Relatório Técnico

Núcleo de Computação Eletrônica

- Agente Fenix - Sistema de Filtragem Personalizada de Informações

Flávia Coimbra Delicato
Luci Pirmez
Luiz Fernando Rust da Costa Carmo

NCE - 37/99

Universidade Federal do Rio de Janeiro

Agente Fenix - Sistema de Filtragem Personalizada de Informações

Flávia Coimbra Delicato*, Luci Pirmez e Luiz Fernando Rust da Costa Carmo
NCE/UFRJ - Núcleo de Computação Eletrônica - Universidade Federal do Rio de janeiro
Tel: 021 5983159 - Caixa Postal: 2324 Rio de Janeiro RJ Brasil
E-mails: flavia@eng.uerj.br, luci, rust@nce.ufrj.br

Resumo

Atualmente, a Internet disponibiliza uma extensa quantidade de informações, para uma vasta gama de usuários, tornando-se difícil de manipular. Inspirado nesta dificuldade, o presente trabalho propõe o uso de Agentes Inteligentes para a filtragem personalizada de informações. A idéia é desenvolver um conjunto de agentes autônomos, fixos e adaptativos, com o objetivo de satisfazer as necessidades de informação do usuário. Para a representação das informações, será adotado o modelo vetor espacial [Salton83], e o mecanismo de aprendizagem dos agentes será o feedback de relevância [Frakes92]. Esse artigo apresenta a descrição do sistema e os resultados de uma primeira etapa, onde foram feitos testes em um ambiente simulado. Em uma próxima etapa, serão conduzidos testes com usuários reais. O espaço de busca por onde serão recuperadas as informações são documentos existentes nas páginas da WWW. O sistema foi desenvolvido para ambientes mono-usuários, e o público-alvo serão professores, alunos e funcionários de uma Universidade.

Abstract

Nowadays Internet offers such an extensive amount of information, that it becames very difficult for the users to take advantage of it. Searching of a solution, the present work proposes a methodology of implementation of an Intelligent Agent for the personalized information filtering. The idea is to develop a set of autonomous, non-mobile, adaptative agents.The learning mechanism adopted by the agents is "relevance feedback". The search space where the information will be recovered from are the articles and works found in the WWW pages, selected and classified in agreement with subjects of interest. The system was developed to help the academic public, composed by teachers, graduation and masters degree students. The initial results were quite promising. The proposed agent demonstrated to be a powerful tool of information filtering, reducing the time wasted in that activity.

1 INTRODUÇÃO

Atualmente, várias aplicações estão sendo desenvolvidas na área de redes de computadores, mais especificamente, no âmbito da rede mundial Internet. Sistemas como o "World Wide Web" (WWW) tornaram a Internet acessível a uma gama muito grande de usuários leigos. A todo momento, estão surgindo servidores de informação oferecendo os mais diversos tipos de dados; pesquisadores estão tentando encontrar meios confiáveis para o pagamento eletrônico, tornando a rede um importante "mercado virtual". A grande quantidade de informações disponíveis torna a rede difícil de se manipular. Surgem questões sobre como os usuários serão capazes de localizar a informação que eles precisam, ou como poderão encontrar a melhor oferta para um determinado serviço. Uma possível solução para este problema consiste no uso de agentes.

*Aluna do curso de Mestrado em Informática do Instituto de Matemática da UFRJ e funcionária da UERJ

Inspirado nesta necessidade, o presente trabalho propõe o uso de Agentes Inteligentes para a filtragem personalizada de informações. O sistema proposto será formado por um conjunto de agentes autônomo, adaptativos, fixos, localizados na máquina do usuário, e com o objetivo de satisfazer suas necessidades de informação. Os agentes devem receber feedback do usuário sobre a relevância das informações recuperadas e refinar sua busca, obtendo resultados melhores ao longo do tempo.

Este conjunto de agentes é autônomo pelo fato de poder executar sua tarefa sem a presença do usuário, e de realizar filtragens de assuntos novos e potencialmente interessantes, de acordo com o perfil de cada usuário. É adaptativo, porque aprende as preferências do usuário e adapta-se quando as mesmas mudam ao longo do tempo. O mecanismo de aprendizagem adotado pelo agente é o feedback de relevância ("relevance feedback"), amplamente utilizado em sistemas de recuperação de informações [Frakes92].

O sistema desenvolvido será voltado para ambientes mono-usuários e acadêmicos, compostos de professores, alunos de graduação e pós-graduação, e funcionários de nível especializado, dentro de uma Universidade. O espaço de busca a partir do qual serão recuperadas as informações são artigos e trabalhos existentes nas páginas da WWW, selecionados e classificados de acordo com assuntos de interesse.

