof. Dr. Jacques Wainer (suplente)
Instituto de Computação – UNICAMP

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Lachi, Ricardo Luís

L117c Chapa: um agente de interface para ferramentas de
bate-papo em ambientes de ensino à distância na Web / Ricardo Luís
Lachi -- Campinas, [S.P. :s.n.], 2003.

Orientador : Heloísa Vieira da Rocha

Dissertação (mestrado) - Universidade Estadual de Campinas,
Instituto de Computação.

1. Educação à distância. 2. Interação homem-máquina. 3.
Inteligência artificial. I. Rocha, Heloísa Vieira da. II. Universidade
Estadual de Campinas. Instituto de Computação. III. Título.

**Chapa: Um Agente de Interface
para Ferramentas de Bate-papo em Ambientes de Ensino
à Distância na Web**

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Ricardo Luís Lachi e aprovada pela Banca Examinadora.

Campinas, 27 de maio de 2003.

**Profa. Dra. Heloísa Vieira da Rocha
(orientadora)**

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

© Ricardo Luís Lachi, 2003.
Todos os direitos reservados.

Para minha mãe e meu pai, com carinho.

A persistência é o caminho do êxito.

Charles Chaplin

Agradecimentos

Aos meus pais, pelo carinho, apoio e incentivo no desenvolvimento deste trabalho.

À Prof^a Dr^a Heloísa Vieira da Rocha, pela orientação na realização desta pesquisa.

Aos colegas do Núcleo de Informática Aplicada à Educação – NIED da Unicamp, em especial ao Diogo e Sérgio pelo apoio técnico, a Carmen, ao Celmar e Ciça pelas contribuições a esta pesquisa.

A todos os amigos e colegas que, direta ou indiretamente, contribuíram para a realização deste trabalho.

Ao CNPq, pelo apoio financeiro.

Resumo

A educação à distância se caracteriza pelo não existência, parcial ou total, do elemento ‘presencial’ na relação de aprendizagem professor-aluno. Isto é, o professor e aluno encontram-se em localidades distintas (fora de uma sala de aula) e necessitam do uso de um canal de comunicação para mediar a interação.

Nos últimos anos, a educação à distância, via ambientes que utilizam a Internet como meio de comunicação, tem aumentado significativamente. Esses ambientes caracterizam-se por disponibilizar inúmeras ferramentas que dão suporte ao processo de comunicação via rede (ferramentas de bate-papo, correio eletrônico, grupos de discussão). Na literatura, existem inúmeros artigos que relatam o enorme volume de trabalho necessário por parte de um professor de um curso à distância para poder acompanhar adequadamente o desenvolvimento dos alunos.

Com o intuito de auxiliar o professor nessa tarefa de acompanhar o desenvolvimento dos alunos de um curso à distância foi desenvolvido o agente de interface Chapa descrito em detalhes nesta dissertação.

O foco do Chapa é ajudar na análise das contribuições dos participantes de uma sessão de bate-papo ocorrida no ambiente de Educação à Distância TelEduc. Nesse sentido, o Chapa é capaz de selecionar quais as mensagens mais interessantes a um determinado usuário permitindo a este analisar somente uma fração do número total de mensagens

existentes em uma sessão. Essa seleção é feita com base em um dicionário de dados específico, que constrói para cada usuário, por meio das técnicas de aprendizagem inerentes ao conceito de agentes de interface.

Neste trabalho, já são apresentados alguns testes preliminares feitos com o Chapa por um grupo de usuários. No entanto, mesmo tendo sido bastante positivos os resultados desses testes iniciais, ainda é necessária a realização de outros mais, em situações reais de um curso à distância, para se validar melhor o Chapa.

Abstract

Distance education is characterized by the partial or total absence of the “presential” element in the teacher-student relationship, that is, teacher and student are far from each other and need to use a communication channel to mediate the interaction.

In the last few years, distance education based on environments that use Internet as a communication channel has grown larger and larger. These environments make available many tools that support communication through the Internet (for example: chat, e-mail, newsgroup). In the literature, many articles relate the large amount of work needed by the teacher of a distance course to accompany properly the development of the students.

In order to help the teacher with the task of accompanying the development of the students in a distance course we developed the interface agent Chapa, described in details in this dissertation.

Chapa’s focus is on helping the analysis of participants’ contributions to a chat session that took place in TelEduc, a supporting environment for distance education. In order to do that, it is capable of selecting which messages are more interesting to one user, allowing him to analyze only a fraction of the total number of messages that exist in a session. It is able to do this selection based on a specific data dictionary which is built for each user by means of learning techniques inherent to the technology of interface agents.

In this work, some preliminary tests were done with Chapa. However, even though the results were very positives, it is still necessary to realize more tests in real world courses in order to better proof Chapa's effectiveness.

Lista de Figuras

| | |
|--|----|
| Figura 2.1 – Caracterização esquemática das quatro categorias de agentes apresentadas com base nas características de <i>Cooperação, Capacidade de Aprendizagem e Autonomia</i> | 31 |
| Figura 2.2 – Classificação dos diferentes tipos de agentes | 32 |
| Figura 2.3 – Apresentação das quatro fontes de aprendizagem de um agente de interface propostas por Maes (1998) | 39 |
| Figura 4.1 – Visualização do registro de uma sessão de bate-papo | 62 |
| Figura 4.2 – Localização das informações disponíveis na nova interface de visualização do registro do bate-papo..... | 63 |
| Figura 4.3 – Interface de visualização das informações apresentadas pelo agente | 65 |
| Figura 4.4 – Primeira vez que se visualiza o registro de uma determinada sessão de bate-papo. A opção “Selecionar” aparece disponível para todas as mensagens da sessão, uma vez que nenhuma delas foi selecionada ainda..... | 66 |
| Figura 4.5 – Janela de indicação dos critérios de seleção de uma mensagem..... | 67 |
| Figura 4.6 – Feedback apresentado pelo agente após a seleção de uma mensagem pelo usuário..... | 69 |
| Figura 4.7 – Janela com os critérios pelos quais uma mensagem foi selecionada | 70 |
| Figura 4.8 – (a) Janela onde aparece um menu com todas as opções de configuração do agente. (b) Janela onde aparece a base de conhecimentos do agente. | 71 |
| Figura 4.9 – Visão em detalhes de um motivo armazenado na base de conhecimentos do agente | 72 |
| Figura 4.10 – Janela que permite a definição de um novo motivo de seleção de mensagens pelo usuário..... | 73 |

| | |
|--|----|
| Figura 4.11 – Janela de busca de motivos armazenados na base de conhecimentos do agente..... | 74 |
| Figura 4.12 – Resultado da busca pelos motivos cujo elemento “Remetente” tem o valor “Elielma Alves Salgado” | 75 |
| Figura 4.13 – Apresentação visual do resultado de uma análise efetuada pelo Chapa | 77 |
| Figura 4.14 – Janela que ativa ou não a aplicação do algoritmo de classificação Bayesiano. | 80 |
| Figura 4.15 – Esquema de visões. O agente do participante A pode visualizar a análise feita pelo agente do participante B | 81 |
| Figura 4.16 – Compartilhamento do agente | 82 |
| Figura 4.17 – Usuários que compartilharam os agentes. No caso da figura está destacado que o participante ‘Aluno’ compartilhou seu agente | 83 |
| Figura 4.18 – Visão do participante “Usuário”. | 83 |
| Figura 4.19 – As modificações efetuadas na interface do Chapa com base nas sugestões feitas pelos usuários durante a análise | 86 |
| Figura 4.20 – As modificações na interface de seleção uma mensagem por meio de um clique sobre o link de opção “Selecionar” | 87 |
| Figura 4.21 – As modificações na interface que permite desmarcar a seleção de uma mensagem. | 88 |

Lista de Tabelas

Tabela 3.1 – Algoritmo Bayesiano simples para a aprendizagem e classificação de documentos textuais.....57

Tabela 4.1 – Exemplo da análise e seleção de uma mensagem feita pelo Chapa..... 79

Sumário

| | |
|---|-----------|
| <i>Resumo</i> | <i>ix</i> |
| <i>Abstract</i> | <i>xi</i> |
| Capítulo 1 – Introdução | 18 |
| Capítulo 2 – Agentes de software | 26 |
| 2.1 Definição | 26 |
| 2.2 Classificação | 29 |
| 2.3 Construção de Agentes de Interface | 32 |
| 2.3.1 Abordagem 1..... | 33 |
| 2.3.2 Abordagem 2..... | 34 |
| 2.3.3 Abordagem 3..... | 35 |
| 2.4 Considerações finais | 40 |
| Capítulo 3 – Aprendizagem de Máquina | 43 |
| 3.1 Introdução | 43 |
| 3.2 Teoria Bayesiana | 46 |
| 3.3 Classificador Bayesiano Simples | 50 |
| 3.3.1 Exemplo ilustrativo | 52 |
| 3.3.2 Estimando probabilidades | 54 |
| 3.4 Algoritmo de aprendizagem | 56 |
| 3.5 Considerações finais | 58 |

| | |
|--|-----------|
| Capítulo 4 – Agente de Interface Chapa | 60 |
| 4.1 Introdução | 60 |
| 4.2 Interface para seleção de mensagens | 62 |
| 4.3 Formas de seleção de mensagens | 65 |
| 4.3.1 Opção de ação “Selecionar”/ “Desmarcar seleção”..... | 66 |
| 4.3.2 Link “Configurando o agente”..... | 71 |
| 4.4 Aprendizagem do Chapa | 75 |
| 4.4.1 Aprendizagem por observação..... | 75 |
| 4.4.2 Aprendizagem via feedback do usuário..... | 76 |
| 4.4.3 Aprendizagem a partir de exemplos..... | 78 |
| 4.5 Análise do Chapa | 78 |
| 4.6 Múltiplas visões | 81 |
| 4.7 Testes preliminares | 84 |
| 4.7.1 Análise 1..... | 85 |
| 4.7.2 Análise 2..... | 89 |
| 4.7.3 Análise 3..... | 90 |
| 4.8 Considerações Finais | 91 |
| Capítulo 5 – Conclusões | 93 |
| Referências bibliográficas | 97 |

Capítulo 1

Introdução

A Internet, nos dias de hoje, está se expandindo cada vez mais e alcançando um número cada vez maior de pessoas. Isso porque é um meio de comunicação largamente difundido que permite a comunicação sem fronteiras, independente de distâncias, políticas nacionais ou internacionais. Um fator importante, decisivo para a sua expansão, foi o surgimento da interface gráfica World Wide Web (*WWW*) que facilitou o seu uso e, por conseqüência, agregou um número maior de pessoas que dela passaram a fazer uso, devido à não necessidade de um grande conhecimento de protocolos de comunicação para se conseguir acesso às informações.

Como resultado direto da expansão da Internet, foi aberto um amplo campo de opções de uso a serem exploradas, dentre elas, pode-se destacar a sua aplicação na área de Educação à Distância (EaD).

Educação à Distância é uma forma de aprendizado onde instrutor e aprendiz estão separados no tempo e/ou espaço durante todo ou parte do processo de aprendizagem. Sua aplicação é uma alternativa para a obtenção de conhecimento fugindo às tradicionais formas de ensino, principalmente para pessoas que não têm acesso às formas presenciais tradicionais.

Atualmente, a EaD baseada na Internet tem tido uma expansão assustadora, principalmente como conseqüência da sua possibilidade de utilização no treinamento de pessoal em grandes empresas e na ligação entre centros formadores e regiões distantes geograficamente.

Um outro fator que tem influído na expansão da EaD via computador é o surgimento e o desenvolvimento de ambientes para apoiar cursos à distância na rede. Existem diversos ambientes disponíveis para uso no mercado, gratuitos ou não. Dentre eles pode-se citar:

- **WebCT** – criado e comercializado pelo Departamento de Ciência da Computação da *University of British Columbia*, dá suporte à criação de cursos sofisticados baseados na Internet para usuários leigos na área de computação (GOLDBERG e SALARI, 1997; WEBCT, 2000). Disponibiliza uma gama variada de ferramentas de comunicação tanto síncronas quanto assíncronas (WEBCT, 2000).
- **AulaNet** – criado pelo Laboratório de Engenharia de Software do Departamento de Informática da PUC-Rio, em 1997. Ele dá suporte à criação, manutenção, administração e assistência de cursos via Internet para pessoas sem conhecimentos técnicos de computação. Seu principal objetivo é permitir a utilização da Internet como um meio viável para a educação, tentando reutilizar material didático já existente e incentivando a criação de comunidades de conhecimento (FUCKS, 2000; AULANET, 2000).
- **TelEduc** – é um ambiente de suporte ao ensino-aprendizagem à distância criado em 1997 e que vem sendo continuamente aprimorado pelo Núcleo de Informática Aplicada à Educação (NIED) em conjunto com o Instituto de Computação, ambos da Universidade Estadual de Campinas (UNICAMP). Dentre as características desse ambiente pode-se citar a possibilidade de

contextualização das atividades realizadas pelos alunos, disponibilidade de ferramentas de comunicação tanto síncronas quanto assíncronas, além de um conjunto de ferramentas adicionais que permitem ao professor a autoria e o gerenciamento de um curso ministrado por meio do ambiente (TELEDUC, 2000).

Como pode ser visto, os ambientes de EaD agrupam e disponibilizam, em maior ou menor quantidade, ferramentas que dão suporte ao processo de comunicação via rede. Como exemplo de algumas das ferramentas existentes nesses ambientes pode-se citar: ferramentas para troca de mensagens (correio eletrônico), de bate-papo, de vídeo-conferência, listas de discussão, mural de recados, agendas, etc. Em suma, todos os ambientes de EaD constituem-se, no fundo, de um conjunto sofisticado de ferramentas e, a cada nova versão, procuram adicionar novos recursos que permitam um melhor acompanhamento do desenvolvimento e aprendizado por parte dos alunos. No entanto, esses recursos ainda são insuficientes, pois os professores continuam tendo dificuldades para acompanhar a turma, interagir com ela e ter uma percepção melhor do que ocorre com os alunos de forma a poder auxiliá-los.

Em geral, a carga de trabalho de um professor que ministra um curso à distância é maior do que seria se o mesmo curso fosse dado presencialmente. Atualmente, quase toda a interação aluno-professor é feita via texto e, desse modo, acaba se tornando muito trabalhoso para o professor acompanhar adequadamente o desenvolvimento dos alunos. Isso porque é necessário que o professor analise toda a informação textual gerada por todos os alunos do curso à distância para poder saber como estão evoluindo, quais as dificuldades que estão enfrentando, para ser capaz de efetuar uma avaliação consistente do desempenho deles durante o transcorrer do curso.

Com base nessa necessidade de análise de um volume muito grande de informações é que a maior parte das pesquisas atuais se concentram: como ajudar o professor a lidar com o enorme volume de informações geradas em um curso à distância.

Na literatura há inúmeros trabalhos que se propõem a auxiliar o professor nessa tarefa. Nesses trabalhos o caminho que está sendo seguido é o de tentar *organizar e filtrar* a informação existente em um curso à distância de forma a que fique mais fácil para o professor recuperá-la e analisá-la; em nenhum dos trabalhos é proposta uma análise qualitativa do valor das informações que são recuperadas e organizadas; essa avaliação final é sempre deixada para ser efetuada pelo próprio professor. Pimentel e Sampaio (2001); Musa e Oliveira (2001); Jaques e Oliveira (2000) são exemplos de alguns dos trabalhos existentes nessa linha.

Nesta dissertação o foco do trabalho desenvolvido também foi o de disponibilizar uma ferramenta capaz de auxiliar o professor na organização e filtragem de informações em um curso à distância. Especificamente para esta dissertação foi tomada como referência principal o trabalho de Jaques e Oliveira (2000).

Nesse trabalho é apresentada uma arquitetura Multi-agente para realizar o monitoramento das ferramentas de comunicação em um ambiente de ensino à distância. O sistema construído no trabalho faz a análise das discussões que se encontram em andamento nas ferramentas de lista de discussão, bate-papo e grupos de notícias, disponibilizando ao professor algumas informações estatísticas (percentual de participação dos alunos e número de mensagens trocadas) e identificando possíveis associações nas interações, tais como tópicos e sub-tópicos que interessam ao aluno, grupos de alunos que interagem mais entre si, etc.

Além disso, no sistema multi-agente proposto por Jaques e Oliveira (2000) há, para cada um dos mecanismos de comunicação Internet (lista de discussão, bate-papo e grupo de

notícias), um agente responsável por buscar periodicamente as informações em cada um deles. Por exemplo, há um agente encarregado da lista de discussão, que é acionado pelo sistema de tempos em tempos para buscar todas as novas mensagens de correio eletrônico que chegam, um agente responsável pelo grupo de notícias e também um agente responsável pelo arquivo de registro do bate-papo. Esses agentes são denominados *agentes coletores* pois são responsáveis por buscar os dados que depois de analisados vão ser enviados para um agente central, denominado *agente do professor*, quando este solicitar.

A análise dos dados coletados também é efetuada pelos agentes coletores e consiste na filtragem das novas mensagens coletadas por meio dos assuntos e sub-assuntos presentes nelas. A identificação desses assuntos e sub-assuntos é feita pelo conteúdo do *subject* ou por palavras-chave no conteúdo da própria mensagem. Os agentes consideram palavra-chave todos os substantivos e verbos encontrados no texto, desconsiderando advérbios, preposições e valores numéricos. Isso é feito dessa forma porque considera-se que os substantivos e verbos possuem valor semântico maior que as demais palavras.

A fim de verificar os significados sintático e morfológico das palavras-chave, os agentes se utilizam de um Dicionário Léxico-Morfológico do projeto Lexis (LIMA et al., 1997 apud JAQUES e OLIVEIRA, 2000) e de um Thesaurus provido pelo sistema. O uso do dicionário léxico é para identificar os substantivos e verbos no texto, enquanto que o Thesaurus é usado para identificar sinônimos e relações de hierarquia entre as palavras (sub-assuntos).

Com base nesse trabalho de Jaques e Oliveira (2000) é que foi pensado e proposto o agente desta dissertação, nomeado Chapa – representando *Chat Analyser Program*, já que esta é justamente a função dele: analisar sessões de bate-papo. Em linhas gerais, foi construído neste trabalho de mestrado um agente de interface capaz de selecionar quais as mensagens, dentre todas as mensagens de uma sessão de bate-papo já realizada e registrada

no ambiente TelEduc, são as mais interessantes para um determinado professor de um curso à distância.

Especificamente, foram focadas as interações efetuadas via ferramenta de bate-papo do ambiente de ensino à distância TelEduc. Isso porque, dada a natureza síncrona desta ferramenta, é a que gera um maior número de informações irrelevantes e descartáveis.

O objetivo principal, à semelhança dos outros trabalhos citados anteriormente, não foi tentar analisar as mensagens de uma sessão de bate-papo e tirar conclusões a respeito das mesmas e sim, diminuir o número de mensagens de uma sessão de bate-papo já realizada que o professor tem que ler para poder analisar o que foi discutido, qual a contribuição/participação dos alunos, etc.

A utilização da classe de agentes de interface foi decorrência do fato de que os agentes dessa classe são semelhantes a *assistentes pessoais* da vida real, isto é, tentam auxiliar o usuário em uma determinada tarefa da forma como este gostaria. No caso da tarefa de seleção, que é o objetivo do agente construído nesta dissertação, vale dizer que essa característica faz com que o agente construído tente selecionar as mensagens de acordo com o “*óculos*” de cada professor, isto é, selecionar as mensagens adequadas aos interesses específicos de cada um deles. Essa característica é particularmente interessante porque professores de um curso à distância podem buscar informações diferentes uns dos outros ao analisarem uma sessão de bate-papo; enquanto um professor pode olhar para aspectos sociais da interação – mensagens do tipo ‘oi, fulano’, ‘como vai?’, ‘passou bem o final de semana?’ – outro pode olhar para aspectos específicos do conteúdo discutido – por exemplo, analisa para ver se foram discutidos todos os tópicos que havia planejado.

Com o intuito de conseguir essa personalização, o agente Chapa é capaz de construir dicionários dinâmicos de dados contendo informações sobre os interesses específicos de cada professor. Esses dicionários são dinâmicos porque durante toda a sua

existência vão sendo adicionadas e retiradas informações conforme o agente vai aprendendo quais as informações interessam ou não para um determinado professor.

O processo de aprendizagem do agente ocorre por meio das seguintes técnicas de aprendizagem características da classe de agentes de interface: aprendizagem por observação, via feedback do usuário e programação a partir de exemplos explícitos passados pelo usuário. Mais detalhes a respeito dessas técnicas serão descritas no capítulo 2, que trata exclusivamente sobre as características de agentes de interface. Particularmente, como o agente construído tem que lidar com texto escrito (conteúdo das mensagens de uma sessão de bate-papo) podendo este conter teoricamente qualquer palavra em qualquer contexto, foi excluída, sumariamente, a aplicação de uma análise das palavras das mensagens para a seleção daquelas mensagens a serem consideradas relevantes para um determinado professor e que, conseqüentemente, são adicionadas ao dicionário de dados do mesmo. Isso porque essa análise requer uma ampla análise sintática e semântica sendo, portanto, uma alternativa de solução altamente complexa de ser executada na prática.