Este trabalho está organizado da seguinte forma: a seção 2 apresenta uma descrição dos conceitos e técnicas utilizados ao longo do desenvolvimento do trabalho. Nessa seção, serão abordados alguns conceitos relacionados com Agentes, com as técnicas de busca e de filtragem de informações, e com o método de feedback de relevância usado no trabalho. Na seção 3, será feita uma comparação com trabalhos anteriores que tratam de assunto semelhante ao abordado no presente trabalho. A seção 4 descreve a arquitetura do sistema desenvolvido, detalhando cada um dos seus componentes. A seção 5 apresenta a análise dos resultados e, finalmente, a conclusão será dada na seção 6.

2 CONCEITOS E TÉCNICAS UTILIZADOS

A seguir serão introduzidos alguns conceitos necessários para o entendimento do presente trabalho.

2.1 AGENTES

Segundo o Webster's New World Dictionary [Guralmile70], a definição de agente é: "uma pessoa ou coisa que age ou é capaz de agir ou é autorizado a agir, por outra." No entanto, é óbvio que o significado da palavra poderá variar de acordo com o contexto em que se insira. A sua inserção no contexto da informática tem-se tornado, nos últimos anos, cada vez mais popular. Mas, neste contexto, a palavra "agente" não tem uma definição consensual, talvez por cobrir diversas áreas de investigação e desenvolvimento. Uma vez que as definições são numerosas, a seguir são analisadas algumas delas.

A definição dada por Nwana [Nwana96] dá ênfase à delegação de tarefas, assemelhando-se bastante à definição comum da palavra "agente": "(...) nós definimos um agente como um componente de software e/ou hardware que é capaz de agir exatamente com o objetivo de realizar tarefas em benefício de seu usuário."

Genesereth e Ketchpel, em [Ketchpel94], restringem a sua definição aos agentes de *software*. Para estes autores, o essencial nos agentes de *software* é a sua capacidade para se comunicarem entre si numa linguagem expressiva: "Softwares Agentes são componentes de software que se comunicam com seus pares trocando mensagens em uma linguagem de comunicação de agentes expressiva".

Shoham, em [Shoham93], dá ênfase à autonomia e à continuidade no tempo, afirmando, ainda, que um agente não tem sentido se não existirem outros agentes: "Mais freqüentemente, o termo agente é usado para referir-se a uma entidade que funciona contínua e autonomamente, em um ambiente no qual outros processos tomam lugar e onde existem outros agentes."

Como foi visto, cada uma destas definições propõe uma série de características que seriam essenciais para a definição do que é um agente. De uma maneira geral, pode-se dizer que a definição de um agente destaca dois atributos:

- um agente faz alguma coisa;
- um agente age em benefício de alguém ou de alguma coisa.

Agentes que residem em computadores sempre incorporam esses atributos centrais [Caglayan97]. No presente trabalho, um agente software será definido como "uma entidade computacional que realiza tarefas delegadas pelo usuário de forma autônoma" [Caglayan97].

Além das características essenciais citadas, existem ainda outras que costumam surgir associadas aos agentes, como, por exemplo, a mobilidade. Conforme estas características estejam presentes ou não, os candidatos a agentes podem ir de uma simples macro até um ser humano. A seguir serão analisadas algumas dessas características, já que elas são cruciais para compreender o mundo dos agentes:

- **Inteligência** - é o grau de raciocínio e comportamento aprendido; a habilidade do agente de aceitar a sentença de objetivos (metas) do usuário e desempenhar a sua tarefa.
- **Autonomia** - é a capacidade do agente de ter controle sobre as suas ações, e de decidir como satisfazer o pedido que recebe, sendo ainda suficientemente flexível para decidir, de forma dinâmica, quais as ações que irá tomar e qual o momento mais apropriado para as tomar.
- **Continuidade temporal** - a continuidade temporal impõe que o agente esteja continuamente ativo. Nota-se que grande parte do *software* existente não tem esta característica, já que executa uma ou mais tarefas e termina.
- **Capacidade social** - se um agente tem capacidade social, então esse agente comunica-se com outros agentes, o que poderá incluir humanos. Dessa comunicação, que será feita utilizando uma linguagem de comunicação entre agentes, poderá resultar uma cooperação.
- **Capacidade de adaptação** - um agente com capacidade de adaptação é capaz de alterar o seu comportamento com base na experiência. Assim, diz-se também que esse agente tem capacidade de aprendizagem.
- **Mobilidade** - corresponde à capacidade do agente de se mover no ambiente. Quando se trata de agentes de *software*, um agente móvel é aquele que é capaz de se transportar de uma máquina para outra durante a sua execução.

Para efetuar as tarefas que lhes foram delegadas, os agentes podem ou não usar as capacidades supracitadas e é em função da utilização dessas capacidades que eles podem receber outras denominações, tais como: agentes fixos, agentes móveis, agentes inteligentes, agentes móveis inteligentes.

Os agentes podem também ser classificados de acordo com as tarefas que executam. Atualmente, algumas das tarefas em que estão sendo usados agentes (ainda que se trate, em alguns casos, apenas de protótipos) são:

- filtragem de informação, por exemplo, de mensagens de correio eletrônico e de "news";
- recuperação de informação (permitindo, entre muitas outras coisas, encontrar páginas pessoais na WWW);

- recomendação, por exemplo, para auxiliar na navegação na WWW;
- assistência do utilizador (por exemplo, através da gestão de horários e a calendarização de reuniões);
- comércio eletrônico (por exemplo, permitindo a utilização de vários métodos de pagamento através de uma só interface ou ajudando o utilizador a encontrar o produto que pretende);
- educação e o treinamento;
- gestão de redes;
- entretenimento (por exemplo, na área dos jogos e da produção de vídeos).

2.2 MECANISMOS DE BUSCA DE INFORMAÇÕES

Os mecanismos de busca existentes na World Wide Web são a forma mais usual para auxiliar as pesquisas na Internet. Estes mecanismos são, certamente, uma boa forma para encontrar documentos que contenham determinadas palavras, designadas por palavras-chaves. No entanto, eles possuem diversos problemas:

- Assumem que o usuário sabe escolher o melhor conjunto de palavras-chaves para a obtenção dos resultados mais relevantes;
- Procuram a informação em um contexto genérico, surgindo como resposta várias informações irrelevantes;
- mapeamento da informação é feito reunindo-se meta-informações sobre informações e documentos disponíveis na Internet. Este método consome muito tempo e gera muito tráfego de dados devido a falta de eficiência nos mecanismos de busca tradicionais;
- É necessário um servidor poderoso para efetuar a indexação dos muitos documentos existentes, guardar o índice e suportar a pesquisa sobre esse índice para muitos usuários. Este problema se agrava à medida que o número de documentos aumenta;
- A informação na Internet é muito dinâmica. Freqüentemente, mecanismos de busca fazem referência a informações que foram transferidas para outro local. Os mecanismos atuais não aprendem através de suas buscas e nem se ajustam a seus usuários;
- servidor onde reside o serviço de busca pode estar fora do ar ou demasiadamente ocupado para permitir uma conexão;
- É necessário que o usuário mantenha uma ligação à Internet durante todo o tempo em que a pesquisa é efetuada .

Nesse contexto, os agentes oferecem algumas vantagens quando comparados com os métodos correntes de recuperação de informações:

- podem buscar informações baseados em contextos semânticos, utilizando ferramentas como tesaurus;
- podem criar suas próprias bases de dados sobre a disponibilidade de informações na Internet, atualizando-a e expandindo-a após cada busca. Quando informações tiverem mudado para outro local, agentes são capazes de encontrá-las e atualizar suas bases. Além disso, agentes podem se comunicar e cooperar com outros agentes, o que capacita-os a realizar a buscas de modo mais rápido e eficiente, reduzindo o tráfego na rede. Eles também são capazes de realizar a busca diretamente na fonte/servidor, levando a uma queda ainda maior no uso da banda passante;
- podem estar sempre disponíveis, quando residem na própria máquina do usuário;
- podem se ajustar de acordo com as preferências e hábitos de seus usuários individuais, aprendendo com eles, e fornecendo um serviço personalizado.

2.3 SISTEMAS DE FILTRAGEM DE INFORMAÇÕES

Paralelamente ao conceito de recuperação de informações, existe o conceito de filtragem, no qual o usuário é muito mais passivo. Geralmente, um sistema de filtragem de informações envolve repetidas interações ao longo de múltiplas sessões. Nesse caso, os usuários possuem objetivos a longo prazo. Isso contrasta com os mecanismos de busca de informações, onde os usuários tipicamente têm uma necessidade de informação a curto prazo, satisfeita em uma única sessão.