A alternativa foi a utilização de um algoritmo de aprendizagem que fosse eficaz no estabelecimento de padrões nas mensagens textuais sem ter que, necessariamente, analisar o conteúdo sintático e semântico das mensagens. Na literatura existem diversos algoritmos que se propõem a essa tarefa, indo desde os que aplicam árvores de decisão até os que se utilizam de sofisticadas redes neurais. No agente construído foi implementado um algoritmo Bayesiano de classificação simples que utiliza princípios da teoria Bayesiana de probabilidades para o seu funcionamento, daí resultando o nome do algoritmo. Os motivos pelos quais esse algoritmo foi o escolhido e maiores detalhes a respeito do mesmo são apresentados no Capítulo 3.

Concluindo este capítulo de introdução, é apresentada agora a organização desta dissertação: no Capítulo 2 são apresentadas a definição de agentes de software e suas classificações, entrando em maiores detalhes na definição da classe de agentes de interface,

suas principais características e técnicas de aprendizagem; no Capítulo 3 é apresentada a definição e objetivo das técnicas de aprendizagem de máquina, descrevendo em detalhes a teoria que dá embasamento ao algoritmo de aprendizagem implementado no agente construído e o algoritmo implementado propriamente dito; no Capítulo 4 são apresentadas a interface do agente dentro do ambiente TelEduc e todas as suas funcionalidades; no Capítulo 5 são apresentadas as conclusões finais; no Capítulo 6 são apresentadas as referências bibliográficas utilizadas na escrita desta dissertação.

Capítulo 2

Agentes de software

2.1 Definição

A expectativa de se chegar a um consenso absoluto sobre a definição do termo “agente” é a mesma que os pesquisadores de inteligência artificial têm de chegar a um consenso sobre o conceito de “inteligência artificial”, ou seja, nenhuma (NWANA, 1996). Isso porque o conceito de “agente” abrange um campo muito amplo de pesquisa e desenvolvimento. Existem agentes no mundo físico (robôs de fábricas), em vastas redes de computadores (robôs de software), existem agentes que realizam tarefas específicas, que são capazes de aprender (assistentes pessoais, por exemplo) e diversos outros com as mais variadas finalidades. Conseqüentemente, existem na literatura diversas definições do termo agente elaboradas com base nas idéias que os principais pesquisadores da área têm da aplicação final do mesmo.

Jennings e Wooldridge (1995) definem um agente como sendo um sistema de computador situado em algum ambiente e que é capaz de ações autônomas nesse ambiente, procurando sempre alcançar seus objetivos. Essa autonomia da qual deve estar imbuído o agente é descrita como sendo a sua capacidade de agir sem necessitar da intervenção humana ou de outros agentes e de ter o controle das suas ações e estados internos. Como

exemplos, Jennings e Wooldridge citam um termostato e um sistema de controle de reatores nucleares, uma vez que ambos são capazes de monitorar um ambiente no mundo real e realizar ações para modificá-lo conforme as condições do ambiente vão mudando.

Outra definição proposta, agora por Russel e Norvig (1995), segue essa mesma linha de raciocínio só que em uma forma mais geral. Segundo eles, "*um agente é qualquer coisa que percebe seu ambiente por meio de sensores e atua sobre este ambiente por meio de atuadores*". Com base nela, percebe-se que para um sistema ser classificado como um agente não é necessário que seja capaz de ações autônomas, somente é preciso que seja capaz de *sentir* o ambiente e de *atuar* sobre ele.

Já Franklin e Graesser (1996) e Shoham (1997 apud BRADSHAW, 1997) vão no sentido contrário de Russel e Norvig (1995). Ao invés de generalizarem o conceito do termo agente apresentado por Jennings e Wooldridge (1995), procuram caracterizá-lo como sendo um sistema que deve, além de ser autônomo e ser capaz de sentir/atuar sobre um ambiente, possuir o conceito de *continuidade temporal*. Por exemplo, um simples programa de folha de pagamento não é considerado um agente segundo a concepção de Franklin e Graesser (1996). Isso porque apesar desse programa agir em um ambiente (mundo real), sentir o mundo por meio de sua entrada e agir sobre ela por meio de sua saída, não é considerado um agente, pois não passa no teste da continuidade temporal, ou seja, roda uma vez e então entra em "coma", esperando ser chamado novamente.

Foner (1993) considera que, além da capacidade de realizar ações autônomas, um sistema só deve ser classificado como sendo um agente se suportar alguma forma de diálogo com o usuário e for capaz de se degradar gradualmente com o tempo. Um exemplo que cita para ilustrar essa sua concepção é o de um programa coletor de lixo (*garbage collector*) feito em Lisp disponível em seu ambiente computacional. Segundo ele, esse programa não se enquadra na sua definição de agente porque, embora realize ações

espontâneas para manter o ambiente computacional limpo, não suporta nenhuma forma de diálogo.

Outra pesquisadora importante na área, Maes (1998), tenta captar a essência do que são agentes observando que eles são, fundamentalmente, programas que diferem do software comum pelo fato de que podem ser descritos como *entidades independentes*. À semelhança de Jennings e Wooldridge (1995), os agentes, para Maes, devem ser autônomos, ou seja, serem capazes de sentir o estado atual do ambiente onde se encontram, saberem quais os seus objetivos e agirem independentemente com o intuito de tentar alcançá-los. Além disso, também devem ser robustos e adaptáveis, capazes de aprender a partir de experiências passadas e de responderem a situações não previstas com um repertório de métodos diferentes.

Para Negroponte (1997 apud BRADSHAW, 1997), agentes são programas que possuem um grande conhecimento sobre uma certa tarefa e sobre as peculiaridades dos seus usuários. Ele cita a característica comum entre agentes de software e agentes humanos como sendo o ponto-chave na sua definição de um agente:

(...)Um bom agente de viagens combina conhecimentos sobre hotéis e restaurantes com conhecimentos sobre você, um bom corretor de imóveis, constrói um modelo do cliente a partir de uma sucessão de imóveis que tenham agradado o cliente com diferentes níveis de sucesso. Agora imagine um agente de notícias ou de gerenciamento de correio eletrônico. O que todos eles têm em comum é a habilidade de modelar você. (NEGROPONTE, 1997 apud BRADSHAW, 1997, p.6)

A partir das diversas definições apresentadas, nota-se que não existe uma definição precisa de um agente de software; o que se pode observar é que o termo agente de software é um *meta-termo* que abrange todos os outros tipos mais específicos de agentes. Por isso,

uma das melhores formas de *abordar* o termo agente de software não é tentando defini-lo, mas sim, caracterizando as propriedades/atributos que se espera que possua.

Jennings e Wooldridge (1995) e Etzioni e Weld (1995 apud BRADSHAW, 1997) apresentam o seguinte conjunto de atributos que esperam que um sistema apresente, em maior ou menor grau, para poder ser caracterizado como um agente de software:

- **Reatividade:** capacidade de perceber seu ambiente e responder às mudanças que ocorrerem nele;
- **Mobilidade:** capacidade de se locomover pela rede, de forma auto-dirigida;
- **Autonomia:** habilidade do agente de operar por conta própria, sem um ser humano por trás guiando suas ações. Um elemento chave para a sua autonomia é a proatividade, isto é, a habilidade de “tomar a iniciativa” ao invés de simplesmente agir em resposta ao ambiente.
- **Capacidade de aprendizagem:** capacidade de aprender e melhorar com a experiência;
- **Comportamento Colaborativo:** trabalham em conjunto com outros agentes para alcançarem um objetivo comum.

Na próxima subseção é apresentada uma classificação para os agentes de software construída com base nessas diversas propriedades.

2.2 Classificação

Com base nas propriedades apresentadas na subseção anterior, Nwana (1996) apresenta a seguinte classificação dos agentes de software:

- **Deliberativos/reativos:** agentes deliberativos derivam do paradigma do pensamento deliberativo, ou seja, possuem um modelo interno de raciocínio para decidirem que ações devem realizar. Eles exibem habilidades de planejamento e negociação quando em contato com outros agentes (humanos ou de software) sempre procurando fazer progressos no sentido de atingirem os objetivos para os quais foram projetados. Ao contrário destes, os agentes reativos não possuem um modelo interno de raciocínio, ao invés disso agem sobre o ambiente utilizando um comportamento do tipo estímulo/resposta (FERBER, 1994).
- **Móveis/estáticos:** os agentes são classificados dessa forma se apresentam ou não a propriedade de locomover-se entre servidores (*mobilidade*).
- **Colaborativos / aprendizes colaborativos / Interface / Inteligentes:** os agentes são classificados nessas quatro categorias de acordo com a maior ênfase que apresentam nos seguintes quesitos: *cooperação*, *capacidade de aprendizagem* e *autonomia*. Aqueles agentes que foram projetados para serem mais capazes de cooperar com outros agentes e de serem autônomos na realização de suas ações são classificados como *colaborativos*. Aqueles que são notadamente autônomos e que conseguem aprender com o passar do tempo são classificados como *agentes de interface*. Já os *agentes aprendizes colaborativos* são aqueles que possuem, como características principais, a capacidade de cooperar com outros agentes e de aprender. Por fim, aqueles agentes que possuem desenvolvidos os três quesitos mencionados são classificados como *agentes inteligentes*. Na Figura 2.1 é apresentada uma representação esquemática dessas quatro classes de agentes de software. É importante frisar, com relação a Figura 2.1, que esta somente indica as características mais comuns de uma determinada classificação de agentes, não implicando obrigatoriamente que agentes pertencentes a uma determinada classe não possam apresentar características de uma outra classe. Por exemplo, as características mais comuns de agentes de interface são a autonomia e

capacidade de aprendizagem. Isso, no entanto, não significa que os agentes dessa classe não possam cooperar uns com os outros.

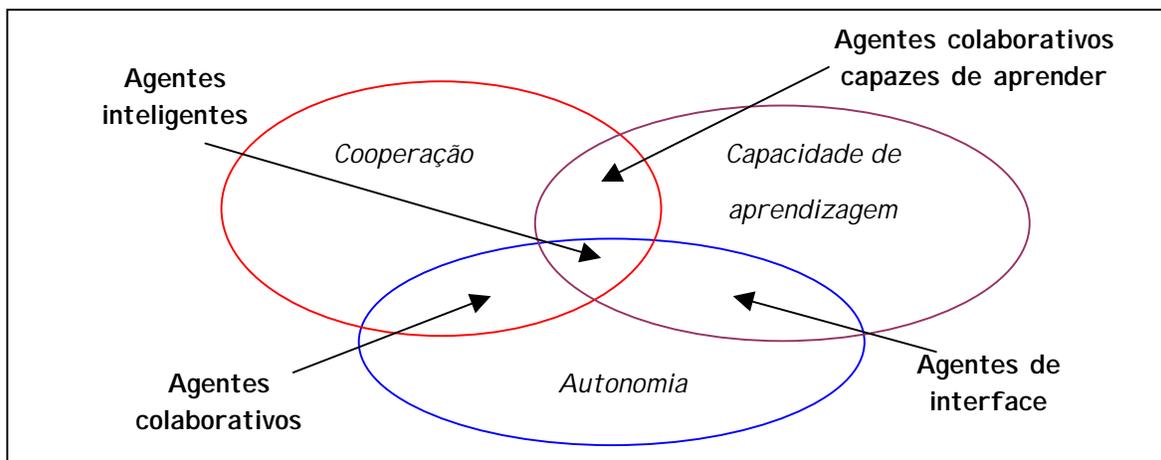


Figura 2.1 - Caracterização esquemática das quatro categorias de agentes apresentadas com base nas características de *Cooperação*, *Capacidade de Aprendizagem* e *Autonomia*.

- **Informação:** os agentes também são classificados de acordo com os nichos onde atuam (preferencialmente, se são importantes). Um exemplo seriam os agentes de informação da Internet. Exemplos desta categoria de agentes são os agentes de busca utilizados em páginas como o *Cadê*¹ e *Achei!*².
- **Híbridos:** são agentes que combinam características de duas ou mais das classes apresentadas anteriormente.

A Figura 2.2 apresenta uma representação esquemática de todas as classes de agentes proposta por Nwana(1996).

¹ <http://www.cade.com.br>

² <http://www.achei.com.br>

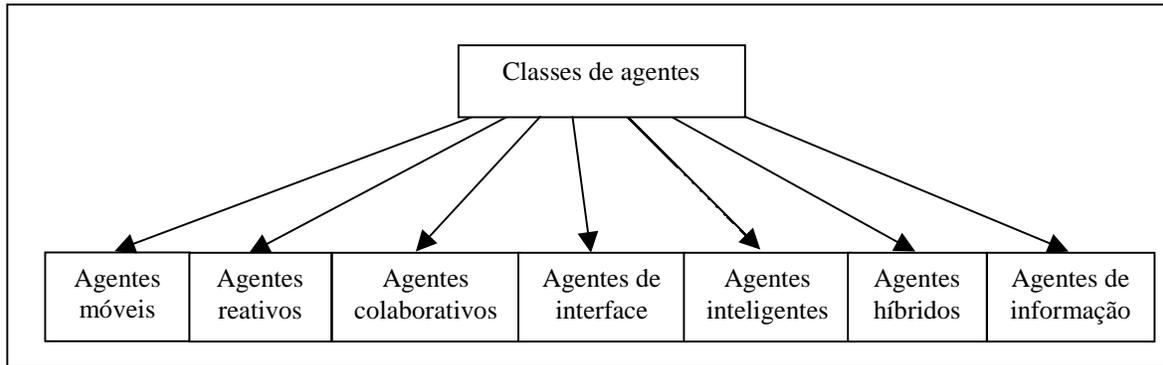


Figura 2.2 - Classificação dos diferentes tipos de agentes

Como o agente desenvolvido nesta dissertação pertence à classe dos agentes de interface são apresentadas, na próxima subseção, as abordagens existentes para a construção de um agente de interface e os problemas envolvidos nesta tarefa.

2.3 Construção de Agentes de Interface

Negroponete (1970, apud MAES, 1998) e Kay (1990) foram os primeiros que vislumbraram a idéia de se aplicar agentes na interface com o intuito de diminuir a execução de tarefas repetitivas por parte do ser humano deixando-as ao encargo do computador. Com o passar do tempo, vários fabricantes de computadores adotaram essa idéia e um grande esforço vêm sendo feito, desde então, no sentido de modelar e construir agentes de interface. No entanto, as técnicas existentes atualmente ainda estão longe de permitirem a construção de um agente de interface capaz de executar tarefas tão bem quanto um ser humano.

Maes (1998), uma das mais importantes pesquisadoras na área de agentes de interface, aponta dois problemas principais que têm que ser solucionados quando da construção de um agente de interface: o da *competência do agente* e o da *confiança do*

usuário no agente. O primeiro problema, o da *competência*, é decorrência direta da seguinte pergunta: “*Como um agente adquire o conhecimento necessário para decidir como, quando e em que deve ajudar o usuário ?*”. Já o segundo problema, o da *confiança*, relaciona-se ao fato de como fazer com que o usuário se sinta confortável delegando tarefas para um agente de interface.

Nas próximas subseções são descritas as abordagens, citadas por Maes (1998), utilizadas na construção de agentes de interface e de como essas abordagens procuram lidar com os problemas da competência/confiança apresentados. Além disso, é apresentada uma conclusão onde é feita uma consideração a respeito das abordagens apresentadas e de qual técnica foi escolhida para o desenvolvimento do agente Chapa implementado neste trabalho.

2.3.1 Abordagem 1

A primeira abordagem mencionada por Maes (1998) consiste em fazer *do próprio programa final do usuário um agente de interface*. Um exemplo é o sistema construído por Lai et al. (1988), denominado Oval. Esse sistema é constituído por “agentes semi-autônomos”, que nada mais são do que um conjunto de regras programadas pelo próprio usuário do sistema, com o objetivo de realizar uma determinada tarefa. Por exemplo, o usuário do *Oval* pode criar um agente para a organização das suas mensagens de correio eletrônico por meio da criação de regras que processem as mensagens que vão chegando e as coloquem nas pastas apropriadas. Uma vez criadas, essas regras realizam essa tarefa para o usuário sem que este tenha que invocá-las explicitamente.

O principal problema desta abordagem é que ela não lida com o critério da competência de forma satisfatória uma vez que requer muito conhecimento e esforço por parte do usuário. Isso porque o próprio usuário tem que reconhecer a oportunidade de aplicar o agente, tomar a iniciativa de criá-lo e passar explicitamente o conhecimento necessário ao agente para o desempenho da tarefa. Além disso, também deve atualizar

continuamente as regras do agente para que estas sempre estejam de acordo com as suas mudanças de hábitos e/ou interesses.

Já o quesito *confiança* não é quase problema neste tipo de abordagem dado que o usuário confia nas suas próprias habilidades de programação. Mas, mesmo assim, podem surgir alguns inconvenientes, uma vez que é comum os programas escritos por um humano tipicamente se comportarem diferentemente do que é esperado, mesmo quando se confia nas próprias habilidades de programação.

2.3.2 Abordagem 2

A segunda abordagem denominada de “*abordagem baseada em conhecimento*” consiste no fornecimento ao agente de interface, *a priori* da sua utilização, de um amplo conhecimento sobre o domínio e o usuário da aplicação (denominados *modelo do domínio* e *modelo do usuário*, respectivamente). Esta abordagem é adotada pela grande maioria das pessoas que trabalham com Inteligência Artificial aplicada a interfaces inteligentes (SULLIVAN, 1991 apud MAES, 1998). No momento da execução o agente de interface utiliza seus conhecimentos para reconhecer os planos do usuário e para achar oportunidades de contribuir para a realização deles.

Um exemplo de um agente de interface que utiliza essa abordagem é o UCEgo (CHIN, 1991). Este é um agente que foi construído para ajudar um usuário a solucionar problemas na utilização do sistema operacional Unix. O UCEgo possui uma vasta base de conhecimentos sobre como usar o Unix e também uma série de tarefas passíveis de serem executadas pelo usuário no sistema operacional. A partir dessa sua base de conhecimentos, oferece-se para auxiliar o usuário a utilizar o Unix.

Nesta *abordagem baseada em conhecimento*, tanto a competência quanto a confiança constituem problemas sérios. O primeiro problema relaciona-se ao quesito competência do agente. Nesta abordagem é necessário um enorme trabalho por parte de um

engenheiro de conhecimento, para transmitir o maior conjunto possível de conhecimentos relacionados ao domínio da aplicação e à própria aplicação para a base de conhecimentos do agente. Mesmo assim, pouco deste conhecimento ou da arquitetura de controle passada ao agente pode ser reutilizada quando da construção de agentes para outras aplicações. O segundo problema, também relacionado à competência do agente, está ligado ao fato de que o conhecimento do agente, uma vez definido pelo engenheiro do conhecimento, continuará o mesmo para sempre, não podendo ser personalizado para as preferências e hábitos individuais de cada usuário. Além disso, outro ponto questionável dessa abordagem é com relação a possibilidade de se conseguir fornecer ao agente todo o conhecimento que vai necessitar para sempre ser capaz de compreender as ações, algumas vezes imprevisíveis, do usuário.

Além dos problemas já apresentados relativos à competência do agente, também há problema relacionados à confiança do usuário no agente neste tipo de abordagem. Provavelmente, não é uma idéia muito boa dar ao usuário um agente de interface que seja muito sofisticado, qualificado e autônomo logo de começo. Schneiderman argumentou, convincentemente, que tal agente poderia dar ao usuário a *sensação de perda do controle e do conhecimento sobre o que está acontecendo na aplicação* (MYERS et al., 1991 apud MAES, 1998). Isso decorre diretamente do fato de que, como o agente foi programado por outra pessoa, o usuário acaba não tendo um bom modelo das limitações e formas de trabalho do agente.

2.3.3 Abordagem 3

Maes (1998) propõe uma abordagem alternativa baseada na *aprendizagem*, para a construção de agentes de interface utilizando técnicas de *aprendizagem de máquina*. A hipótese que assume é que, sob certas condições, um agente de interface pode “*programar a si mesmo*”, ou seja, é capaz de adquirir, com o tempo, o conhecimento que necessita para auxiliar seus usuários. Para isso, é dado ao agente um conjunto inicial mínimo de conhecimentos e, com base neste conjunto inicial, mais os conhecimentos que adquirir com

o seu usuário, ou por meio de outros agentes, deveria ser capaz de aprender a realizar suas tarefas de forma apropriada. As condições que Maes (1998) menciona que têm que ser preenchidas para que a sua hipótese seja válida são:

- *Uso da aplicação tem que envolver um comportamento substancialmente repetitivo* porque, senão, o agente não será capaz de aprender nada, uma vez que não há nenhuma regularidade no uso da aplicação a ser aprendida.
- *Comportamento repetitivo no uso da aplicação é, potencialmente, diferente para diferentes usuários.* Caso isso não tenha possibilidades de acontecer, ou seja, o comportamento apresentado por diferentes usuários no uso da aplicação é sempre o mesmo, então a *abordagem baseada em conhecimento*, comprovadamente, é capaz de levar a resultados mais rápidos que uma abordagem baseada em aprendizagem.

Segundo Maes, esta abordagem foi inspirada na metáfora de um *assistente pessoal*:

Inicialmente, um assistente pessoal não está muito familiarizado com os hábitos e preferências do seu empregador e acaba não sendo muito útil no começo. Isso porque o assistente necessita de algum tempo para se familiarizar com os métodos específicos de trabalho do chefe e da organização. Com o passar do tempo, a partir de cada experiência aprendida pelo assistente, seja por meio de instruções específicas ou da simples observação do seu chefe realizando tarefas, seja a partir da aprendizagem com outros assistentes mais experientes da organização, o assistente passa, gradualmente, a ser capaz de realizar as tarefas que eram executadas anteriormente pelo seu chefe. (MAES, 1998, p.527).