Um sistema de filtragem de informações assiste ao usuário filtrando o fluxo de dados e liberando apenas as informações relevantes. As preferências de informações variam muito de acordo com o usuário, portanto, esses sistemas devem ser altamente personalizados. Um exemplo desses sistemas é a criação de jornais personalizados, já existentes na WWW. Por exemplo, o NewsHound analisa diversos jornais para construir um jornal personalizado, de acordo com as preferências indicadas por cada utilizador. Esse jornal é depois enviado ao utilizador através de correio eletrônico. Um sistema de filtragem personalizado deve satisfazer 3 requisitos:

- Especialização: uma vez que a filtragem envolve interações repetidas com o usuário, o sistema deve ser capaz de identificar padrões no seu comportamento; o sistema deve inferir seus hábitos e especializar-se a eles, isto é, recomendar o máximo de assuntos relevantes e o mínimo de irrelevantes;
- Adaptação: Os interesses do usuário não devem ser considerados constantes; quando esses mudarem, o sistema deve ser capaz de perceber e adaptar seu comportamento a essas mudanças;
- Exploração: um sistema de filtragem deve ser capaz de explorar novos domínios de informações para encontrar assuntos de interesse potencial para o usuário.

Atualmente, encontram-se na literatura, três abordagens distintas para os sistemas de filtragem de informações:

- sistemas que se baseiam no perfil do usuário para filtrar as informações, tentando adequá-las aos seus interesses e expectativas [Brusilovski94];
- sistemas que realizam a filtragem de forma cooperativa, compartilhando informações [Twidale95]; e
- sistemas que utilizam agentes, nos quais a mobilidade, autonomia e habilidade de interagir independentemente da presença de seus usuários são fatores fundamentais [Nissen95].

De uma maneira geral, os sistemas de filtragem que utilizam o conteúdo semântico das informações para decidir se as mesmas são relevantes ou não, são classificados como sistemas de filtragem cognitivos.

2.4 FEEDBACK DE RELEVÂNCIA

Uma das operações mais importantes e difíceis em recuperação de informações é gerar consultas ("query") que possam identificar sucintamente documentos relevantes e rejeitar os irrelevantes. Os usuários freqüentemente submetem consultas contendo termos que não equivalem aos termos usados para indexar uma grande parte dos documentos relevantes, e, quase sempre, vários documentos relevantes não recuperados são indexados por termos diferentes daqueles da consulta ou da maioria dos outros documentos relevantes. Esse problema tem sido reconhecido como uma das principais dificuldades em sistemas de recuperação de informações [Lancaster69]. Em 1986, van Rijsbergen [Rijsbergen86] falou

dos limites de fornecer resultados cada vez melhores baseados apenas na consulta inicial e indicou a necessidade de modificar aquela consulta para um aumento adicional de performance.

Uma vez que se reconhece a dificuldade de realizar uma pesquisa com sucesso na primeira tentativa, costumam-se conduzir buscas iterativamente, e reformular as sentenças da consulta baseando-se na avaliação dos documentos previamente recuperados. Um método para gerar automaticamente formulações de consultas melhoradas é o conhecido processo de feedback de relevância [Rocchio71], [Salton89]. Uma consulta pode ser melhorada iterativamente usando-se um vetor (de termos) de consulta disponível e adicionando termos a partir de documentos relevantes, enquanto se subtraem termos a partir de documentos irrelevantes. Uma única iteração de feedback de relevância freqüentemente produz melhoras de 40 a 60% na precisão da busca [Salton89]. Uma abordagem similar pode também ser usada para alterar a representação do documento.

3 COMPARAÇÃO COM TRABALHOS ANTERIORES

Algumas semelhanças e diferenças podem ser identificadas ao comparar a abordagem proposta com abordagens anteriores no contexto de filtragem de informações. O problema de filtragem e classificação de documentos situa-se na interseção entre Máquinas de Aprendizado ("Machine Learning" ou ML) e Recuperação de Informações ("Information Retrieval" ou IR), havendo, portanto, uma vasta literatura a respeito.

Na comunidade de IR, variantes de feedback de relevância têm sido estudadas no contexto da tarefa de roteamento de informações, nas conferências TREC [Harman95], como por exemplo [Buckley95] [Allan95]. Há, também, várias comparações entre essas variantes e técnicas de ML não-incrementais [Schutze95] [Lang95]. Uma das desvantagens de tais técnicas é o grande número de exemplos necessários antes do algoritmos de aprendizado poder ser aplicado [Balabanovic97]. Em contraste, no presente trabalho é adotado um modelo no qual o usuário vê páginas gradualmente melhores, com seu feedback influenciando quais páginas serão apresentadas. Alguns trabalhos de filtragem de informações que também utilizam o modelo citado são [Sheth93] e [Foltz92]. Outros sistemas, ainda, usam diversas técnicas para tentar detectar padrões no comportamento do usuário. Por exemplo, InfoScope [Fischer91] aprende usando sistemas baseados em regras os quais lembram-se de tópicos interessantes cobertos no passado. Novas recomendações de tópicos para o usuário são feitas com base em quão recentes, freqüentes e espaçados são os tópicos passados. A desvantagem dessa abordagem é que ela é restrita a fazer recomendações de tópicos que estavam dentro do domínio de interesses passados do usuário. Da mesma forma, sistemas de navegação assistida [Armstrong95], que compartilham o objetivo do presente trabalho de recomendar páginas WWW, são restritos às seções da Web visitadas pelo usuário, recomendando, a partir delas, links apropriados para seguir. Em contraste, no sistema proposto, o agente busca novos domínios para informações que podem ser de interesse potencial para o usuário. É bem possível que o usuário nunca tenha visto o tópico apresentado antes.

O sistema de News-t [Sheth] usa feedback de relevância e algoritmos genéticos (AG) para fornecer filtragem personalizada de artigos da USENET. A abordagem difere principalmente no espaço de busca, pois, ao considerar a filtragem de páginas WWW em vez de artigos, há grande impacto no modelo de representação dos documentos e consultas, e na adoção do AG para otimizar os resultados obtidos apenas com o processo de feedback de relevância.

4 DESCRIÇÃO DO SISTEMA

O presente trabalho propõe o uso de Agentes Inteligentes e não móveis para a filtragem personalizada de informações. O modelo adotado para a representação das informações será o vetor-espacial [Salton83]. O mecanismo de aprendizagem adotado pelos agentes será o feedback de relevância. Os perfis usados para filtrar as informações baseiam-se em termos que são comparados com os conteúdos dos documentos pesquisados. Dessa forma, o sistema proposto pode ser classificado como cognitivo.

No sistema de filtragem de informações proposto, um agente é modelado como uma conjunto de perfis individuais, onde cada perfil é utilizado para a busca de artigos em um pequeno domínio. Enquanto cada perfil satisfaz tipicamente uma pequena parcela dos interesse do usuário, o conjunto de todos os perfis de um agente tenta atender aos interesses completos de um usuário em determinado assunto e adaptar-se a eles. Um usuário pode ter vários agentes, cada um atendendo às suas necessidades de informação em um determinado assunto.

Os perfis dos agentes buscam documentos que são similares a eles. Aqueles com maior grau de similaridade são recuperados e apresentados ao usuário, que pode, então, fornecer feedback positivo ou negativo. O feedback do usuário tem o efeito de modificar o perfil usado para recuperar aquele documento, baseado na sua relevância.

A linguagem de programação adotada para a implementação do agente foi Java, e o ambiente de desenvolvimento foi o Jbuilder, da Borland Corporation. Sua escolha baseou-se principalmente no fato de Java ser uma linguagem portátil e voltada para ambientes de Internet. Os perfis foram representados por instâncias de uma classe Perfil e os agentes foram implementados como subclasses da classe Thread de Java, que representa uma tarefa do sistema operacional. Os dados persistentes (informações sobre páginas, usuários, representações vetoriais dos documentos que compõem os perfis e outras) foram armazenados em classes de arquivos de acesso aleatório.

Nessa primeira versão do sistema, optou-se por fazer um aplicativo, o qual poderá ser obtido através de um download realizado na página do agente, em vez de uma applet. Essa escolha foi devido às restrições de segurança dos browsers atuais, que não permitem a criação de arquivos na máquina do usuário. Como não é adotado um sistema de filtragem colaborativa, no qual os perfis dos diferentes usuários devem ser compartilhados, os arquivos devem preferencialmente ser gravados localmente. Em uma versão posterior, pretende-se fazer uma applet, que irá rodar a partir da página do agente, e irá gravar os arquivos em um servidor de arquivos remoto.

A seguir, será descrita em detalhes a arquitetura do sistema proposto. O agente Fenix é formado essencialmente pelos seguintes componentes (fig. 1):

- módulo de interface com o usuário: permite que o usuário crie seus agentes para assuntos específicos; carregue agentes já existentes, leia artigos recuperados e forneça feedback para artigos lidos;
- módulo de aprendizado: Esse módulo é responsável por garantir que o conjunto de perfis reflita os interesses dinâmicos do usuário. Cada perfil é armazenado como um arquivo de acesso aleatório definido a partir da classe RandomAccessFile da linguagem Java. O método de aprendizagem adotado é o feedback de relevância.
- módulo de filtragem das informações: utilizando os perfis do usuário como sua entrada, é responsável por encontrar nas páginas WWW, documentos semelhantes aos perfis. O processo de filtragem consiste em transformar documentos em suas representações vetoriais, calcular valores de similaridade (score) entre documentos e perfis, ordenar os

documentos de acordo com seu score, e apresentar ao usuário apenas aqueles com score maior que um limite pré-fixado;

- módulo de busca e indexação de informação: responsável por reunir uma certa quantidade de informações da rede WWW e salvá-las em um banco de dados local;
- banco de dados local: conjunto de arquivos nos formatos ASCII e binário, contendo os dados dos usuários, os seus respectivos perfis e os endereços (URLs) das páginas capturadas na rede WWW.