O objetivo desta abordagem proposta por Maes (1998) é, justamente, permitir a construção de um agente de interface que se torne cada vez mais prestativo e competente, da mesma forma que um assistente pessoal na vida real.

De acordo com Maes (1998), um agente de interface construído dessa forma apresenta uma solução satisfatória para o problema da confiança. Isso porque o agente vai desenvolvendo suas habilidades gradualmente, dando tempo ao usuário para construir, gradualmente, um modelo de como o agente toma suas decisões. Esse tempo que o usuário acaba tendo para entender, ele próprio, como o agente age é, justamente, um dos pontos necessários para que se estabeleça uma relação de confiança entre o usuário e o agente. Outro ponto importante para o estabelecimento dessa confiança é o fato de que esta abordagem permite ao agente dar “explicações” sobre o seu comportamento e sobre a forma como se deu o seu raciocínio, por meio da apresentação de casos passados similares à situação atual na qual está agindo.

Com relação ao aspecto competência, Maes (1998) cita que um agente de interface construído por meio desta abordagem tem quatro fontes de onde adquirir competência para desempenhar suas tarefas.

Na primeira delas, o agente de interface aprende *olhando continuamente sobre o ombro do usuário enquanto este está realizando ações*. O agente de interface monitora as atividades do usuário, mantém um histórico de todas as ações dele durante longo períodos de tempo (semanas ou meses), acha regularidade e padrões recorrentes e se oferece para automatizá-los. Por exemplo, se um agente de busca de notícias detecta algum padrão nos artigos lidos pelo usuário, pode oferecer artigos similares a ele quando os encontrar.

A segunda fonte de aprendizagem é via um *feedback direto* ou *indireto* do usuário. O *feedback indireto* acontece no momento em que o usuário não leva em conta uma sugestão do agente e toma uma ação diferente no lugar. Um exemplo de feedback indireto ocorre quando um agente projetado para sugerir artigos para seu usuário, sugere um artigo para a leitura por parte do seu usuário mas este não o lê ou, no caso contrário, quando o usuário lê algum artigo que não foi sugerido pelo agente. Já no *feedback direto*, o usuário explicita diretamente para o agente o que achou das ações automatizadas por este, é como

se dissesse para o agente coisas do tipo: “não faça *isso* de novo” ou “eu não gosto *deste* artigo”.

A terceira fonte de aprendizagem é quando o agente aprende a partir de exemplos dados explicitamente pelo usuário. O usuário pode treinar o agente dando a ele exemplos hipotéticos de eventos e situações dizendo ao agente o que fazer nesses casos. O agente de interface grava estas ações, relaciona-as e muda a sua base de exemplos para incorporar o exemplo que é apresentado. Por exemplo, o usuário pode ensinar a um agente de correio eletrônico a salvar todas as mensagens enviadas por uma determinada pessoa em uma pasta específica. Para isso, basta simplesmente apresentar ao agente um exemplo hipotético de uma mensagem chegando (com todos os campos não-preenchidos menos o campo do remetente) e dele, usuário, arrastando esta mensagem para uma pasta. O agente aprenderia esse comportamento do usuário e passaria a automatizá-lo.

A quarta fonte de aprendizagem passível de ser usada por um agente de interface para adquirir competência é por meio de pedidos de conselhos a outros agentes que auxiliam outros usuários na mesma tarefa. Se um agente não sabe qual ação é apropriada em uma determinada situação, pode apresentá-la a outros agentes e perguntar qual a ação que eles recomendam para aquela situação. Por exemplo, se uma mensagem de correio eletrônico enviada por uma pessoa X chega, então o agente de e-mail pode perguntar a outros agentes o que fazer com aquela mensagem. Se a maioria dos outros agentes recomendar que aquela mensagem tem uma prioridade alta e deve ser apresentada ao usuário para leitura na hora, então o agente pode oferecer esta recomendação ao seu usuário, mesmo que o agente nunca tenha observado anteriormente seu usuário lidar com mensagens da pessoa X. Na Figura 2.3 é apresentada uma representação esquemática das quatro fontes de aprendizagem passíveis de serem utilizadas por um agente de interface para adquirir conhecimento nesta abordagem proposta por Maes (1998).

Finalizando, segundo Maes, além de apresentar boas alternativas para os problemas de confiança e competência dos agentes de interface, essa abordagem de aprendizagem também apresenta vantagens em relação às outras nos seguintes aspectos:

- requer menos trabalho do ponto de vista do usuário final e do projetista da aplicação;
- agente pode se adaptar ao usuário com o passar do tempo e se tornar personalizado aos hábitos e preferências individuais do mesmo;
- esta abordagem facilita a transferência de informação, hábitos e “*know-how*” entre os diferentes usuários de uma comunidade.

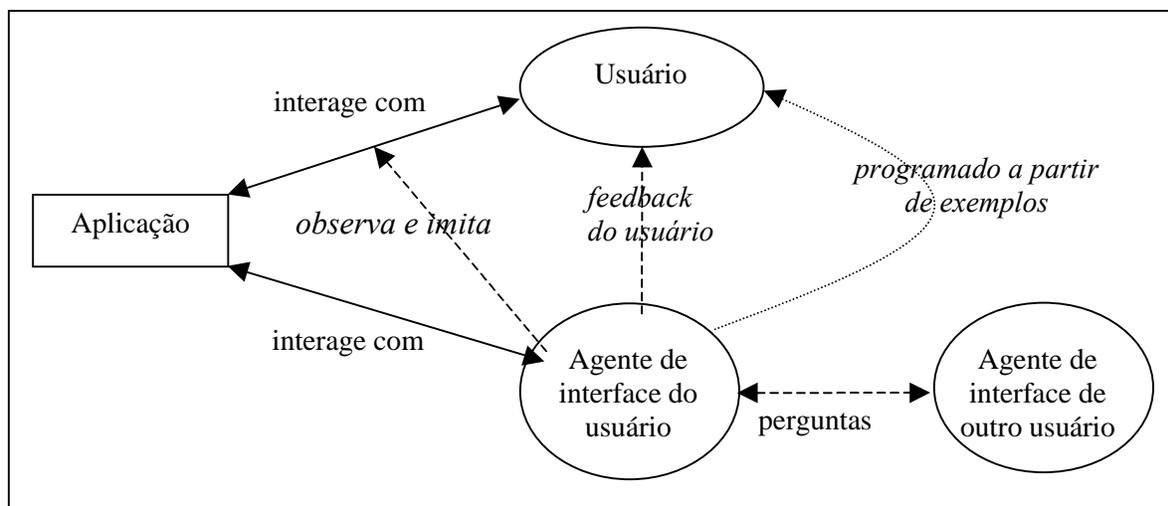


Figura 2.3 - Apresentação das quatro fontes de aprendizagem de um agente de interface propostas por Maes (1998)

2.4 Considerações finais

Como foi apresentado nas seções anteriores, as duas primeiras abordagens, a que tenta fazer *do próprio programa final do usuário um agente de interface* e a que é *baseada em conhecimento*, não lidam bem com os problemas da competência e da confiança intrínsecos à construção de um agente de interface. Além disso, ambas requerem um grande conhecimento para o desenvolvimento de um agente.

Na primeira abordagem, o conhecimento que se faz necessário é por parte do usuário, uma vez que o *obriga* a possuir conhecimentos profundos sobre o funcionamento da aplicação e do modo como o agente atua, de forma a conseguir utilizá-lo no auxílio à realização de suas tarefas.

Na segunda abordagem, é necessário um grande esforço por parte do engenheiro de conhecimento responsável por armazenar na base de conhecimentos do agente de interface todas as informações necessárias para que este saiba como agir em todas as situações, que porventura venha a encontrar, durante a realização de suas tarefas.

Como o problema tratado nesta dissertação é o da construção de um agente de interface capaz de selecionar as mensagens relevantes de uma sessão de bate-papo de acordo com os interesses de cada um dos participantes de um curso à distância, essas duas abordagens, devido a essas limitações apresentadas, não permitem solucionar adequadamente este problema.

A primeira abordagem não é adequada simplesmente porque a idéia de se construir um agente de interface é, justamente, diminuir ao máximo o esforço necessário por parte do usuário na realização de uma tarefa. No entanto, essa abordagem parte justamente do princípio contrário: deixa praticamente todo o trabalho nas mãos do usuário! Isso é,

inclusive, uma das causas que leva grande parte dos pesquisadores da área a não optarem por essa abordagem na hora de construir um agente de interface.

Já o problema da segunda abordagem para a solução do problema apresentado nesta dissertação, tem mais a ver com a inviabilidade de se conseguir fornecer ao agente uma base de conhecimentos suficientemente completa de forma a deixá-lo apto a selecionar as mensagens relevantes de qualquer sessão de bate-papo passível de existir registrada no ambiente TelEduc. Isso porque, como é possível que seja ministrado dentro do ambiente TelEduc um curso de qualquer uma das áreas do conhecimento, seria necessário que fosse passado para o agente o significado de *todas as palavras em todos os contextos possíveis* de forma a que ele pudesse atuar eficazmente selecionando as mensagens relevantes; ou então, que fossem fornecidas regras que permitissem ao agente deduzir o significado de qualquer palavra em qualquer contexto (análise semântica). No entanto, ambas as alternativas são de solução altamente complexa de serem executadas na prática.

Tendo em vista esses fatos, optou-se pela construção de um agente de interface utilizando-se a abordagem proposta por Maes (1998). Isso porque além dessa abordagem lidar com os problemas de competência e de confiança, como já foi apresentado anteriormente, permite ao agente adquirir e *aumentar* o seu conhecimento *com o passar do tempo*, ponto fundamental no problema de seleção das mensagens relevantes de uma sessão de bate-papo, uma vez que as palavras e os contextos das mensagens variam de acordo com o curso em que o agente estiver atuando. Um ponto interessante dessa abordagem é a possibilidade do agente se adaptar aos hábitos e preferências do usuário; isso quer dizer que um agente de interface construído para o problema da seleção de mensagens será capaz de selecionar um conjunto de mensagens *mais adequadas aos interesses individuais* de cada um dos participantes de um curso à distância.

Com base nessa escolha da abordagem proposta por Maes (1998) são apresentados no próximo capítulo o campo de aprendizagem de máquina e a técnica de aprendizagem

implementada no Chapa que permite a ele aprender e melhorar automaticamente sua performance com o passar do tempo e o aumento de sua experiência.

Capítulo 3

Aprendizagem de Máquina

3.1 Introdução

O campo de aprendizagem de máquina procura atacar o seguinte problema: como construir programas de computadores capazes de melhorar automaticamente o seu desempenho na realização de uma determinada tarefa com o passar do tempo, por meio do ganho de experiência.

No intuito de resolver este problema foram desenvolvidas várias técnicas, denominadas *técnicas de aprendizagem de máquina*, baseadas em conceitos e resultados das mais variadas áreas, incluindo Estatística, Inteligência Artificial, Biologia, entre outras. Muitas das técnicas desenvolvidas procuram fazer com que um programa seja capaz de aprender uma função, qualificada como *função-objetivo*, que modele adequadamente a tarefa para a qual foi projetado e implementado. Esta função é construída a partir de um conjunto inicial de exemplos e de situações, comumente denominados *dados de treinamento*, junto com as respectivas ações/decisões que o programa deve tomar com relação a eles(as). Além de englobar os dados de treinamento iniciais, a função-objetivo a ser aprendida pelo programa deve permitir que ela faça inferências sobre como agir com relação a situações futuras que ainda não tenham sido observadas anteriormente.

Atualmente, há diversas aplicações, nas mais variadas áreas, que utilizam com sucesso técnicas de aprendizagem de máquina com as mais diferentes finalidades. Essas aplicações vão desde programas de mineração de dados capazes de aprender como detectar transações fraudulentas com cartões de créditos até sistemas de filtragem de informações que aprendem as preferências de leitura de um usuário.

Apesar de terem as mais variadas aplicações, Mitchell (1997), classificou as diferentes técnicas existentes de aprendizagem de máquina de acordo com os seguintes paradigmas de aprendizagem: o *paradigma de aprendizagem indutiva* e o *paradigma de aprendizagem analítica*.

No paradigma de aprendizagem indutiva, de um modo geral, a forma como o programa aprende o comportamento de um usuário ocorre da seguinte maneira: o programa procura identificar as características que distinguem exemplos positivos de exemplos negativos – observando um conjunto inicial de dados de treinamento – para, a partir da identificação empírica de padrões nesse conjunto de exemplos, poder fazer generalizações do comportamento do usuário na execução de uma determinada tarefa. Como exemplos de técnicas que se enquadram dentro deste paradigma de aprendizagem têm-se: as redes neurais (FREEMAN e SKAPURA, 1991; BISHOP, 1996), árvores de decisão (MITCHELL, 1997), aprendizagem Bayesiana (MITCHELL, 1997), algoritmos genéticos (BOOKER et al., 1989), regras proposicionais (LAVRAC e DZEROSKI, 1994) etc.

No paradigma de aprendizagem analítica, os algoritmos de aprendizagem desenvolvidos aceitam explicitamente como entrada um conhecimento sobre o *domínio do problema*, isto é, além de identificarem padrões sobre um conjunto de dados de treinamento inicial, à semelhança do que é feito pelos algoritmos de aprendizagem indutiva, também se utilizam de um conhecimento sobre o próprio problema para o qual foram desenvolvidos. Um exemplo de uma técnica que se baseia neste paradigma de aprendizagem é a técnica de *aprendizagem baseada em explicações*. Nesta técnica utiliza-se um conhecimento existente

a priori sobre o problema de que se está tratando, para analisar e explicar, para cada exemplo de treinamento, quais características do exemplo são relevantes e quais são irrelevantes, de forma a poder inferir e generalizar padrões sobre os exemplos (MITCHELL, 1997).

Como pôde ser visto, há inúmeras técnicas de aprendizagem de máquina, no entanto, a adequação e efetividade de cada uma delas dependem diretamente da finalidade à qual serão aplicadas. Michie et al. (1994) fez um estudo comparativo detalhado entre um algoritmo de classificação simples baseado na técnica de aprendizagem Bayesiana e diversos outros algoritmos que utilizavam outras técnicas de aprendizagem, tais como árvores de decisão e redes neurais artificiais. Esses pesquisadores mostram na sua análise que, em muitos casos, o algoritmo Bayesiano apresentado no estudo tem um desempenho equivalente aos outros algoritmos de aprendizagem. Além disso, no estudo também é demonstrado que, para determinadas aplicações, a abordagem Bayesiana supera o desempenho dos outros algoritmos; em particular, para problemas que envolvam a aprendizagem de como se classificar documentos textuais (exemplo, artigos eletrônicos de notícias).

Como foi apresentado no parágrafo anterior, a abordagem Bayesiana é descrita na literatura como uma técnica de aprendizagem de máquina eficiente para se lidar com documentos textuais. E, tendo em vista que o agente proposto nesta dissertação tem que lidar, fundamentalmente, com texto escrito³, o algoritmo de aprendizagem selecionado para ser implementado é baseado em um algoritmo de aprendizagem Bayesiano, adaptado para lidar com as mensagens postadas pelos participantes de uma sessão de bate-papo do ambiente TelEduc.

³ Embora o texto de uma mensagem de uma sessão de bate-papo seja escrito de forma informal e livre, ainda assim é composto, unicamente, de texto escrito.

Nas próximas seções são apresentados: a Teoria Bayesiana, focando particularmente o teorema de Bayes, que é o ponto chave da técnica de aprendizagem Bayesiana; um método de classificação bayesiano desenvolvido com base no teorema de Bayes; a construção de um algoritmo geral de classificação de documentos textuais a partir desse método e as conclusões finais deste capítulo.

3.2 Teoria Bayesiana

A técnica de aprendizagem Bayesiana é denominada dessa forma por ter como base a *teoria Bayesiana de probabilidades*. Esta teoria baseia-se na suposição de que a quantidade de interesse do usuário sobre algum assunto é governada por *funções de distribuições de probabilidades*⁴ e que decisões ótimas podem ser tomadas a partir da análise das probabilidades calculadas sobre um conjunto de dados previamente observado.

Segundo Mitchell (1997), o ponto fundamental da teoria Bayesiana utilizada pelos métodos que aplicam esta técnica de aprendizagem de máquina é o seguinte teorema de Bayes:

$$P(h/D) = \frac{P(D/h) P(h)}{P(D)} \quad (3.1)$$

Onde:

h = hipótese que está sendo avaliada

D = dados de treinamento

⁴ De forma simplificada, define-se uma *função de distribuição de probabilidades* como sendo aquela que especifica, para cada valor possível y_i de uma variável aleatória Y , uma probabilidade $\Pr(Y = y_i)$, de que Y assumirá o valor y_i (Casella e Berger, 1990).

O teorema de Bayes permite calcular de forma direta a probabilidade de ocorrência de uma hipótese h , a partir de um conjunto inicial de dados de treinamento D observados anteriormente. Essa probabilidade, $P(h/D)$, é calculada no teorema a partir de três outras probabilidades:

- *Probabilidade $P(h)$* : essa probabilidade reflete o conhecimento anterior⁵ (*background*) que se possa ter sobre a chance de que h seja uma hipótese válida;
- *Probabilidade $P(D/h)$* : representa a probabilidade de se observar dados de treinamento dentro de um conjunto onde uma hipótese h é válida;
- *Probabilidade $P(D)$* : representa o valor da probabilidade dos dados de treinamento, independente de qual hipótese h é válida.

Segundo (MITCHELL, 1997), uma simplificação do teorema 3.1 pode ser feita por meio da eliminação do termo $P(D)$, uma vez que este é uma constante igual a 1, independente de h . Feita essa simplificação, tem-se a seguinte expressão:

$$P(h/D) = P(D/h) P(h) \quad (3.2)$$

⁵ Se não existir nenhum conhecimento *a priori* sobre a hipótese h , então, normalmente, é atribuído o mesmo valor de $P(h)$ para cada hipótese candidata, de forma que não seja influenciada, nem a favor nem contra, a probabilidade final calculada pelo teorema para uma determinada hipótese.

Para um melhor entendimento do significado dos diferentes termos presentes no teorema de Bayes é descrito a seguir um caso simples de sua aplicação e do cálculo das diferentes probabilidades envolvidas.

Considere uma situação em que é necessário diagnosticar se um paciente tem ou não uma forma particular de câncer (MITCHELL, 1997). Suponha que os dados disponíveis são provenientes de testes, em que os dois únicos resultados possíveis são: (+) positivo e (-) negativo; além disso, sabe-se que apenas 0.8% da população apresenta esse tipo de câncer e que o teste de laboratório é um indicador impreciso da existência ou não dessa doença nos pacientes. Essa imprecisão decorre do fato de que em 98% dos casos testados, o resultado retorna um resultado positivo correto – acusa a doença em pacientes nos quais a doença está de fato presente – e em outros 97% dos casos retorna um resultado negativo correto – a doença não está presente e, de fato, o paciente não a possui. Nos outros casos, o teste retorna um resultado oposto ao estado correto do paciente.

A situação descrita pode ser resumida pelas seguintes probabilidades:

$$P(\text{câncer}) = 0.008$$

$$P(\text{NOT câncer}) = 0.992$$

$$P(+/\text{câncer}) = 0.98$$

$$P(-/\text{câncer}) = 0.02$$

$$P(+/\text{NOT câncer}) = 0.03$$

$$P(-/\text{NOT câncer}) = 0.97$$

Nota: a simbologia $P(x/y)$, indica a probabilidade da hipótese x ocorrer dado que a hipótese y ocorreu. Exemplo: $P(+/\text{câncer})$ representa a probabilidade de um paciente que tem câncer realmente apresentar um resultado positivo para a doença no teste de laboratório.

Suponha agora que seja observado um novo paciente para o qual o teste de laboratório tenha retornado um resultado positivo. Deve-se ou não diagnosticar o paciente como tendo a doença ?

Aplicando-se a equação 3.2 pode-se estimar quais são as probabilidades envolvidas nessa situação e obter a resposta para essa pergunta:

$$P(+/\text{câncer}) P(\text{câncer}) = 0.98 \times 0.008 = 0.0078$$

$$P(+/\text{NOT câncer}) P(\text{NOT câncer}) = 0.03 \times 0.992 = 0.0298$$

Normalizando essas probabilidades, isto é, recalculando-as de forma que a sua soma dê 100%, por meio do teorema de Bayes (3.1), temos:

$$P(\text{câncer}) = \frac{0.0078}{0.0078+0.0298} = 0.21 \text{ ou } \mathbf{21\%} \quad P(\text{NOT câncer}) = \frac{0.0298}{0.0078+0.0298} = 0.79 \text{ ou } \mathbf{79\%}$$

Com isso, foi estimado que a probabilidade do paciente *não* estar com câncer é de 79%, logo, esse paciente seria diagnosticado como não estando com câncer !

A partir do exemplo apresentado nota-se que o teorema de Bayes permite calcular de forma direta a probabilidade *a posteriori* de uma nova hipótese, baseando-se em um conjunto inicial de dados previamente observados. Com base nesse fato é que são construídos os algoritmos de aprendizagem de máquina que utilizam esta técnica: *calculam, usando o teorema de Bayes, a probabilidade para todas as novas hipóteses possíveis com as quais se defrontam e, a partir das hipóteses que calcularam como tendo as maiores probabilidades, tomam as suas decisões.*

Na próxima seção é apresentado um método de classificação simples mas efetivo (MITCHELL, 1997), que se baseia na teoria de Bayes exposta nesta subseção.