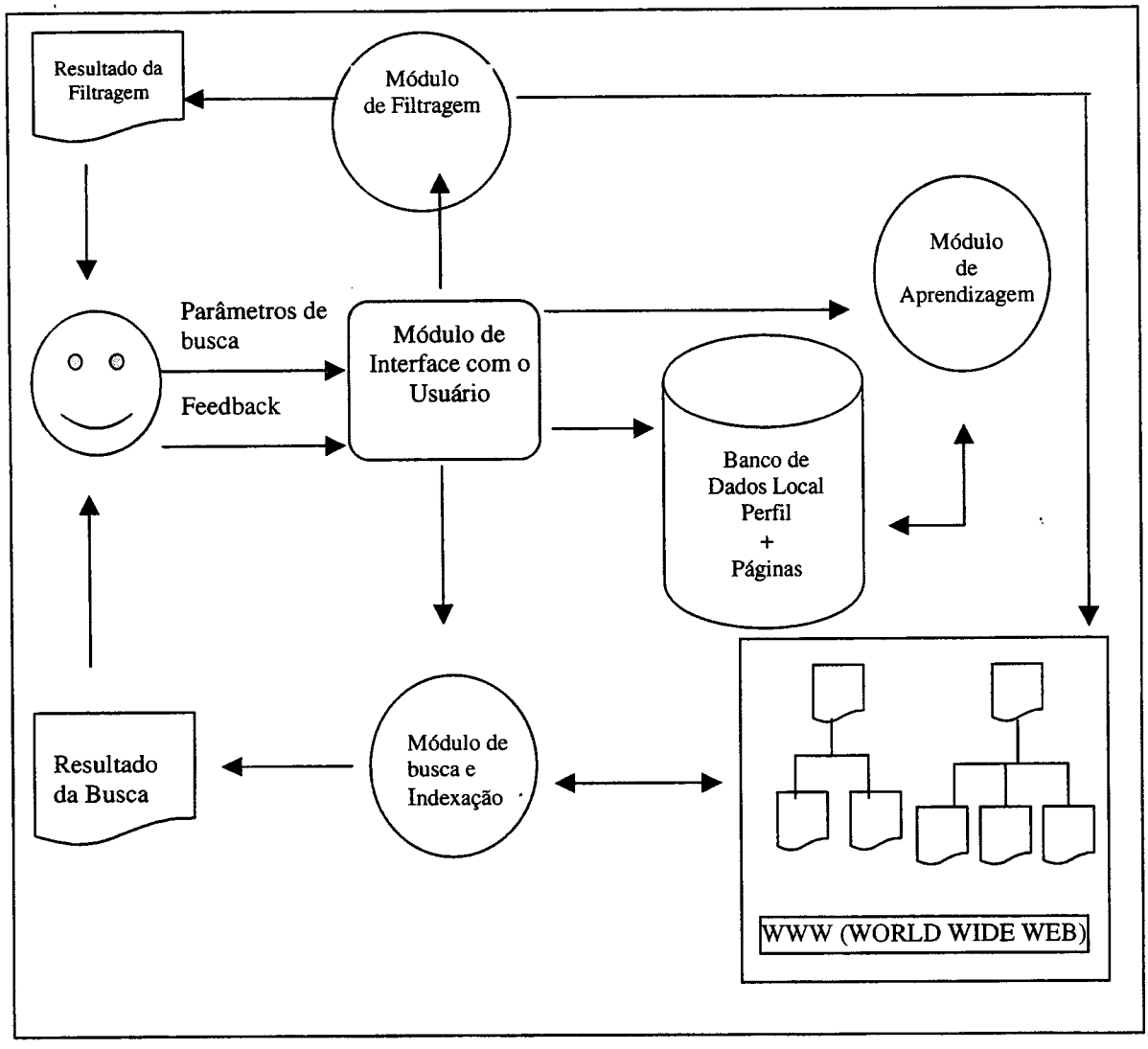

Figura 1: Arquitetura do Sistema

A seguir, será feita a descrição de cada um dos componentes.

4.1 MÓDULO DE INTERFACE COM O USUÁRIO

Após se identificar, o usuário pode escolher entre duas opções: criar um novo agente, ou carregar um agente já existente.

Ao criar um novo agente, o usuário deve escolher um nome para ele, e fornecer os seguintes parâmetros de busca: número máximo de páginas por documento, profundidade de

links por página, número máximo de artigos a ser recuperado por sessão, e a expressão de consulta. Uma consulta no sistema Fenix é uma combinação de palavras-chaves (tecnicamente chamadas de *termos*) e do conectivo lógico &. Como resultado da busca inicial, será apresentada uma lista com os nomes dos artigos recuperados. Após a leitura dos artigos desejados, o usuário pode fornecer feedback positivo ou negativo conforme sua relevância para aquele assunto específico. Ao terminar a sessão, o usuário pode ou não salvar o agente recém-criado (serão salvos os artigos e os perfis referentes àquele assunto e ao usuário específico).

Ao se identificar e escolher a opção de carregar um agente existente, o usuário seleciona um dentre uma lista de agentes pertencentes a ele. A seguir, pode escolher uma das seguintes ações:

- ler algum artigo recuperado;
- fornecer feedback sobre algum artigo lido;
- iniciar uma nova busca por artigos, referente ao assunto específico;

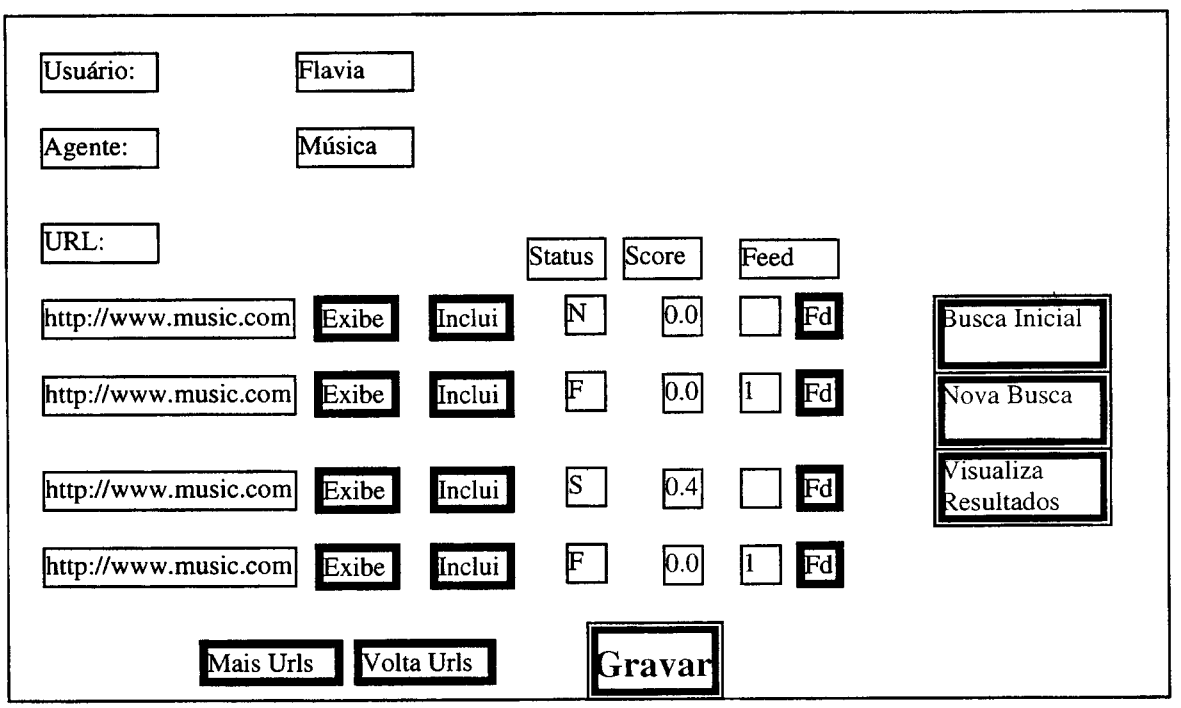

Figura 2: Janela do Agente. O botão "Exibe" chama o browser local para o usuário visualizar o documento. A letra "F" indica artigos lidos e avaliados pelo usuário, "N" indica artigos não lidos, e "S" artigos já classificados pelo agente. O campo "Sc" contém a pontuação dada ao artigo pelo agente. O campo "Feed" corresponde ao feedback do artigo.