3.3 Classificador Bayesiano Simples

Este método de aprendizagem se aplica a tarefas onde cada instância x de um problema é descrita por uma tupla de valores de atributos (a_1, a_2, \dots, a_n) e a função objetivo $f(x)$ pode assumir qualquer valor de um conjunto finito V .

Neste método, a classificação de uma nova instância é feita por meio da atribuição de um valor mais provável, denominado v_{map} , com base nos valores dos atributos (a_1, a_2, \dots, a_n) que descrevem a instância. Em notação matemática sua representação é a seguinte:

$$v_{map} = \operatorname{argmax}^6 \{P(v_j|a_1, a_2, \dots, a_n)\} \quad \text{onde } v_j \in V$$

Com base no teorema de Bayes apresentado na subseção anterior essa expressão pode ser rescrita da seguinte forma:

$$\begin{aligned} v_{map} &= \operatorname{argmax} \left\{ \frac{P(a_1, a_2, \dots, a_n | v_j) P(v_j)}{P(a_1, a_2, \dots, a_n)} \right\} \\ &= \operatorname{argmax} \{P(a_1, a_2, \dots, a_n | v_j) P(v_j)\} \quad \text{onde } v_j \in V \end{aligned}$$

Dada essa equação 3.3, o próximo passo é tentar estimar os dois termos presentes nela, $P(a_1, a_2, \dots, a_n | v_j)$ e $P(v_j)$, com base nos dados de treinamento. A estimativa de cada termo $P(v_j)$ é simples, basta efetuar a contagem da frequência com que cada valor v_j ocorre nos dados de treinamento. Já a estimativa de cada um dos diferentes termos $P(a_1, a_2, \dots, a_n | v_j)$

⁶ Dado um conjunto de elementos $C = \{1, 2, 3, 4\}$; teremos que $\operatorname{argmax}(C) = 4$. Isto é, argmax é uma função que retorna o maior elemento de um conjunto de elementos

é possível mas não é viável de ser feita. Isso decorre do fato de que o número desses termos é igual ao número das possíveis instâncias do problema, multiplicado pelo número dos possíveis valores dos atributos, ou seja, é necessário o cálculo de um número muito grande de termos.

Tentando viabilizar o cálculo do termo $P(a_1, a_2, \dots, a_n | v_j)$, o algoritmo de classificação Bayesiano desta subseção utiliza-se da suposição de que os valores dos atributos são *condicionalmente independentes* uns dos outros, dado um valor objetivo. Em outras palavras, a suposição assumida é que, dado um valor objetivo da instância, a probabilidade de observação da conjunção a_1, a_2, \dots, a_n é, simplesmente, o produto das probabilidade dos atributos individuais, isto é, $P(a_1, a_2, \dots, a_n | v_j) = \prod_i P(a_i | v_j)$. Substituindo esse resultado na equação 3.3 obtém-se a equação 3.4 a seguir, que é a base do algoritmo de classificação Bayesiano desta subseção:

$$v_{nb} = \operatorname{argmax} \{P(v_j) \prod_i P(a_i | v_j)\} \quad (3.4)$$

Onde,

$v_j \in \mathcal{V}$ e v_{nb} denota o valor de saída objetivo do classificador Bayesiano simples.

Um ponto que deve ser notado na equação 3.4 é que o número de termos distintos $P(a_i | v_j)$, que devem ser estimados a partir dos dados de treinamento, é apenas o número de valores de atributos distintos, multiplicado pelo número de valores objetivos distintos – um número muito menor do que se fosse necessário estimar todos os termos $P(a_1, a_2, \dots, a_n | v_j)$ presentes na equação 3.3.

Nas próximas subsecções são apresentados: um exemplo ilustrativo da aplicação do classificador Bayesiano simples descrito nesta seção e o cuidado que se deve tomar na hora de aplicá-lo.

3.3.1 Exemplo ilustrativo

No exemplo desta subsecção é utilizada a equação 3.4 para o seguinte problema: classificar os dias de acordo com alguém que vai jogar tênis. A Tabela 3.1 apresenta um conjunto de 14 exemplos de treinamento do conceito *JogarTênis* em que, cada dia é descrito pelos atributos Clima, Temperatura, Umidade e Vento. Suponha agora que seja usado o classificador Bayesiano apresentado e os dados de treinamento da tabela 3.1 para classificar a seguinte nova instância:

(Clima = ensolarado; Temperatura = Agradável; Umidade = Alta; Vento = Forte)

| Dia | Clima | Temperatura | Umidade | Vento | JogarTênis |
|-----|------------|-------------|---------|-------|------------|
| 1 | Ensolarado | Quente | Alta | Fraco | Não |
| 2 | Ensolarado | Quente | Alta | Forte | Não |
| 3 | Nublado | Quente | Alta | Fraco | Sim |
| 4 | Chuvoso | Amena | Alta | Fraco | Sim |
| 5 | Chuvoso | Agradável | Normal | Fraco | Sim |
| 6 | Chuvoso | Agradável | Normal | Forte | Não |
| 7 | Nublado | Agradável | Normal | Forte | Sim |
| 8 | Ensolarado | Amena | Alta | Fraco | Não |
| 9 | Ensolarado | Agradável | Normal | Fraco | Sim |
| 10 | Chuvoso | Amena | Normal | Fraco | Sim |
| 11 | Ensolarado | Amena | Normal | Forte | Sim |
| 12 | Nublado | Amena | Alta | Forte | Sim |
| 13 | Nublado | Quente | Normal | Fraco | Sim |
| 14 | Chuvoso | Amena | Alta | Forte | Não |

Tabela 3.1. Conjunto de exemplos de treinamento para a aprendizagem do conceito *JogarTênis*. Tabela extraída de Mitchell (1997).

A tarefa é prever o valor objetivo (*Sim* ou *Não*) do conceito *JogarTênis* para esta nova instância. Para isso, instancia-se a equação 3.4 com os valores dos atributos da nova instância:

$$\begin{aligned} v_{nb} &= \operatorname{argmax} P(v_j) \prod_i P(a_i|v_j) \quad \text{onde } v_j \in [\textit{Sim}, \textit{Não}] \\ &= \operatorname{argmax} P(v_j) P(\textit{Clima} = \textit{ensolarado} | v_j) P(\textit{Temperatura} = \textit{Agradável} | v_j) \\ &\quad P(\textit{Umidade} = \textit{Alta} | v_j) P(\textit{Vento} = \textit{Forte} | v_j) \end{aligned}$$

Primeiramente, vamos calcular o valor de $P(v_j)$ na expressão. Este valor pode ser calculado facilmente com base nas frequências em que os valores-objetivos (*Sim* ou *Não*) aparecem nos exemplos de treinamento:

$$P(\textit{JogarTênis} = \textit{Sim}) = 9/14 = 0.64$$

$$P(\textit{JogarTênis} = \textit{Não}) = 5/14 = 0.36$$

Similarmente, calculam-se as probabilidades condicionais – termos $P(a_i|v_j)$ – presentes na expressão. Por exemplo, para o valor de atributo *Vento = Forte* o cálculo é feito do seguinte modo:

$$P(\textit{Vento} = \textit{Forte} | \textit{JogarTênis} = \textit{Sim}) = 3/9 = 0.33$$

$$P(\textit{Vento} = \textit{Forte} | \textit{JogarTênis} = \textit{Não}) = 3/5 = 0.60$$

Aplicando o mesmo tipo de cálculo para os outros valores de atributos (*Clima = Ensolarado*; *Temperatura = Agradável*; *Umidade = Alta*) estimam-se todas as outras probabilidades condicionais. Fazendo esses cálculos e substituindo os valores na equação 3.4 obtém-se o seguinte resultado:

$$P(\text{Sim}) P(\text{Ensolarado}|\text{Sim}) P(\text{Agradável}|\text{Sim}) P(\text{Alta}|\text{Sim}) P(\text{Forte}|\text{Sim}) = 0.0053$$

$$P(\text{Não}) P(\text{Ensolarado}|\text{Não}) P(\text{Agradável}|\text{Não}) P(\text{Alta}|\text{Não}) P(\text{Forte}|\text{Não}) = 0.0206$$

*nota: para simplificar a notação foram omitidos os nomes dos atributos. Por exemplo, ao invés de escrever $JogarTênis=Sim$, escreveu-se simplesmente, Sim .

A partir do resultado calculado o classificador Bayesiano atribuiria o valor $JogarTênis = Não$ para esta nova instância, com base na estimativa de probabilidade aprendida a partir do conjunto de exemplos de treinamento inicial que possui à disposição.

Na próxima subseção é apresentado um problema que pode surgir durante o cálculo dos termos da equação 3.4 e uma forma de contorná-lo.

3.3.2 Estimando probabilidades

No exemplo apresentado na subseção anterior verifica-se que as probabilidades são estimadas por meio do cálculo da fração de vezes que um evento acontece em um total de vezes possíveis. Por exemplo, estimou-se $P(\text{Vento} = \text{Forte}|\text{JogarTênis} = \text{Não})$ por meio da fração n_e/n onde, $n = 5$ é o número total de exemplos de treinamento onde $JogarTênis=Não$ e $n_e = 3$ é o número de vezes na qual ocorre o evento $Vento=Forte$ nesse conjunto de exemplos de treinamento onde $JogarTênis=Não$.

Normalmente, essa fração fornece uma boa estimativa da probabilidade de ocorrência do evento e . No entanto, nos casos em que o valor de n_e é muito pequeno, essa fração nos dá uma estimativa não muito boa. Imagine o caso em que, de fato, o valor de $P(\text{Vento} = \text{Forte}|\text{JogarTênis} = \text{Não})$ é 0.08 e que se tem uma amostra contendo somente 5 exemplos para os quais $JogarTênis=Não$. Com isso, o valor mais provável de n_e é 0. Como consequência, o valor desse termo – 0 (zero) – vai acabar sempre dominando o produto dos termos calculado pelo classificador Bayesiano se, no futuro, for feita uma consulta que contenha $Vento=Forte$.

Uma forma de evitar o problema mencionado é por meio do uso de uma *m-estimativa* para estimar a probabilidade dos termos condicionais da equação 3.4 (MITCHIE, 1997). Essa *m-estimativa* é definida da seguinte forma:

$$\mathbf{m-estimativa:} \quad \frac{n_e + mp}{n + m} \quad (3.5)$$

Na expressão 3.5, n_e e n são definidas da mesma forma como foram apresentadas no começo desta subseção, p é a estimativa de probabilidade que se deseja determinar e m é uma constante denominada *tamanho da amostra equivalente*, que determina o quanto de peso se quer que p possua em relação aos dados observados.

Um método típico para a escolha de p na ausência de outra informação é assumir uma distribuição uniforme para p , isto é, se um atributo pode assumir k valores distintos, então define-se $p=1/k$. Por exemplo, ao estimar $P(\text{Vento} = \text{Forte} | \text{JogarTênis} = \text{Não})$, é possível notar que o atributo *Vento* tem dois possíveis valores, o que faz com que $p = 0.5$ – assumindo uma distribuição uniforme para p . Além disso, também é interessante observar que se m for zero, então a *m-estimativa* é equivalente à fração original n_e/n e p será combinada de acordo com o peso m . O motivo de m ser denominada de *tamanho da amostra equivalente* é porque a equação 3.5 pode ser interpretada como *um aumento no número atual, n , de observações, por meio da adição de m exemplos virtuais distribuídos de acordo com p .*

Na próxima seção é apresentado o algoritmo geral de classificação de documentos textuais proposto por Mitchell (1997) com base no método de classificação Bayesiano descrito nesta seção.

3.4 Algoritmo de aprendizagem

Nesta seção é apresentado um algoritmo geral de aprendizagem para a classificação de textos, baseado no método de classificação Bayesiano simples apresentado na seção anterior.

Primeiramente, vamos delimitar e definir mais precisamente o problema a ser tratado. No problema de classificação de textos, que é o foco do algoritmo desta seção, é considerado um espaço de instâncias X , consistindo de todos os possíveis documentos textuais (todas as possíveis *strings* de palavras e pontuações de todos os comprimentos possíveis), e um conjunto de exemplos de treinamentos de alguma função objetivo desconhecida $f(x)$, sendo que esta pode assumir qualquer valor de um conjunto finito V . A tarefa é aprender, a partir desse conjunto inicial de exemplos de treinamento, a prever o valor-objetivo – estimar uma probabilidade que indique o grau de relevância – para os documentos textuais subsequentes.

Na Tabela 3.1 é apresentado um algoritmo Bayesiano capaz de classificar instâncias de documentos com as características mencionadas no parágrafo anterior. Esse algoritmo utiliza dois passos para calcular uma estimativa da relevância para um documento textual qualquer com base em um conjunto inicial de exemplos de treinamento

O primeiro passo é aquele no qual o algoritmo analisa os exemplos de treinamento armazenados na sua base de conhecimentos e estima probabilidades para cada uma das palavras presentes neles (função *Aprendizagem_Classificador_Bayesiano*). Nesta etapa são examinados todos os exemplos de treinamento e é extraído todo o vocabulário (palavras) que aparecem nesses exemplos. Uma vez feito isso, o algoritmo conta as frequências de cada uma dessas palavras (termo n_i) presentes nas diferentes classes de exemplos – uma classe é um conjunto de documentos textuais que estão rotulados da mesma forma, como por exemplo: documentos que “interessam”, “não interessam”; documentos de “biologia”,

“matemática”; etc. Por fim, a partir dessa frequência, é calculada uma probabilidade para cada uma dessas palavras (termo $P(w_k|v_j)$).

Aprendizagem_Classificador_Bayesiano (*Exemplos*, V)

Exemplos é um conjunto de documentos textuais. V é o conjunto de todos os possíveis valores-objetivos.

1. Coletar todas as palavras que ocorrem em *Exemplos*:
 - *Vocabulário* \leftarrow o conjunto de todas as palavras distintas que ocorrem em qualquer documento textual do conjunto *Exemplo*
2. Calcular a probabilidade dos termos $P(w_k|v_j)$ e $P(v_j)$:
 - Para cada valor-objetivo v_j em V faça:
 - $docs_j \leftarrow$ subconjunto dos documentos de *Exemplos* para o qual o valor-objetivo é v_j
 - $P(v_j) \leftarrow \frac{|docs_j|}{|Exemplos|}$
 - $Texto_j \leftarrow$ um único documento criado como resultado da concatenação de todos os membros de $docs_j$
 - $n \leftarrow$ número total de posições de palavras distintas em $Texto_j$
 - Para cada palavra w_k do *vocabulário*
 - $n_k \leftarrow$ número de vezes que a palavra w_k ocorre em $Texto_j$
 - $P(w_k|v_j) \leftarrow \frac{n_k + 1}{n + |vocabulary|}$

Classificar_textos_pelo_classificador_Bayesiano (*Doc*)

Retorna o valor-objetivo estimado para o documento *Doc*. a_i representa a palavra encontrada na i -ésima posição de *Doc*.

- *Posições* \leftarrow todas as posições de palavras em *Doc* que contêm palavras encontradas no *vocabulário*
- Retorna v_{nb} , onde: $v_{nb} = \operatorname{argmax} P(v_j) \prod_i P(a_i|v_j)$ onde, $v_j \in V$; $i \in \text{Posições}$

Tabela 3.1 - Algoritmo Bayesiano simples para a aprendizagem e classificação de documentos textuais.

O segundo passo é aquele no qual o algoritmo necessita estimar um certo grau de relevância para um novo documento textual (função *Classificar_textos_pelo_classificador_Bayesiano*) com o qual se depara. Essa estimativa é feita por meio do cálculo do produto

das probabilidades de palavras que estão presentes tanto nos exemplos de treinamento já analisados quanto no documento para o qual será feita uma estimativa de relevância do mesmo. Esse produto (termo v_{nb}) será a relevância calculada para o documento pelo algoritmo.

Na próxima seção são apresentadas as considerações finais deste capítulo.

3.5 Considerações finais

Como foi apresentado neste capítulo, as diversas técnicas de aprendizagem de máquina existentes podem ser classificadas, segundo Mitchell (1997), de acordo com o paradigma de aprendizagem que utilizam: *paradigma indutivo* ou *paradigma analítico*.

A técnica de aprendizagem de máquina implementada no agente Chapa segue o paradigma de aprendizagem indutiva; mais especificamente, foi escolhida a técnica indutiva denominada aprendizagem Bayesiana. Isso porque essa técnica, dentre todas as técnicas pertencentes ao paradigma de aprendizagem indutiva, é uma das mais efetivas citadas na literatura (LEWIS, 1991; LANG, 1995; JOACHIMS, 1996) para a classificação de documentos textuais.

Como o objetivo básico do agente construído nesta dissertação é a análise de mensagens de sessões de bate-papo, que contêm fundamentalmente texto escrito – cada mensagem de uma sessão de bate-papo pode ser considerada um documento textual –, optou-se por essa técnica de aprendizagem visto que oferece uma abordagem comprovadamente efetiva para tratar esse tipo de problema de acordo com a literatura.

Além disso, é importante ressaltar duas outras características inerentes à técnica de aprendizagem Bayesiana descrita neste capítulo.

A primeira delas é o fato de que, na técnica de aprendizagem Bayesiana, a cada novo dado de treinamento observado, é possível *diminuir* ou *aumentar* a probabilidade estimada da validade de uma determinada hipótese. Essa característica permite aos algoritmos que utilizam esta técnica serem mais flexíveis que aqueles que utilizam técnicas onde uma hipótese é descartada sumariamente se é considerada inconsistente com algum exemplo individual previamente observado.

A outra característica inerente a esta técnica é a possibilidade de serem acomodadas hipóteses que fazem previsões probabilísticas, isto é, hipóteses tais como “*este paciente de pneumonia tem 93% de chance de se recuperar completamente*” ou “*esta mensagem tem 85% de chance de interessar ao usuário*”.

Finalizando, é apresentado no próximo capítulo o agente de interface Chapa que auxilia um professor a selecionar as mensagens que considerar interessantes de uma sessão de bate-papo ocorrida no ambiente TelEduc.

Capítulo 4

Agente de Interface Chapa

4.1 Introdução

Neste capítulo é apresentado o agente de interface, denominado Chapa. Todas as suas funcionalidades, sua interface e as formas pelas quais um professor de um curso à distância pode selecionar determinadas mensagens de uma sessão de bate-papo que lhe interessam são apresentadas.

A ferramenta de Bate-papo, como concebida originalmente, é uma ferramenta de comunicação síncrona informal, isto é, não tem grande estruturação⁷ nem restrições nas mensagens enviadas por meio dela. Essa característica faz com que a interação se dê “*de forma natural e informal, tendo em vista que é relativamente não-planejada, ou seja, a construção da interação vai sendo planejada e replanejada a cada novo lance do jogo da linguagem*” (DIONÍSIO, 2001), diferentemente do que, provavelmente, aconteceria se a

⁷ A única estruturação presente nas mensagens dos bate-papos é com relação à existência do nome da pessoa que enviou a mensagem (remetente), da pessoa que recebeu a mensagem (destinatário) e à da entonação da fala (“pergunta para”, “fala para”, etc), além do próprio texto da mensagem. Ex.: *José pergunta para Maria: Que dia é hoje ?*

opinião do participante fosse postada via uma ferramenta de comunicação como o Correio ou Fórum de Discussões.

Essa naturalidade e informalidade na conversação permitem aos participantes de uma sessão de bate-papo dar uma vazão maior às suas idéias o que, conseqüentemente, se reproduz em um número enorme de mensagens irrelevantes e descartáveis.

Quando usada no contexto da EaD, a ferramenta de bate-papo acaba originando uma grande sobrecarga para o professor, uma vez que ele tem que analisar todas as mensagens enviadas em uma sessão de bate-papo para poder avaliar adequadamente as contribuições dos alunos.

A fim de diminuir a sobrecarga do professor na análise de sessões de bate-papo realizadas no ambiente TelEduc, foi desenvolvido o Chapa, uma ferramenta baseada no conceito de agentes de interface, que possibilita a seleção automática de mensagens de acordo com os interesses do professor.

Nas subseções seguintes são apresentadas as principais características do agente de interface construído. A seção 4.2 apresenta o (re)*design* do registro das sessões de bate-papo do ambiente TelEduc feito para dar suporte ao processo de seleção de mensagens pelo usuário. A seção 4.3 apresenta os modos pelos quais o usuário pode selecionar as mensagens que deseja. A seção 4.4 apresenta as formas de aprendizagem do agente de interface. A seção 4.5 apresenta a análise efetuada pelo Chapa sobre as mensagens de uma sessão de bate-papo. A seção 4.6 apresenta a possibilidade de múltiplas visões permitida pelo Chapa. A seção 4.7 apresenta os testes efetuados com usuários. Finalizando, na seção 4.8 são feitas as considerações finais deste capítulo.

4.2 Interface para seleção de mensagens

A ferramenta de bate-papo do TelEduc gera um registro com todas as mensagens de uma sessão de bate-papo realizada no ambiente. A interface original desse registro (Figura 4.1) possibilita apenas a visualização do conjunto de mensagens de uma sessão, não sendo possível a seleção de mensagens pelo usuário.

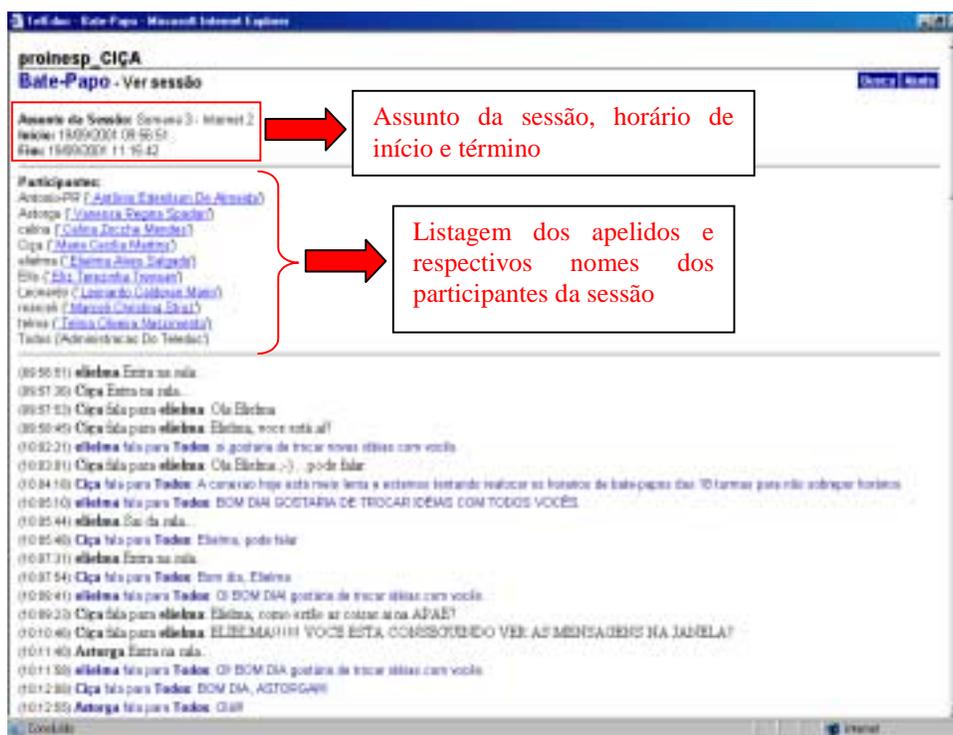


Figura 4.1 - Visualização do registro de uma sessão de bate-papo. Além das mensagens também são apresentadas informações sobre o assunto agendado para a sessão e a listagem dos nomes dos participantes e respectivos apelidos

Dessa forma, foi realizado um (re)design do registro das sessões (Figura 4.2), a fim de possibilitar a seleção de mensagens pelo usuário e indicação dos critérios da seleção. Na nova interface foi mantida a apresentação de todas as informações que já existiam na interface original e acrescentadas algumas novas:

- As informações sobre o assunto da sessão, horários de início e término continuam sendo apresentadas da mesma forma que originalmente (Figura 4.2-a);
- A listagem com os nomes e apelidos dos participantes deixou de aparecer diretamente na interface e passou a estar acessível por meio de um link (Figura 4.2-b);
- Foram acrescentados dois links para acesso as funcionalidades do agente (Figura 4.2-c). O conteúdo e as funções contidas em cada um deles serão explicados nas sessões 4.3.2 e 4.6;

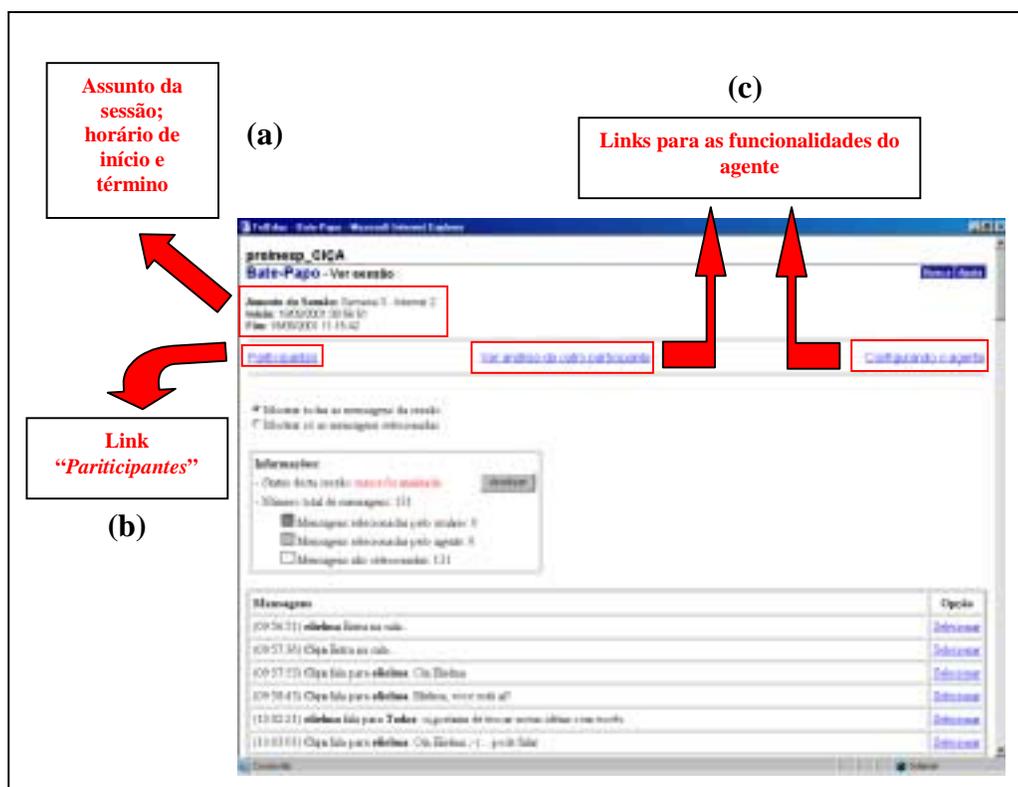


Figura 4.2 – Localização das informações disponíveis na nova interface de visualização do registro do bate-papo. (a) Informações sobre o assunto da sessão, horário de início e término da sessão. (b) Link para a listagem dos nomes e respectivos apelidos dos participantes da sessão. (c) Links para as funcionalidades do agente.

- Foi acrescentada uma legenda com informações do agente relativas ao registro da sessão que está sendo visualizada (Figura 4.3-a). Nela estão presentes informações a respeito do número total de mensagens da sessão, número de mensagens selecionadas pelo agente, número de mensagens selecionadas pelo usuário e um “*status*” – estado – da sessão atual. Uma sessão pode estar em qualquer um dos três estados: “*nunca foi analisada*”, “*necessário analisar*”, “*não é necessário analisar*”. O primeiro é o estado em que uma sessão se encontra quando esta nunca foi analisada pelo agente – na primeira visualização do registro de uma sessão ela se encontra nesse estado. O segundo estado ocorre quando há alguma modificação (acréscimo ou decréscimo de informação) na base de conhecimentos do agente – o agente procura, por meio desse estado, indicar o fato de que possui novas informações que pode aplicar na análise da sessão. O terceiro estado ocorre quando o agente já analisou a sessão e, desde a última análise, não houve nenhuma modificação na sua base de conhecimentos; conseqüentemente, o agente por meio da indicação desse estado verifica que não é necessário analisar a sessão porque não possui nenhum conhecimento novo para aplicar em uma análise. Finalizando, há um botão ‘*Analisar*’ cujo intuito é justamente pedir para que o agente analise a sessão utilizando-se da sua base atual de conhecimentos. Esse botão está presente em dois lugares: ao lado da informação com o estado da sessão e no final da lista de mensagens.
- As mensagens da sessão foram colocadas em uma tabela com duas colunas onde, na primeira coluna aparecem as mensagens enviadas pelos participantes e, na segunda, as duas opções de ação que são possíveis de serem executadas sobre essas mensagens: “*Desmarcar seleção*” e “*Selecionar*” (Figura 4.3-b). A primeira opção aparece somente para as mensagens selecionadas e a segunda somente para as mensagens não selecionadas. Além disso, foi colocado um cabeçalho na primeira linha da tabela indicando o conteúdo das duas colunas evitando que o usuário tenha que memorizar os seus significados.

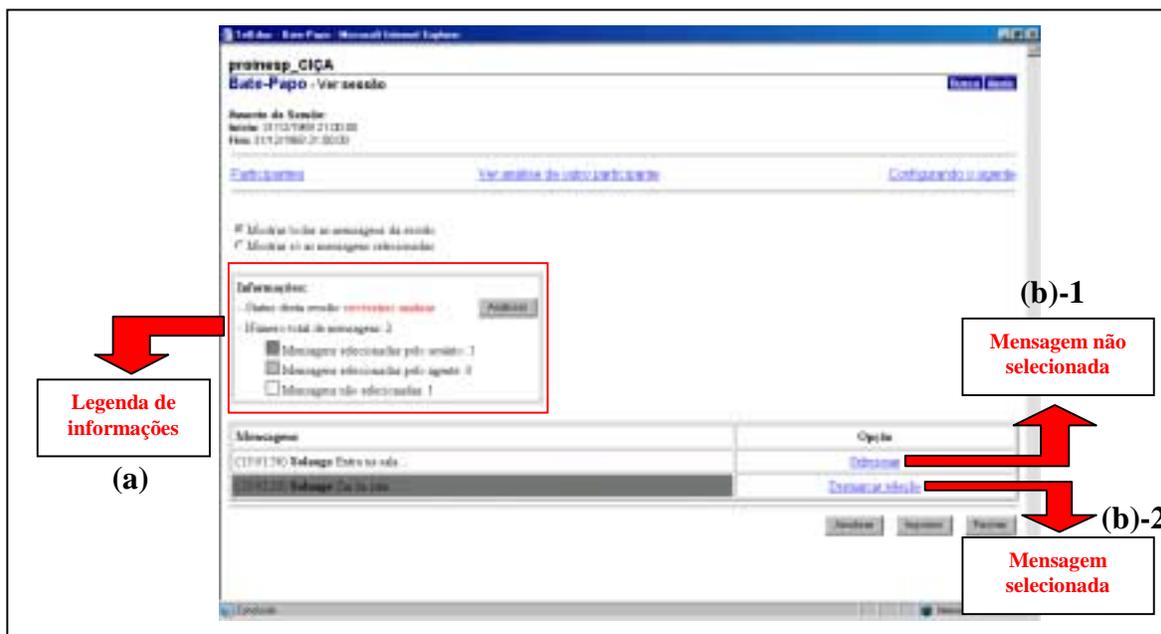


Figura 4.3 – Interface de visualização das informações apresentadas pelo agente. (a) Informações do agente: “status” da sessão, número total de mensagens da sessão, número de mensagens selecionadas pelo usuário e número de mensagens selecionadas pelo agente. (b)-1 Visualização de uma mensagem não selecionada: fundo branco e a única opção disponível é a opção “Selecionar”. (b)-2 Visualização de uma mensagem selecionada pelo usuário: fundo cinza escuro e a única opção disponível é a opção “Desmarcar seleção”.

Na próxima seção são apresentadas os modos pelos quais o usuário pode selecionar as mensagens que lhe interessam.

4.3 Formas de seleção de mensagens

Há, fundamentalmente, duas formas pelas quais um usuário pode selecionar ou desmarcar a seleção de mensagens. Uma delas é por meio da opção de ação “Selecionar”/“Desmarcar seleção” disponível para todas as mensagens não selecionadas de uma sessão. A outra é por meio do link “Configurando o agente”. Nas próximas subseções são apresentadas em detalhes essas duas opções de seleção.

4.3.1 Opção de ação “Selecionar”/ “Desmarcar seleção”

Ao se visualizar o registro de uma sessão de bate-papo por meio do Chapa, está sempre disponível para cada mensagem da sessão um link para a ação que é possível ser executada sobre a mensagem – link “Selecionar” ou link “Desmarcar seleção”. O link para a ação “Selecionar” aparece para cada uma das mensagens que não está selecionada e permite selecionar a respectiva mensagem, já o link “Desmarcar seleção” permite a ação contrária, desmarcando a seleção de uma mensagem que está selecionada.

Por exemplo, na primeira vez em que se visualiza o registro de uma sessão de bate-papo, todas as mensagens aparecem com a opção “Selecionar” disponível (Figura 4.4). O usuário, para selecionar uma mensagem que considerar importante, deve clicar sobre essa opção.

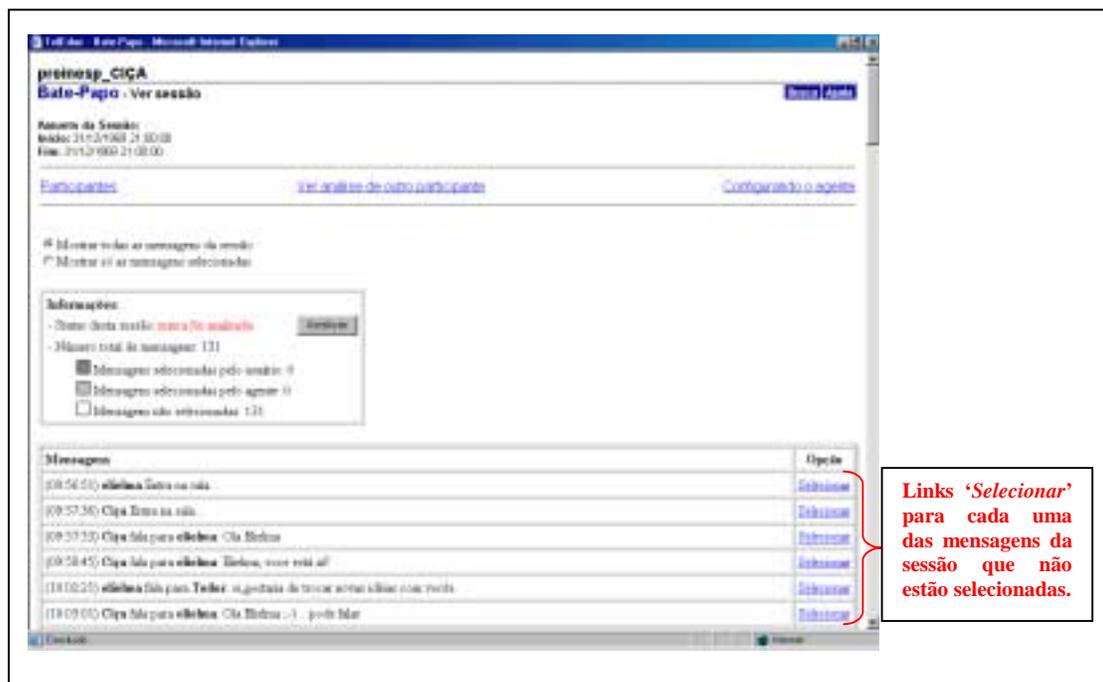


Figura 4.4 – Primeira vez que se visualiza o registro de uma determinada sessão de bate-papo. A opção “Selecionar” aparece disponível para todas as mensagens da sessão, uma vez que nenhuma delas foi selecionada ainda.

Neste momento, é aberta uma janela de seleção da mensagem, contendo a mensagem que está sendo selecionada e os critérios (seleção pelo nome do remetente, pela entonação da fala, pelo nome do destinatário e/ou pelas palavras-chave presentes na mensagem) passíveis de serem indicados como causa da seleção (Figura 4.5). Esses critérios são armazenados na base de conhecimentos do agente e, quando for pedido para o agente analisar uma sessão, ele vai selecionar qualquer mensagem que possua esses mesmos motivos.

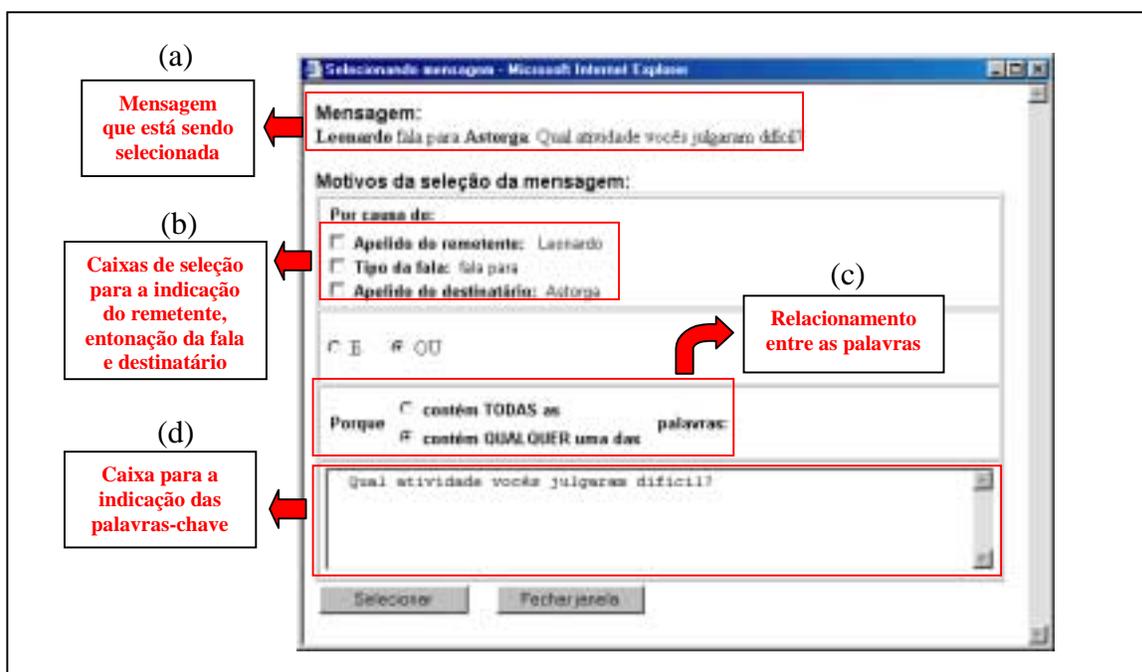


Figura 4.5 - Janela de indicação dos critérios de seleção de uma mensagem. (a) A mensagem que o usuário está selecionando. (b) Caixas de seleção para poder indicar o remetente, entonação da fala e destinatário como motivos da seleção da mensagem. (c) Relacionamento entre as palavras-chave. (d) Caixa para indicar as palavras-chave da mensagem como motivo da seleção da mensagem. Essa caixa já aparece preenchida com o texto da mensagem no intuito de diminuir o trabalho do usuário.

O conjunto de critérios foi definido como podendo ser uma combinação de qualquer um dos elementos componentes de uma mensagem. Basicamente, podem ser identificados em uma mensagem quatro elementos distintos:

1. **Remetente:** o apelido⁸ do participante da sessão de bate-papo que enviou a mensagem;
2. **Entonação da fala:** a forma da fala da pessoa que enviou a mensagem. Há diversas entonações de fala possíveis, tais como, “fala para”, “pergunta para”, “responde para”, etc.;
3. **Destinatário:** o apelido da pessoa a quem se destina a mensagem;
4. **Texto da mensagem:** é o conteúdo propriamente dito da mensagem.

No exemplo seguinte é apresentada uma mensagem com identificação dos quatro elementos descritos anteriormente:

José fala para **Maria**: Estou com problemas na realização da tarefa. Poderia me ajudar?

Remetente: “*José*”;

Entonação da fala: “*fala para*”.

Destinatário: “*Maria*”;

Texto da mensagem: “*Estou com problemas na realização da tarefa. Poderia me ajudar?*”.

Além de poder indicar qualquer um dos quatro elementos mencionados, também é possível indicar como o agente deve entender o relacionamento entre as palavras. Pode-se indicar ou que a mensagem foi selecionada porque contém *todas* as palavras indicadas marcando a opção “contém TODAS as”, ou que foi selecionada porque possui *qualquer* uma das palavras indicadas marcando a opção “contém QUALQUER uma das”. Essa característica é análoga à existente nas ferramentas tradicionais de busca na rede.

⁸ Normalmente, as pessoas que participam de sessões de bate-papo não se identificam, necessariamente, utilizando o seu nome próprio; ao invés disso, se identificam por meio de *apelidos*.

Uma vez indicados todos os motivos que levaram a seleção da mensagem o usuário clica no botão ‘Selecionar’ e termina o processo de seleção da mensagem.

O *feedback* da seleção de uma mensagem é feito por meio da mudança da cor de fundo da mensagem – passa a ser cinza escuro ao invés de branco – e também, ocorre uma mudança na opção disponível da mensagem: passa de “Selecionar” para “Desmarcar seleção” (ver Figura 4.6).

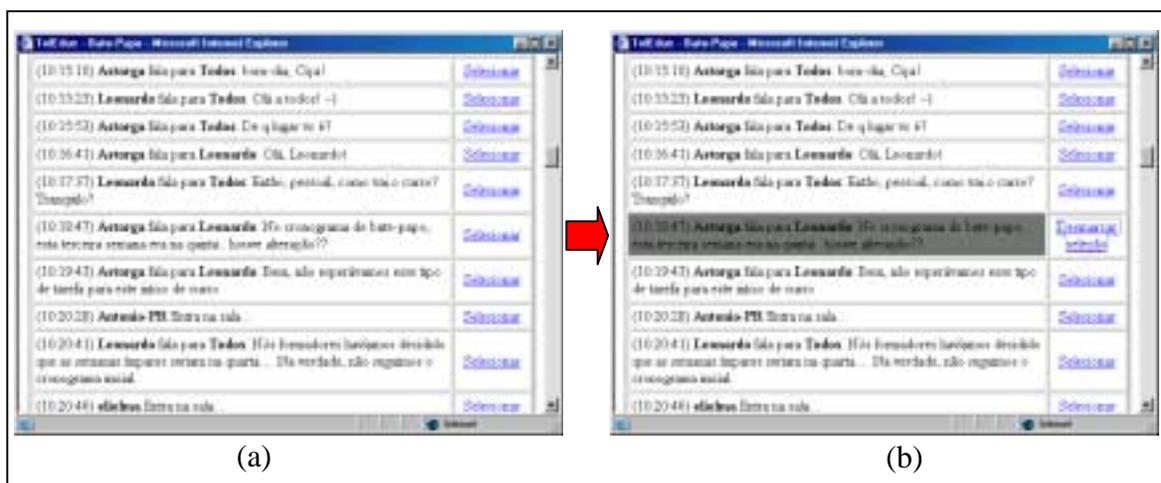


Figura 4.6 – Feedback apresentado pelo agente após a seleção de uma mensagem pelo usuário. (a) Mensagem antes de ser selecionada. (b) Mensagem após ter sido selecionada.

O processo de desmarcar a seleção de uma mensagem é análogo ao processo de seleção. Para desmarcar uma mensagem o usuário deve clicar sobre o respectivo link “Desmarcar seleção” de uma mensagem selecionada. Uma vez feito isso, é aberta uma janela, contendo informações sobre o autor da seleção e os motivos da sua seleção (Figura 4.7).

Além dessas informações, também está presente uma opção (Figura 4.7–d) que permite ao usuário dizer o que fazer com os motivos indicados para justificar a seleção da mensagem uma vez que ela seja desmarcada – como já foi mencionado anteriormente, os motivos indicados para justificar a seleção de uma mensagem são armazenados na base de

conhecimentos do agente e ele os utiliza para selecionar outras mensagens. Há duas opções disponíveis: “*Continuar considerando esses motivos*” e “*Não considerar mais esses motivos*”. A primeira opção serve para fazer com que a mensagem seja desmarcada mas que o agente continue usando esses motivos para selecionar outras mensagens. A segunda opção, além de também desmarcar a mensagem, exclui definitivamente esses motivos da base de conhecimentos do agente; conseqüentemente, o agente não mais selecionará nenhuma outra mensagem com base neles.

A definição dessas opções foi para permitir a um usuário indicar que uma mensagem não é relevante apesar de conter algum dos elementos que ele considera importantes. Por exemplo, uma palavra-chave que em uma mensagem em particular foi usada em um contexto completamente diferente daquele no qual o usuário está interessado.

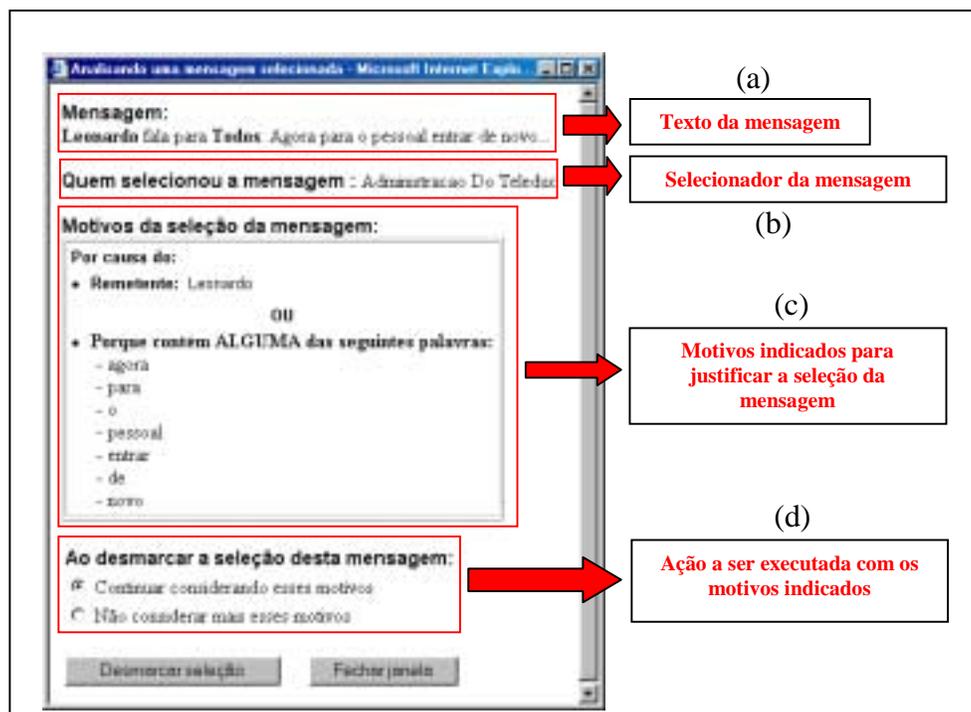


Figura 4.7 - Janela com os critérios pelos quais uma mensagem foi selecionada. (a) O texto da mensagem. (b) O usuário que selecionou a mensagem. (c) Os motivos indicados pelo usuário para ter selecionado a mensagem. (d) Ao desmarcar a seleção desta mensagem qual a ação a ser tomada com os motivos indicados

Concluindo, as opções de “*Selecionar*” e “*Desmarcar seleção*” permitem ao usuário indicar nas próprias mensagens de uma sessão de bate-papo o que considera importante nelas e quais mensagens lhe interessam ou não.

Na subseção seguinte é apresentada a outra opção que o usuário tem à sua disposição para a seleção de mensagens de uma sessão de bate-papo.

4.3.2 Link “*Configurando o agente*”

Essa opção também permite ao usuário indicar um conjunto de elementos que considera importantes e selecionar as mensagens de uma sessão que contenha qualquer um deles. A diferença em relação à opção anterior é a forma como o usuário indica esses elementos. Ao invés de indicá-los nas próprias mensagens de uma sessão de bate-papo, ele os indica separadamente e independentemente das mensagens por meio do link “*Configurando o agente*” (ver Figura 4.2-c).

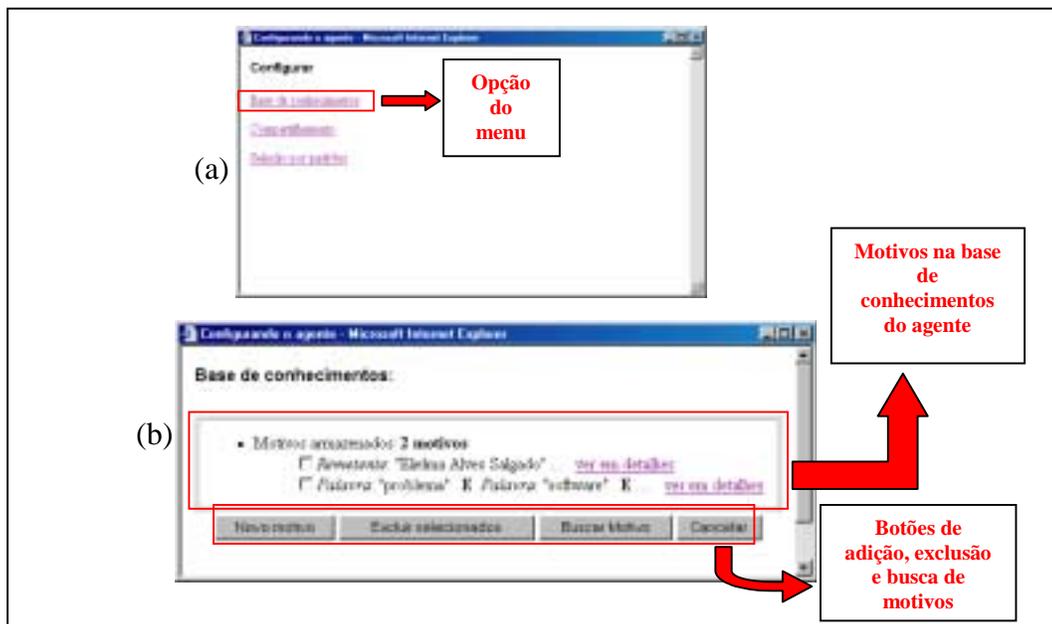


Figura 4.8 – (a) Janela onde aparece um menu com todas as opções de configuração do agente. (b) Janela onde aparece a base de conhecimentos do agente.

Ao clicar sobre o link “*Configurando o agente*” é aberta a janela da Figura 4.8-a. Nela aparece um menu com todas as opções de configuração do agente e deve ser clicado, em um primeiro momento, o link “*Base de conhecimentos*” (cada uma das outras opções do link será explicada oportunamente) abrindo uma nova janela. Nessa nova janela que é aberta (Figura 4.8-b), podem ser vistos todos os motivos armazenados na base de conhecimentos do agente. Esses motivos são aqueles que o agente utiliza e aparecem em uma forma simplificada, de modo a não ocuparem muito espaço na tela. No entanto, existe ao lado de cada motivo um link ‘*ver em detalhes*’ que permite a visualização detalhada do respectivo motivo (Figura 4.9). Além disso, também existem botões que permitem adicionar, excluir e buscar motivos nessa base de conhecimentos do agente. É, justamente, por meio do botão “*Novo motivo*” que é possível indicar um novo motivo de seleção de mensagens além do meio explicado na subseção anterior.

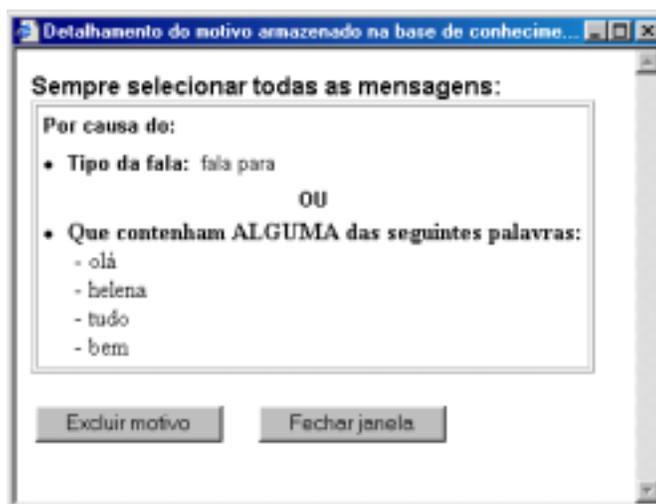


Figura 4.9 – Visão em detalhes de um motivo armazenado na base de conhecimentos do agente

Ao clicar no botão “*Novo motivo*” (Figura 4.8-b), a janela da Figura 4.10 é aberta. Essa janela é semelhante aquela da Figura 4.5 e permite a indicação dos mesmos critérios: seleção pelo nome do remetente, pela entonação da fala, pelo nome do destinatário e/ou pelas palavras-chave presentes na mensagem. Ao finalizar a indicação de um novo motivo

(clitando no botão “*Salvar*”), ele é armazenado na base de conhecimentos do agente e passa a ser utilizado por ele na seleção de mensagens. Um detalhe importante a ser notado é a possibilidade de se excluir qualquer um dos motivos armazenados na base de conhecimentos do agente por meio do botão “*Excluir selecionados*” (Figura 4.8-b).



Figura 4.10 – Janela que permite a definição de um novo motivo de seleção de mensagens pelo usuário. O novo motivo que pode ser definido pode conter uma combinação entre quaisquer dos quatro elementos presentes nas mensagens de uma sessão de bate-papo: remetente, entonação da fala, destinatário e texto da mensagem.

Finalizando, uma outra funcionalidade existente é a capacidade de busca e exclusão de algum motivo que esteja armazenado na base de conhecimentos do agente. Essa funcionalidade foi implementada para se evitar a necessidade de idas e vindas à opção “*ver detalhes*” de cada um dos motivos apresentados (Figura 4.8-b) – o número de motivos armazenados pode ser muito grande e, conseqüentemente, essas idas e vindas demandariam muito tempo do usuário – com o intuito de se encontrar aquele que se quer excluir.

Isso pode ser feito por meio do botão “*Buscar motivo*” da Figura 4.8-b. Ao clicar nesse botão é aberta uma janela (Figura 4.11) onde se pode indicar quais as características dos motivos armazenados que devem ser procurados – a busca pode ser feita por quaisquer motivos que selecionam mensagens com alguma combinação dos elementos: remetente, entonação de fala, destinatário e/ou palavras-chave. É interessante ressaltar, que a interface da janela de busca (Figura 4.11) disponibiliza ao usuário todos os elementos presentes na janela que permite a definição de um novo motivo (Figura 4.10), dando ao usuário a possibilidade de especificar uma busca capaz de localizar qualquer motivo armazenado anteriormente na base de conhecimentos do Chapa.



Figura 4.11 – Janela de busca de motivos armazenados na base de conhecimentos do agente.

Na Figura 4.12 é apresentada a janela com o resultado de uma busca onde foi encontrado um motivo com as características procuradas – remetente: “Elielma Alves Salgado”. Completando, nessa janela existe um botão “*Excluir selecionados*” que permite

excluir da base de conhecimentos qualquer um dos motivos retornados como resultado da busca.



Figura 4.12 – Resultado da busca pelos motivos cujo elemento “Remetente” tem o valor “Elielma Alves Salgado”.

Concluindo, nesta seção foram apresentadas as duas formas que o usuário tem à disposição para selecionar as mensagens de uma sessão de bate-papo e a interface disponibilizada pelo Chapa para esse fim. Na próxima seção, são apresentadas as formas de aprendizagem implementadas pelo Chapa com base nas diferentes formas de aprendizagem vistas no Capítulo 2 a respeito de agentes de interface (subseção 2.3.3).

4.4 Aprendizagem do Chapa

O Chapa possui três formas de aprender os interesses do usuário na seleção das mensagens de uma sessão de bate-papo: por observação, pelo *feedback* do usuário e a partir de exemplos dados explicitamente pelo usuário. Nas próximas subseções são apresentadas essas três formas de aprendizagem implementadas pelo Chapa.

4.4.1 Aprendizagem por observação

Esta forma de aprendizagem ocorre antes do agente efetuar qualquer análise das mensagens de uma sessão. Nessa fase, o usuário seleciona um certo conjunto de mensagens e indica, para cada mensagem, os motivos que o levaram a fazer a seleção.

Durante o processo de seleção do usuário, o agente vai “observando” os motivos indicados para cada mensagem selecionada.

A cada seleção do usuário, o agente armazena todos os motivos indicados na sua base de conhecimentos com o intuito de construir um dicionário dinâmico de dados com informações a respeito dos interesses específicos do usuário – esse dicionário é dinâmico porque, como foi visto na seção anterior, o usuário pode adicionar e excluir motivos continuamente da base de conhecimentos do agente.

O dicionário de dados construído para um usuário é usado pelo agente para analisar as mensagens de uma sessão e selecionar as relevantes. A análise propriamente dita, efetuada pelo agente, será descrita em detalhes na próxima seção; no entanto, aqui deve ser mencionado que é possível pedir para o agente analisar uma sessão a qualquer momento, mesmo antes do usuário selecionar alguma mensagem da sessão para o agente observar – basta clicar sobre o botão “*Analisar*” (ver Figura 4.3-a). No entanto, logo de início, o agente não terá muitos subsídios para poder decidir sobre a seleção ou não de alguma mensagem da sessão, uma vez que não terá a sua disposição muitas informações a respeito dos interesses do usuário armazenadas no seu dicionário de dados.

Na subseção seguinte é apresentada a outra forma de aprendizagem implementada pelo agente: *a aprendizagem via feedback do usuário*.

4.4.2 Aprendizagem via feedback do usuário

Esta forma de aprendizagem ocorre após a análise de uma sessão de bate-papo pelo agente. Ao finalizar a seleção das mensagens de uma sessão de bate-papo o agente apresenta os resultados para o usuário (Figura 4.13). Nesse momento, o usuário pode começar a *revisar* a seleção feita pelo agente, ou seja, o usuário pode marcar mensagens não selecionadas pelo agente e/ou cancelar seleções realizadas pelo agente. Por meio desse

processo, o usuário vai ajustando a base de conhecimentos do agente de forma a que represente o máximo possível os seus interesses.

Dessa forma, o resultado final da análise de uma sessão de bate-papo é um conjunto de mensagens selecionadas tanto pelo usuário quanto pelo agente (Figura 4.13). A fim de facilitar a identificação da procedência de cada seleção, as mensagens selecionadas foram destacados com dois tons de cinza⁹ (um cinza claro identificando mensagens do agente e um cinza escuro identificando mensagens do usuário). Conforme pôde ser visto na Figura 4.3-a, existe uma legenda onde está indicada essa associação de cores com a respectiva procedência da seleção para evitar a necessidade de memorização dessa associação.



Figura 4.13 – Apresentação visual do resultado de uma análise efetuada pelo Chapa.

⁹ Essa foi a coloração definida inicialmente, mas será estudada a utilização de outros tipos de coloração ou então a possibilidade de se deixar o usuário indicar a cor com a qual quer que apareçam destacadas as mensagens selecionadas.

Esse é fundamentalmente o processo de aprendizagem via *feedback* do usuário. Na subseção seguinte é apresentada a terceira forma de aprendizagem implementada pelo agente: *a aprendizagem a partir de exemplos*.

4.4.3 Aprendizagem a partir de exemplos

Esta forma de aprendizagem ocorre quando o usuário indica os motivos de seleção por meio do link “*Configurando o agente*” – ver segunda opção de seleção na subseção 4.3.2. Nela o usuário indica explicitamente quais motivos o agente deve procurar em uma mensagem durante a análise para selecioná-la.

Por exemplo, suponha que um aluno queira selecionar todas as mensagens do professor. Para isso, esse aluno clica no link “*Configurando o agente*”, no botão “*Novo motivo*”, seleciona o nome do professor na lista de mensagens “*enviadas por*” e clica no botão “*Salvar*” para que o motivo seja armazenado na base de conhecimentos do agente. Com isso, a partir desse momento, o agente sempre selecionará todas as mensagens enviadas pelo professor ao analisar uma sessão.

Na próxima seção, é descrita em detalhes a análise de uma sessão de bate-papo realizada pelo Chapa.

4.5 Análise do Chapa

Conforme já foi dito anteriormente, o objetivo do Chapa é selecionar as mensagens de uma sessão de bate-papo que interessam ou não a um usuário. Para atingir esse objetivo o Chapa se utiliza do dicionário de dados que constrói de um usuário a partir das formas de aprendizagem descritos na seção anterior. Nesta seção são apresentados os passos que o Chapa executa durante o seu processo de análise das mensagens de uma sessão de bate-papo.

Primeiramente, antes de analisar qualquer mensagem, o Chapa aplica a função *Aprendizagem_classificador_bayesiano* (Tabela 3.1) sobre a sua base de conhecimentos para calcular os termos $P(v_j)$ e $P(w_k|v_j)$, isto é, para calcular quantas mensagens o usuário normalmente costuma selecionar ou não (termo $P(v_j)$) e para estimar uma probabilidade para cada uma das palavras selecionadas que possui armazenadas na sua base de conhecimentos – entenda-se como sendo somente as palavras pertencentes ao quarto elemento de uma mensagem (texto da mensagem).

Em seguida, faz a seguinte verificação: *se a mensagem que está sendo analisada se encaixa em algum dos motivos armazenados na base de conhecimentos*. Em caso afirmativo seleciona a mensagem, caso contrário, não. Por exemplo, veja na Tabela 4.1 um exemplo da seleção de uma mensagem desse modo.

| |
|--|
| <p>Suponha que o Chapa está analisando a seguinte mensagem:</p> <p>José pergunta para Maria: Olá, como vai ?</p> <p>E suponha que na base de conhecimentos esteja armazenado o motivo:</p> <p>“Sempre selecionar as mensagens com a entonação de fala: <i>pergunta para</i>”.</p> <p>Conseqüentemente, o Chapa ao analisar a mensagem verificará que possui uma entonação de fala que deve ser selecionada e seleciona a mensagem.</p> |
|--|

Tabela 4.1 – Exemplo da análise e seleção de uma mensagem feita pelo Chapa

No exemplo, é apresentada uma seleção simples, mas o Chapa é capaz de selecionar uma mensagem por meio do casamento entre os elementos da mensagem e qualquer dos motivos que tenha armazenado na sua base de conhecimentos. Além disso, se for considerado que o motivo que o Chapa consegue armazenar pode ser composto de qualquer combinação entre os elementos de uma mensagem então, pode-se dizer que esse modo de selecionar uma mensagem é análogo a uma consulta SQL (*Script Query Language*).

Por fim, caso a mensagem ainda não tenha sido selecionada, o Chapa aplica a função *Classificar_textos_pelo_classificador_Bayesiano* (Tabela 3.1) sobre a mensagem e estima uma relevância para ela (valor v_{nb}). Se a relevância que estimar para a mensagem for maior que 50% então a mensagem é selecionada. Fundamentalmente, essa é a análise que o Chapa executa sobre as mensagens de uma sessão para selecionar aquelas que interessam a um usuário.

Finalizando, é possível desativar a seleção de mensagens feita pelo Chapa a partir da aplicação da função *Classificar_textos_pelo_classificador_Bayesiano* e selecionar somente as mensagens que casam com algum dos motivos adicionados pelo usuário diretamente à base de conhecimentos por meio das opções de seleção descritas nas seções 4.3.1 e 4.3.2. Para isso é necessário configurar o agente por meio dos seguintes passos: clicar no link “*Configurando o agente*” (Figura 4.2-c), opção “*Seleção por padrões*” (Figura 4.8-a), marcar a opção “*Não*” e clicar sobre o botão “*Salvar*” (Figura 4.14).



Figura 4.14 – Janela que ativa ou não a aplicação do algoritmo de classificação Bayesiano.

Na próxima seção é apresentada uma outra característica existente no agente que permite múltiplas visões, isto é, ver a análise feita pelos agentes de outros usuários.

4.6 Múltiplas visões

Uma outra característica implementada no agente é a possibilidade de compartilhá-lo com outros participantes, isto é, permitir que um outro participante do curso possa ver um conjunto de mensagens selecionadas por um agente que não seja o seu (“visão”).

Na Figura 4.15, é apresentada uma representação esquemática dessa possibilidade de configuração do agente. Na figura, o agente do participante A tem visão do conjunto de mensagens selecionadas de uma sessão de bate-papo cuja seleção foi efetuada pelo agente do participante B.

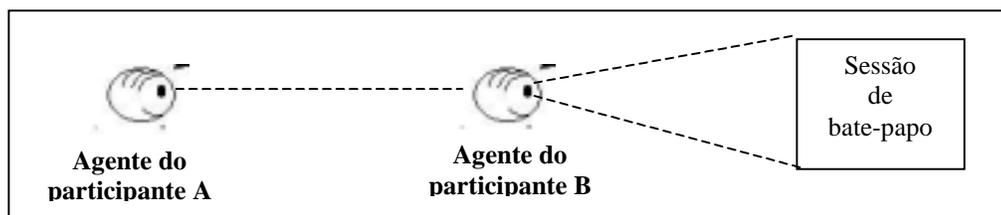


Figura 4.15 – Esquema de visões. O agente do participante A pode visualizar a análise feita pelo agente do participante B.

Um fato que é importante ressaltar é que o agente do participante A não importa/copia para a sua base de conhecimentos os motivos armazenados na base de conhecimentos do participante B ao visualizar a análise do agente de B. Em um primeiro momento, poderia se pensar que essa importação seria interessante – como por exemplo, um usuário, que nunca tenha efetuado uma análise, poderia preencher a base de conhecimentos de seu agente com os motivos de um outro usuário e utilizá-los na análise de alguma sessão. No entanto, como depois, inevitavelmente, o usuário acabaria tendo que gastar um tempo ajustando a base de conhecimentos do seu agente para selecionar somente as mensagens que lhe interessam essa vantagem inicial já não seria mais tão significativa, se é que existiria. Por isso, foi definido que é possível, simplesmente, a visualização do conjunto de mensagens selecionadas pelo agente de um outro usuário e não a importação/cópia dos motivos.

Para poder visualizar a visão de um outro usuário é necessário que ele compartilhe o seu agente. Esse compartilhamento é feito por meio do link “*Configurando o agente*” (Figura 4.2-c), opção “*Compartilhamento*” (Figura 4.16-a). Essa opção abre uma janela (Figura 4.16-b) com a listagem de todos os outros usuários de um curso à distância. Pode-se selecionar um ou mais deles com que se queira compartilhar o agente e clicar no botão “*Aplicar*”. Uma vez feito isso, todos os usuários indicados têm a possibilidade de visualizar as mensagens selecionadas por seu agente sobre o registro de qualquer sessão de bate-papo existente no ambiente.

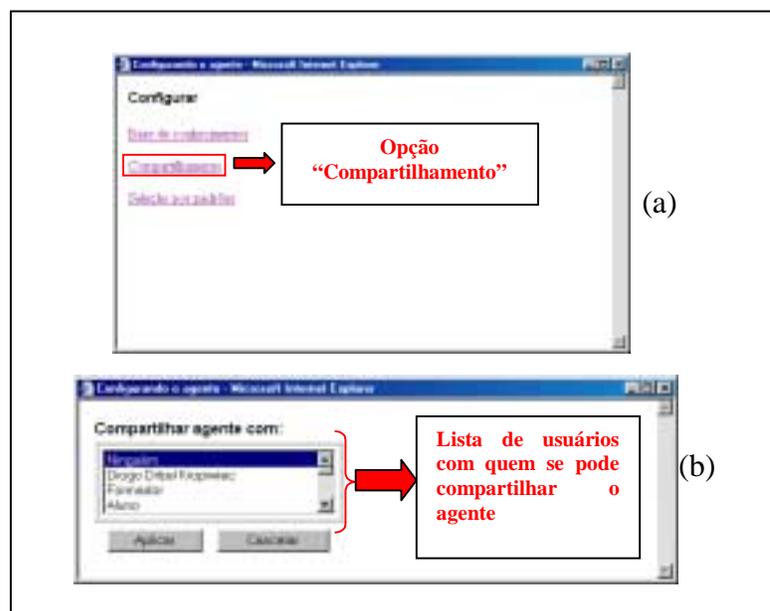


Figura 4.16 – Compartilhamento do agente. (a) Opção “*Compartilhamento*” do menu de configurações. (b) Lista de usuários com os quais se pode compartilhar o agente.

A visualização do conjunto de mensagens selecionadas por um agente compartilhado de um outro usuário pode ser feita por meio do link “*Ver análise de outro participante*” (Figura 4.2-c). Ao clicar sobre esse link é aberta uma janela com a listagem de todos os usuários que compartilharam seus agentes e, portanto, é possível visualizar o conjunto de mensagens selecionadas pelo agente deles (Figura 4.17).



Figura 4.17 – Usuários que compartilharam os agentes. No caso da figura está destacado que o participante ‘Aluno’ compartilhou seu agente.

Ao clicar sobre o botão “Visualizar” é aberta uma janela com a visão do participante selecionado. Nela o nome do participante de quem se está tendo a visão é apresentado na parte superior da janela, a fim de facilitar a sua identificação (Figura 4.18).

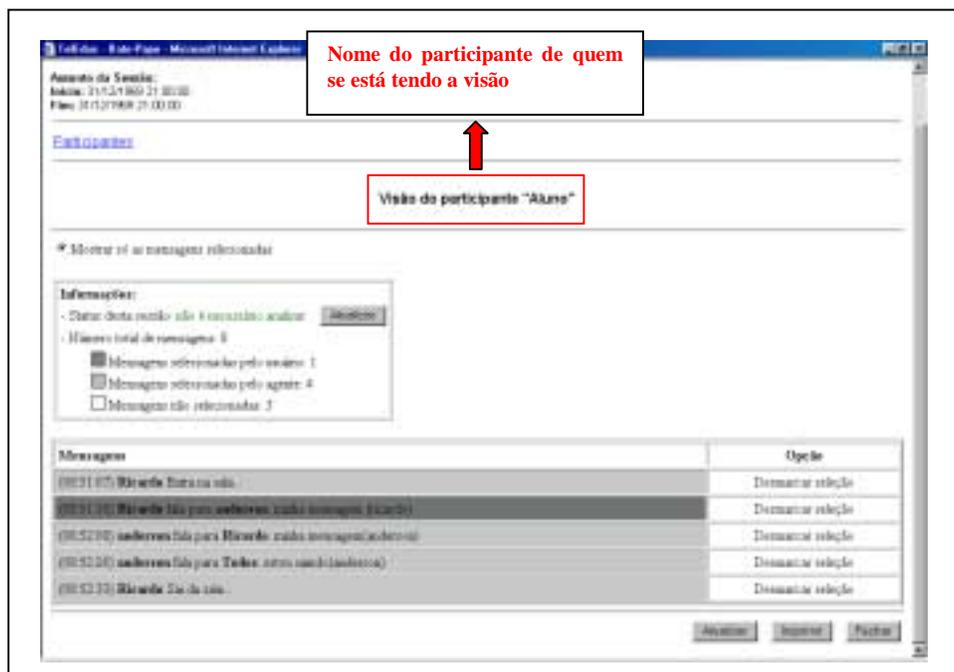


Figura 4.18 – Visão do participante “Usuário”.

Finalizando, como é aberta uma nova janela para cada visão, é possível abrir mais de uma visão ao mesmo tempo, permitindo a visualização simultânea da análise do agente de outros participantes (múltiplas visões simultâneas).

Na próxima seção, são apresentados os testes preliminares, efetuados com alguns usuários, utilizando o Chapa para a análise de sessões de bate-papo.

4.7 Testes preliminares

Os testes realizados com o Chapa consistiram na análise de três sessões de bate-papo por um grupo de quatro usuários – todos alunos de doutorado ou mestrado em Ciência da Computação e Educação da UNICAMP que já haviam participado, ou como alunos, ou como professores de cursos à distância – onde cada usuário do grupo analisou individualmente todas as três sessões.

O objetivo da primeira análise foi deixar o usuário ter um primeiro contato com o Chapa, conhecer as suas funcionalidades e interface. Foi indicado que o usuário deveria tentar analisar a sessão procurando explorar as funcionalidades do agente, apontando sugestões de melhorias na interface e quaisquer dificuldades encontradas.

Na segunda análise, foi procurado observar se o agente efetivamente foi capaz de auxiliar aos usuários na tarefa de análise de uma sessão de bate-papo. Basicamente, foi apresentada uma outra sessão, completamente diferente da primeira, sobre a qual cada usuário deveria tentar selecionar as mensagens relevantes para ele por meio do agente. Ao final, foi perguntado o grau de satisfação do usuário com o auxílio do agente.

Já o objetivo da terceira análise foi aplicar o agente para analisar por conta própria uma sessão de bate-papo, tendo por base somente as informações armazenadas na sua base de conhecimentos a partir das duas análises anteriores efetuadas pelo usuário. Uma vez efetuada essa primeira análise por parte do agente, o usuário deveria observar o resultado apresentado pelo agente, revisá-lo e emitir uma opinião a respeito, tanto do resultado inicial apresentado quanto do trabalho de revisão realizado.

Nas próximas subseções são apresentados os resultados obtidos nas três análises realizadas.

4.7.1 Análise 1

Nesta análise os usuários apresentaram diversas propostas de alterações na interface decorrentes de problemas que enfrentaram ao realizar a análise da sessão e, em média, cada usuário gastou 1 hora e meia nessa primeira análise. A seguir, são apresentadas as principais sugestões propostas:

1. Inicialmente, a terminologia utilizada para se referir ao texto enviado por um usuário em uma sessão de bate-papo foi a palavra ‘*comentário*’. No entanto, esse termo causou grande confusão e foi trocado pelo termo ‘*mensagem*’ (ver Figura 4.19).
2. Foi acrescentado um botão para poder ser impresso o registro da sessão, contendo todas as mensagens selecionadas e não selecionadas (ver Figura 4.19).
3. Foi acrescentada uma legenda para indicar qual a cor de uma mensagem quando a mesma não está selecionada, equivalentemente, às que já existiam para as mensagens selecionadas pelo usuário e pelo agente (ver Figura 4.19).
4. Foi acrescentado um indicativo do *status* da sessão para permitir ao usuário ter um *feedback* do agente sobre a necessidade ou não de analisar a sessão. Isso foi consequência do fato de que os usuários pediam para o agente analisar a sessão e ficavam em dúvida se o agente realmente havia analisado a sessão causando um repetido número de re-análises desnecessárias (ver Figura 4.19).
5. Foi acrescentado um botão ‘*Analisar*’ com a mesma função do que já existia na parte inferior da tela ao lado do link de *status*, pois o usuário ficava sem saber como fazer para o agente analisar a sessão quando o número de mensagens da

sessão era muito grande. Nessa situação o botão ‘Analisar’ presente na parte inferior ficava “escondido” do usuário (ver Figura 4.19).

6. Foi feita a substituição do título da segunda coluna da tabela contendo as mensagens, de ‘Selecionado’ para ‘Opção’. Isso foi feito como decorrência de dois pontos que foram apontados: eliminar a redundância de informação e facilitar o entendimento da sua função. A redundância existia porque a informação passada pela coluna ‘Selecionado’ – cujas opções possíveis eram ‘Sim’ ou ‘Não’ – já existia de forma mais clara e explícita, por meio da cor de fundo da mensagem, por isso não acrescentava nenhuma informação nova ao usuário. O outro ponto, relativo ao entendimento da sua função, era mais séria, isso porque os links ‘Sim’ e ‘Não’ não indicavam para o usuário qual a sua função – permitir selecionar ou não a mensagem correspondente – que é justamente um dos dois meios pelos quais o usuário podia selecionar as mensagens da sessão (ver Figura 4.19).

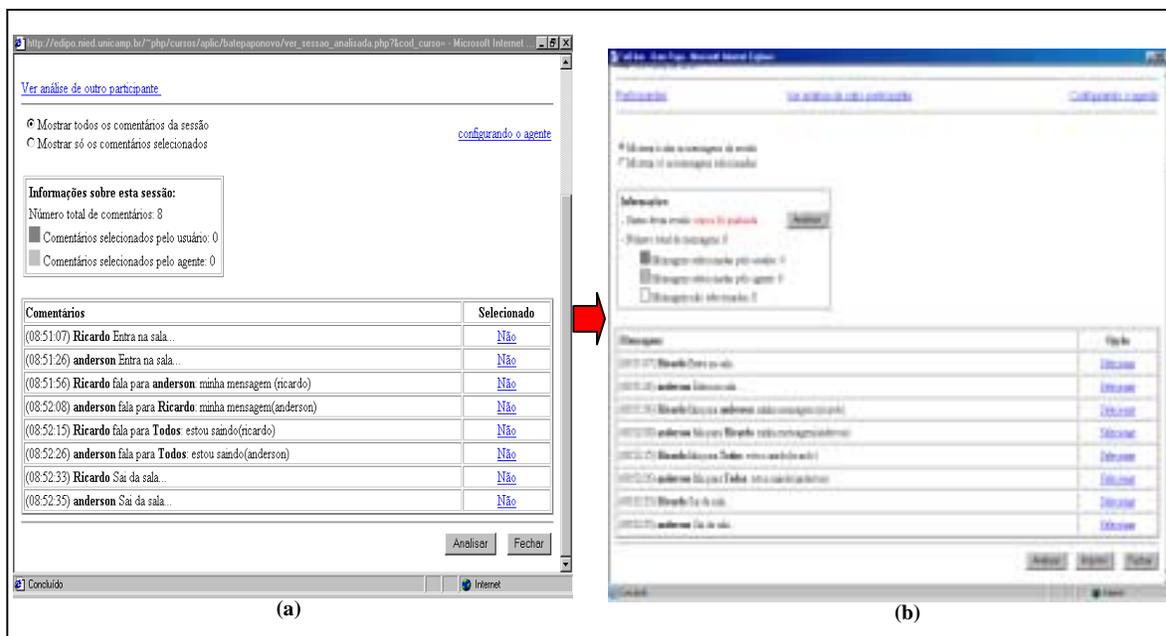


Figura 4.19 – As modificações efetuadas na interface do Chapa com base nas sugestões feitas pelos usuários durante a análise. (a) Interface antes das modificações. (b) Interface após as modificações propostas nos itens 1-6.

7. Na janela onde são indicados os motivos da seleção de uma mensagem – é aberta após se ter clicado sobre a opção “*Selecionar*” de uma mensagem não selecionada – foram acrescentados os conectores lógicos ‘E’ e ‘OU’ ligando os elementos 1-3 e o elemento 4 da mensagem. Isso foi feito para permitir ao usuário expressar de forma mais flexível e visível esse relacionamento (antes era considerado sempre um ‘E’ lógico ligando esses elementos). Veja o resultado desta modificação na Figura 4.20.
8. Na mesma janela do item anterior, foram colocados os elementos 1-4 da mensagem dentro de uma tabela de forma a indicar mais claramente que todos eles podem ser indicados como motivos da seleção da mensagem – na interface anterior, foi apontado por alguns usuários que não era fácil visualizar que o elemento 4 também podia ser indicado como um dos motivos da seleção da mensagem, devido a existência de um ‘espaço em branco’ entre esse elemento e o três primeiros (ver Figura 4.20).

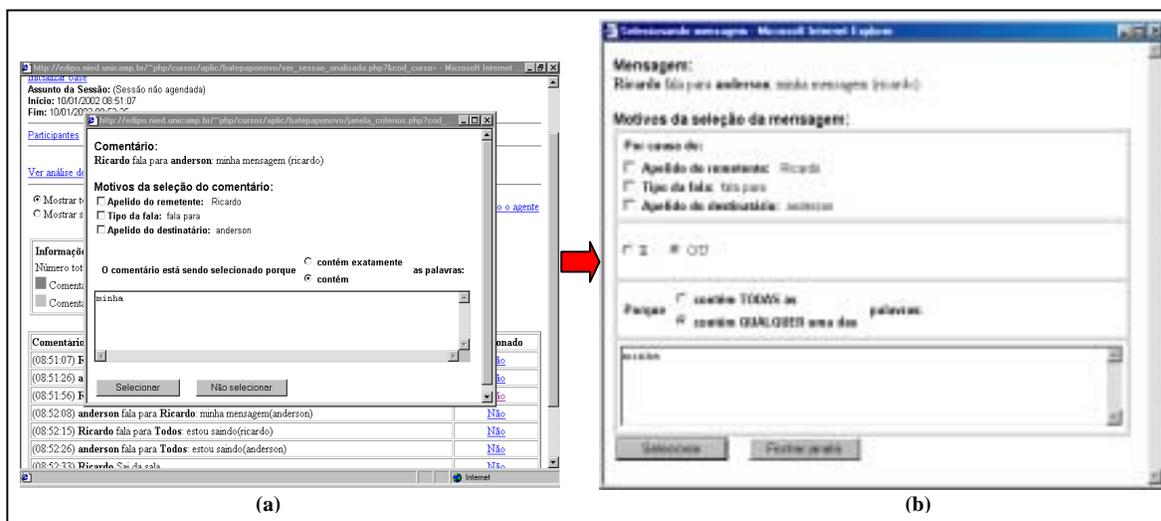


Figura 4.20 – As modificações na interface de seleção de uma mensagem por meio de um clique sobre o link de opção “*Selecionar*” – primeira opção de seleção vista na subseção 4.3.1. (a) Interface antes das modificações. (b) Interface após das modificações propostas nos itens 7-8.

9. Analogamente, a janela dos itens 7 e 8, também a janela onde aparecem os motivos indicados para justificar a seleção de uma mensagem – aberta após se ter clicado sobre a opção “*Desmarcar seleção*” de uma mensagem selecionada – foi reestruturada. As seguintes mudanças foram feitas: foi reescrito o texto que aparece mostrando qual o relacionamento entre as palavras selecionadas (passou de “*Palavras que motivaram a seleção do comentário*” para “*Porque contém ALGUMA das seguintes palavras*”); foram acrescentadas duas opções de ação ao desmarcar a seleção da mensagem (“*Continuar considerando esses motivos*” e “*Não considerar mais esses motivos*”). A primeira mudança foi necessária para corrigir uma falha relevante de interface que foi apontada pelos usuários durante a análise, uma vez que não era possível saber qual havia sido o relacionamento indicado para as palavras de uma mensagem selecionada. Já a segunda mudança, foi apontada pelos usuários como uma forma de evitar ter que desmarcar a seleção de uma mensagem e, nos casos em que também não se quer mais selecionar nenhuma outra mensagem por causa dos motivos indicados, não precisar ir na base de conhecimentos do agente e apagá-los diretamente de lá (ver Figura 4.21).

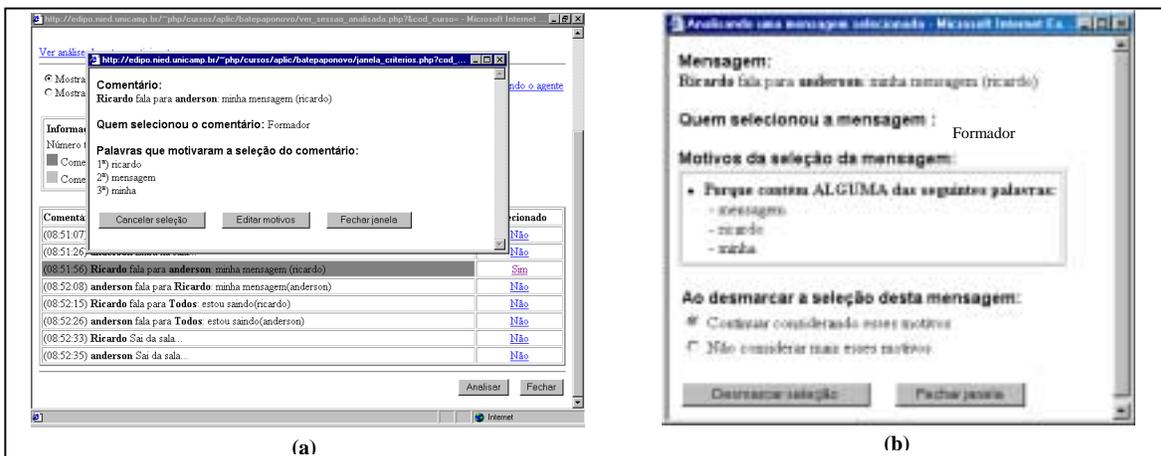


Figura 4.21 – As modificações na interface que permite desmarcar a seleção de uma mensagem (a) Interface antes das modificações. (b) Interface após as modificações propostas no item 9.

10. Além disso, foram apontados diversos problemas de consistência em vários outros lugares do sistema, tais como, falta de títulos em algumas janelas, escrita de um mesmo termo de formas diferentes, etc. Todos esses problemas apontados foram corrigidos.

Na próxima subseção é apresentado o resultado da segunda análise de uma sessão de bate-papo por parte de um grupo de quatro usuários.

4.7.2 Análise 2

Nesta análise foram observados dois comportamentos principais apresentados pelos usuários: um no qual o usuário utilizava predominantemente a forma de seleção por meio das opções – “*Selecionar*” e “*Desmarcar seleção*” – disponíveis para cada mensagem (ver subseção 4.3.1); outro, no qual o usuário incluía diretamente uma lista de motivos na base de conhecimentos do agente por meio do link “*Configurando o agente*” (ver subseção 4.3.2).

Os usuários do primeiro grupo, no geral, analisaram detalhadamente as mensagens que selecionaram e tomaram cuidado ao indicar os motivos para justificar cada seleção. Depois de terem analisado um certo número de mensagens – na média umas 70 mensagens de um número total de 350 – pediram para o agente analisar a sessão. Após essa análise do agente, observaram o resultado e verificaram todas as mensagens selecionadas pelo agente e também as não selecionadas, procurando por mensagens que julgavam interessantes mas que não foram selecionadas. Com relação a outra forma de seleção existente, eles a usaram de forma bem tímida e o tempo gasto em todo o processo de análise – seleção e revisão – foi por volta de 1 hora. Consideraram o agente interessante mas como disseram que gostam de ler e selecionar por si próprios todas as mensagens ficaram *razoavelmente* satisfeitos com o auxílio oferecido pelo agente.

O segundo grupo de usuários utilizou predominantemente a opção de adicionar novos motivos diretamente na base de conhecimentos do Chapa por meio do link “*Configurando o agente*”. Inicialmente, fizeram uma leitura rápida do registro da sessão (~7 minutos); uma vez feita essa leitura inicial, gastaram uma média de 20 minutos no processo de preenchimento da base de conhecimentos, após os quais, pediram para o agente analisar a sessão. Depois revisaram o resultado apresentado¹⁰, etapa na qual gastaram mais uns 15 minutos. Por fim, ficaram satisfeitos com o auxílio disponibilizado pelo Chapa.

Na próxima subseção é apresentado o resultado da análise da terceira sessão de bate-papo pelos usuários.

4.7.3 Análise 3

Nesta etapa o tempo gasto foi bem menor, em média, cada usuário gastou 10 minutos (a sessão utilizada para esta análise continha aproximadamente 500 mensagens e o conteúdo dela não tinha *a priori* nada particularmente semelhante a qualquer uma das outras duas sessões analisadas).

Dois pontos interessantes foram notados durante esta análise. O primeiro ponto foi com relação aos usuários do primeiro grupo mencionados na análise 2. Os usuários deste grupo ficaram bem mais impressionados e satisfeitos com a utilização do agente. Isso provavelmente ocorreu pela possibilidade de utilização de uma ferramenta capaz de selecionar um conjunto das mensagens da sessão – que possuía 500 mensagens – do que ter que ler todas as mensagens da sessão após já terem efetuado uma média de 2 horas e meia de análises anteriores.

¹⁰ Neste momento foi sugerida pelos usuários, a colocação de uma opção para desmarcar de forma mais rápida as mensagens que foram selecionadas incorretamente – algo do tipo ‘*Desmarcar rapidamente*’, além da opção “*Desmarcar seleção*” – sem que houvesse a necessidade da janela com a apresentação dos motivos da seleção da mensagem ser aberta.

O outro ponto interessante foi com relação ao segundo grupo de usuários. Este grupo utilizou de uma espécie de *tática* para a seleção das mensagens relevantes. Aplicaram o agente, deram uma olhada rápida no resultado e depois executaram o seguinte processo: primeiro, procuraram saber quem era(m) o(s) formador(es) da sessão. Depois indicaram para o agente selecionar todas as mensagens enviadas por ele(s). Observaram essas mensagens, selecionaram algumas das palavras-chaves contidas nelas e as adicionaram à base de conhecimentos do agente. Pediram para o agente analisar novamente a sessão e, por fim, visualizaram o resultado fazendo algumas poucas alterações. Nesse momento pararam e disseram que estavam satisfeitos com o resultado.

Na próxima seção são apresentadas as considerações finais relativas a este capítulo.

4.8 Considerações Finais

Neste capítulo foi descrita, em detalhes, uma ferramenta baseada no conceito de agentes de interface capaz de selecionar quais as mensagens, dentre todas as mensagens de uma sessão de bate-papo já realizada e registrada no ambiente TelEduc, são as mais interessantes para um determinado usuário de um curso à distância.

Um ponto a ser ressaltado nessa ferramenta é a construção dos dicionários de dados *personalizados* para cada usuário pois são eles, justamente, que permitem a seleção de um conjunto de mensagens específicas para um usuário. Essa personalização é importante porque permite que a ferramenta escolha as mensagens mais adequadas aos interesses de um professor e/ou aluno de um curso à distância.

A aplicação do conceito de agentes de interface foi, justamente, para se conseguir essa *personalização*: a idéia foi utilizar um agente, cuja metáfora na qual se baseia é a de *um assistente pessoal da vida real*, para preencher um dicionário de dados com

informações personalizadas de um usuário, dando um caráter personalizado a esse dicionário.

Concluindo, os testes preliminares apresentaram resultados promissores do desempenho do agente na tarefa de auxiliar na seleção das mensagens de uma sessão de bate-papo. No entanto ainda é necessário testá-lo em situações reais de curso para poder validar efetivamente a sua flexibilidade, capacidade e eficácia.

No próximo capítulo são apresentadas as conclusões finais deste trabalho de mestrado.

Capítulo 5

Conclusões

Em geral, é descrito na literatura que um curso ministrado à distância acaba ocasionando uma carga maior de trabalho ao professor do que ocasionaria se o mesmo curso fosse dado presencialmente. Os professores queixam-se do excesso de mensagens, da sobrecarga de informação, da falta de ferramentas para um melhor acompanhamento do desempenho dos alunos e do aumento do tempo despendido com os cursos. Tomando por base essas queixas relatadas, o objetivo central desta pesquisa foi: *como ajudar o professor a lidar com o enorme volume de informações geradas em um curso à distância.*

Na literatura são apresentados inúmeros trabalhos que se propõem a auxiliar o professor a lidar com esse enorme volume de informações. Particularmente, nesta dissertação, foi tomada como referência principal o trabalho de Jaques e Oliveira (2000) que aplica agentes de interface e dicionários de dados para a coleta de informações relevantes em ambientes de EaD. Foi com base nesse trabalho que surgiu a idéia utilizada nesta dissertação: *aplicar agentes de interface na construção de dicionários de dados personalizados para a seleção de mensagens de sessões de bate-papo visando auxiliar na tarefa de análise das sessões de bate-papo em ambientes de EaD.*

Enquanto no trabalho de Jaques e Oliveira (2000), são pré-embutidos dicionários de dados sintáticos e léxico-morfológicos para determinar a relevância das palavras, nesta

dissertação procurou-se aplicar agentes de interface para construir esses dicionários de dados. Só que, ao invés deles conterem a sintaxe e/ou semântica de palavras em geral, conterão um conjunto de palavras relevantes para um professor.

O resultado foi a concepção do agente de interface *Chapa*, um agente de interface capaz de selecionar quais as mensagens, dentre todas as mensagens de uma sessão de bate-papo já realizada e registrada no ambiente TelEduc, são as mais interessantes para um determinado professor de um curso à distância.

Ele foi construído baseado no conceito de agentes de interface para poder aproveitar a possibilidade de personalização inerente a essa classe de agentes e aplicá-la na construção de dicionários de dados contendo os interesses de um usuário relativos às mensagens de sessões de bate-papo. Especificamente, esforços foram centrados na ferramenta de bate-papo porque, dada sua natureza síncrona, é a ferramenta de comunicação que gera o maior número de informações irrelevantes e descartáveis.

Como a construção de um dicionário de dados específico para cada usuário é o ponto fundamental sobre o qual se baseia a utilidade do Chapa, disponibilizou-se um conjunto flexível e variado de possibilidades de preenchimento desse dicionário procurando sempre deixar “na mão do usuário” o controle sobre o que está contido no dicionário e, conseqüentemente, as mensagens que são selecionadas com base nele.

Os testes preliminares efetuados com o Chapa, foram promissores e indicaram que o Chapa obteve uma boa aceitação e foi considerado uma ferramenta útil no auxílio a seleção das mensagens interessantes de uma sessão de bate-papo.

Com essa primeira avaliação do Chapa pôde-se perceber que é necessária a realização de novas experiências no sentido de identificar novas funcionalidades e

melhorias que podem ser incorporadas ao Chapa, além de permitir uma validação dele em situações reais de um curso à distância.

Nas novas versões do agente é preciso considerar a inclusão de algumas funcionalidades sugeridas durante os testes como: caminhos mais rápidos para a seleção de mensagens e a disponibilização de mais informações para o usuário, como por exemplo, quais participantes de uma sessão de bate-papo fizeram contribuições relevantes, isto é, tiveram mensagens selecionadas, dentre outras. Além disso, como uma extensão deste trabalho vislumbra-se uma série de possibilidades.

Uma primeira possibilidade poderia ser o desenvolvimento de trabalhos correlatos visando estender o uso do Chapa a outras ferramentas de comunicação do ambiente TelEduc, tais como, fóruns de discussão, correio eletrônico entre outras. Ainda dentro dessa possibilidade, seria importante analisar e incorporar as funcionalidades sugeridas pelos professores durante a realização dos testes do Chapa na ferramenta de bate-papo.

Outra possibilidade seria a incorporação da capacidade de se agrupar um conjunto de critérios sob um mesmo “*nome*” e poder aplicá-lo individualmente na seleção das mensagens. Por exemplo, suponha que existam dois critérios já definidos na base de conhecimentos, onde um deles seleciona todas as mensagens **enviadas** pelo professor e outro que seleciona todas as mensagens **recebidas** pelo professor. De acordo com a possibilidade proposta poderia ser possível agrupar esses dois critérios sob um novo nome, tipo, “*Conjunto de critérios que selecionam mensagens recebidas e enviadas pelo professor*” e aplicá-lo independentemente dos demais na seleção das mensagens relevantes de uma sessão. Generalizando, o usuário poderia agrupar e nomear novos critérios que podem ser compostos por uma combinação de critérios individuais e/ou por um conjunto de critérios já agrupados e nomeados anteriormente, em um processo teoricamente infinito.

Um outro caminho bem mais geral e abrangente poderia ser o estudo e a análise da possibilidade de incorporação ao Chapa da possibilidade de "*map matching*" estendendo o atual "*pattern matching*", isto é, ao invés de buscas por padrões poderiam ser feitas buscas por casamento de *Mapas Conceituais*.

De acordo com seu criador, Novak (1977), Mapas Conceituais são representações gráficas semelhantes a diagramas, que indicam relações entre conceitos ligados por palavras e representam uma estrutura que vai desde os conceitos mais abrangentes até os menos inclusivos.

Na literatura já há trabalhos que aplicam agentes e mapas conceituais para a representação de conhecimento (BAX e SOUZA, 2001) sendo que, estes últimos, já vêm sendo usados há algum tempo como sistemas de representação formal de conhecimentos em redes semânticas na área de inteligência artificial (GAINES e SHAW, 1995).

Além dessas possibilidades, trabalhos também podem ser realizados no sentido de generalizar a ferramenta para outros ambientes de educação à distância na Web além do TelEduc. Isso implicaria na busca de soluções para a leitura e tratamento dos dados a partir de diferentes bases de dados utilizadas pelos ambientes.

Referências bibliográficas

- AULANET (2000). AulaNet 2.0 beta 3 – 1997-2000. Fundação Padre Leonel Franca – PUC-Rio. Visitado em 25 ago. 2000. Online. Disponível na Internet: <http://anuel.cead.puc-rio.br/aulanet/index.html>.
- BAX, M. P.; SOUZA, R. R. Uma Proposta de Uso de Agentes e Mapas Conceituais para Representação de Conhecimentos Altamente Contextualizados In: SIMPÓSIO INTERNACIONAL DE GESTÃO DO CONHECIMENTO/GESTÃO DE DOCUMENTOS - ISKM/DM, 4, Curitiba, 2001. **Anais**. Capturado em 01 de fevereiro 2003. Online. Disponível na Internet: <http://cuba.eci.ufmg.br/Bax/Publis/Publis/agentes.pdf>.
- BISHOP, C. M. **Neural networks for pattern recognition**. Oxford, England: Oxford University Press. 1995, 504 p.
- BOOKER, L. B.; GOLDBERG, D. E.; HOLLAND, J. H. Classifier systems and genetic algorithms. **Journal of Artificial Intelligence**. n.40, p.235-282, 1989.
- BRADSHAW, J. M. An Introduction to Software Agents. In: BRADSHAW, J.M. (Ed.) **Software Agents**. CA:AAAI/MIT Press, 1997. Cap.1, p.3-46.

- CASELLA, G.; BERGER, R. L. **Statistical inference**. Pacific Grove, CA: Wadsworth & Brooks/Cole. 1990. cap.1, p.1-44: Probability Theory.
- CHIN, D.N. Intelligent Interfaces as Agents. In: SULLIVAN, J.; TYLER, S. (Eds.) **Intelligent User Interfaces**. Addison-Wesley, Reading, MA, 1991. cap.9, p.127-206.
- DIONÍSIO, A. P. Análise da Conversação. In: MUSSALIM, F.; BENTES, A. C. (Org.) **Introdução à Lingüística: Domínios e Fronteiras**. São Paulo, SP: Cortez, 2001. v.2, cap.3, p.69-100.
- ETZIONI, O.; WELD, D.S. Intelligent agents on the Internet: Fact, fiction, and forecast. **IEEE Expert**, v.10, n.4, p.44-49, Ago. 1995.
- FERBER, J. Simulating with Reactive Agents, In HILLEBRAND, E.; STENDER, J. (Eds.) **Many Agent Simulation and Artificial Life**, Amsterdam: IOS Press, 1994. p.8-28.
- FONER, L. **What's an Agent, Anyway? A Sociological Case Study**. 1993. Capturado em 14 set. 2001. Online. Disponível na Internet em: http://kuba.korea.ac.kr/~ixix/Article/agent/what_agent.pdf.
- FRANKLIN, S.; GRAESSER, A. Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents. In: PROCEEDINGS OF THE INTERNATIONAL WORKSHOP ON AGENT THEORIES, ARCHITECTURES AND LANGUAGES, 3, Springer-Verlag, 1996. **Proceedings**. Capturado em 05 set. 2001. Online. Disponível na Internet em: <http://www.msci.memphis.edu/~franklin/AgentProg.html>.
- FREEMAN, J. A.; SKAPURA, D. M. **Neural Networks: Algorithms, Applications, and Programming Techniques**. Reading, MA: Addison Wesley, 1991. 401p.

FUCKS, H. Aprendizagem e trabalho cooperativo no ambiente AulaNet. Rio de Janeiro, 2000. 23p. Monografia - Faculdade de Ciência da Computação, Pontifícia Universidade Católica do Rio de Janeiro. Capturado em 12 out. 2002. Online. Disponível na Internet em: <http://piano.dsi.uminho.pt/grupok3/bibdig/aprendizagem.pdf>.

GAINES, B.R.; SHAW, M.L.G. Collaboration through Concept Maps. In: INTERNATIONAL CONFERENCE ON COMPUTER SUPPORT FOR COLLABORATIVE LEARNING, Bloomington, 1995. **Proceedings**. Capturado em 2 fev. 2003. Online. Disponível na Internet em: <http://ksi.cpsc.ucalgary.ca/articles/CSCL95CM/>.

GOLDBERG, M.W.; SALARI, S. An update on WebCT (world-wide-web course tools) – a tool for the creation of sophisticated Web-based learning environments. In: NAUWEB'97: CURRENT PRACTICES IN WEB BASED COURSE DEVELOPMENT, Arizona, 1997. **Proceedings**. Capturado em 03 dez. 2001. Online. Disponível na Internet em: <http://star.ucc.nau.edu/~nauweb97/papers/goldberg/goldberg.html>.

JAQUES, P.; OLIVEIRA, F. M. de. Um Experimento com Agentes de Software para Monitorar a Colaboração em Aulas Virtuais. In: WORKSHOP DE INFORMÁTICA NA EDUCAÇÃO (WIE), 6, Curitiba, 2000. **Anais**. Capturado em 01 jan. 2000. Online. Disponível na Internet em: http://www.dcc.unicamp.br/~rlachi/Artigo_Jaques_Oliveira_2000.htm.

JENNINGS, N. R.; WOOLDRIDGE, M. J. Applications of Intelligent Agents. In: JENNINGS, N.R.; WOOLDRIDGE, M.J. (Eds.) **Agent Technology Foundations, Applications and Markets**. Springer-Verlag, 1998. cap.1, p.3-72.

- JOACHIMS, T. **A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization.** Carnegie Mellon University, 1996. 26p. (Computer Science Technical Report CMU-CS-96-118).
- KAY, A. User Interface: A Personal View. In: LAUREL, B.; MOUNTFORD, S.J. (Eds.) **The Art of Human-Computer Interface Design.** New York: Addison-Wesley, 1990. p.191-208.
- LAI, K.; MALONE, T.; YU, K. Object Lens: A “spreadsheet” for cooperative work. **ACM Transactions Office Information System**, v.6, n.4, p.332-353, Apr. 1988.
- LANG, K. Newsweeder: Learning to filter netnews. In: PRIEDITIS; RUSSEL (Eds.) **Proceedings of the International Conference on Machine Learning**, 12, San Francisco: Morgan Kaufmann Publishers, 1995. **Proceedings.** p.331-339.
- LAVRAC, N., DZEROSKI, S. **Inductive logic programming: Techniques and applications.** Chichester, UK: Ellis Horwood, 1994. 293p.
- LEWIS, D. Representation and learning in information retrieval. Massachusetts, 1991. 205p. Thesis (Ph.D.) – Dept. of Computer and Information Science, University of Massachusetts. Capturado em 08 set. 2001. Online. Disponível na Internet em: http://www.cs.umass.edu/Dienst/UI/2.0/Describe/ncstrl.umassa_cs%2FUM-CS-1991-093.
- LIMA, V.L.S.; ABRAHÃO, P. R.; PARABONI, I. Approaching the dictionary in the implementation of a natural language processing system: toward a distributed structure. In: **SOUTH AMERICAN WORKSHOP ON STRING PROCESSING**, 4, Valparaíso, 1997. **Anais.** Montreal: McGill-Queen's University Press, 1997. p.166-178.

- MAES, P. Agents that Reduce Work and Information Overload. In: MAYBURY, M.; WAHLSTER, W.(Eds.) **Readings in Intelligent User Interfaces**. San Francisco: Morgan Kaufmann, 1998. parte 7, p.525-535.
- MICHIE, D.; SPIEGELHALTER, D. J.; TAYLOR, C. C. Machine learning, neural and statistical classification. New York: Ellis Horwood, 1994. 290p. Capturado em 21 nov. 2001. Online. Disponível na Internet em: <http://www.amsta.leeds.ac.uk/~charles/statlog/>.
- MITCHELL, T. M. **Machine Learning**. Massachussets: McGraw-Hill Science/Engineering/Math, 1997. 432p.
- MUSA, D. L.; OLIVEIRA, J. P. M. de O. Alertas Inteligentes na Educação à Distância. 7ª Semana Acadêmica do programa de pós-graduação em computação da UFRGS, Rio Grande do Sul, 2001.
- MYERS, B.A.; CYPHER, A.; MAULSBY, D.; SMITH, D.C.; SHNEIDERMAN, B. Demonstrational Interfaces: Coming Soon? In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, New Orleans: ACM Press, 1991. **Proceedings**. p.393-396
- NEGROPONTE, N. Agents: From Direct Manipulation to Delegation. In: BRADSHAW, J.M. (Ed.) **Software Agents**. California: AAAI Press, 1997. cap.3, p.57-66
- NOVAK, J.D. **A Theory of Education**. Ithaca, Illinois: Cornell University Press, 1977.
- NWANA, H. S. Software Agents: An Overview. **Journal of Knowledge Engineering review**. v.11, n.2, p.205-244, 1996. Capturado em out. 2001. Online. Disponível na Internet em: <http://citeseer.nj.nec.com/nwana96software.html>.

PIMENTEL, M. G.; SAMPAIO, F. F. HiperDiálogo: uma ferramenta de bate-papo para diminuir a perda de co-texto. In: XII SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, Espírito Santo, 2001. **Anais**. Vitória: UFES, 2001. p.255-266.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. Englewood Cliffs, NJ: Prentice Hall, 2002. 1132p.

SHOHAM, Y. An Overview of Agent-oriented Programming. In: BRADSHAW, J.M. (Eds.) **Software Agents**. California: AAAI Press, 1997. cap.13, p.271-290.

SULLIVAN, J.W.; TYLER, S.W. **Intelligent User Interfaces**. Massachussets: Addison-Wesley, 1991. 500p.

TELEDUC (2000). **Ambiente de suporte para ensino-aprendizagem a distância**. Visitado em 29 ago. 2001. Online. Disponível na Internet em: <http://teleduc.nied.unicamp.br/teleduc/>.

WEBCT (2000). **The e-Learning hub**. Visitado em 06 dez. 2001. Online. Disponível na Internet em: <http://www.webct.com>.