4.2 MÓDULO DE BUSCA DE INFORMAÇÕES

Funciona de forma semelhante aos mecanismos de busca na WEB baseados em palavras-chaves ("Web Crawlers"). Através da utilização de classes pré-definidas da linguagem de programação adotada, percorre as páginas WWW procurando pelas palavras-chaves indicadas. Cada página possui links para outras, as quais são também recuperadas recursivamente, até o limite fixado pelo usuário. A página raiz e as obtidas através dos links são armazenadas na forma de uma string única. O local de origem das páginas (URL) também é armazenado e, juntamente com outras informações, irá compor o perfil do usuário.

4.3 MÓDULO DE FILTRAGEM

O processo de filtragem consiste em transformar documentos em suas representações vetor-espaciais, encontrar documentos que sejam similares aos perfis e selecionar aqueles de maior pontuação para apresentar ao usuário.

Descrição do módulo:

Após uma busca inicial, baseada nas palavras-chaves fornecidas pelo usuário, o sistema apresenta uma lista de documentos para serem lidos e avaliados pelo usuário. Os artigos que receberem feedback positivo serão convertidos para a sua representação vetorial e irão compor o perfil do usuário. Além dos artigos, suas respectivas URLs também são armazenadas no perfil.

Nas buscas posteriores, cada novo artigo recuperado será convertido para sua representação vetorial, será calculada sua similaridade com relação ao perfil do usuário, e um score lhe será atribuído. Apenas artigos com score maior do que um limite fixado serão apresentados.

Representando perfis e documentos

A representação adotada neste trabalho para perfis e documentos é baseada na representação vetor espacial [Salton83] comumente usada em sistemas de recuperação de informações. Nela, tanto os documentos como as consultas são representados como vetores em um hiper-espaço. Uma distância métrica, que mede a proximidade entre os vetores, é definida sobre o espaço. Cada perfil de usuário é convertido para sua representação espacial, bem como os documentos recuperados ao longo das sessões. Os documentos cujas representações vetoriais possuem maior grau de proximidade do vetor perfil são os resultados da filtragem. Assim, cada perfil é usado para pesquisar as páginas WWW em busca de artigos que são similares a ele.

Documentos

Um método padrão de indexar texto consiste em remover marcas de pontuação, reconhecer palavras individuais, eliminar as palavras comumente usadas incluídas em uma lista de exclusão de palavras (como "e", "que", etc), e usar as palavras restantes para identificação do conteúdo do texto. Às vezes, frases são compactadas e tratadas como um único termo. As palavras também podem ser truncadas, permanecendo apenas seus radicais. Uma vez que os termos não são todos igualmente importantes para a representação do conteúdo, pesos são associados aos termos, na proporção de sua presumida importância para a identificação do conteúdo do texto. Um texto é, então, representado como um vetor de termos Ti = < Wij >, onde Wij representa o peso do termo tj no texto Ti.

Um documento WWW pode conter vários campos em seu cabeçalho, como URL, autor, local, etc, seguidos pelo texto que compõe o corpo da página. Nesse trabalho, serão considerados relevantes apenas o campo URL e o texto em si. O texto é aqui denominado de campo palavra-chave, e consiste na área delimitada pelas tags </BODY> da página.

Na representação adotada, o peso de um termo palavra-chave é o produto de sua frequência de termo e seu inverso da frequência de documento. A frequência de termo (tf) é a frequência de ocorrência do termo no texto e normalmente reflete a importância desse termo. O inverso da frequência do documento (idf) é um fator que realça os termos que aparecem em poucos documentos, enquanto desvaloriza os termos que ocorrem em muitos documentos. O efeito resultante é que as feições específicas do documento ficam destacadas, enquanto as

feições espalhadas pelo conjunto de documentos têm menor importância. O peso dos termos é, então, dado por:

$$\mathbf{Wij = tf_{ik} \ X \ idf_k, \quad (1)}$$

onde tf_{ik} é o número de ocorrências do termo t_k no documento i, e idf_k é o inverso da frequência de documento do termo t_k na coleção de documentos. Uma medida comumente usada para idf é $idf_k = \log (N / n K)$, onde N é o número total de documentos na coleção, dos quais nK contêm um termo t_k.

Perfis

Um perfil consiste de um conjunto de informações sobre os artigos recuperados e avaliados, como por exemplo, sua localização na rede (URL) e o feedback associado a ele. Além disso, contém a representação vetorial dos artigos que receberam feedback positivo do usuário. A representação vetorial consiste em um vetor de termos semelhante ao descrito anteriormente para documentos. Então:

$$\mathbf{P = (Fi^p) \ , onde \ Fi^p = < Wij^p >, \quad (2)}$$

onde "p" indica que é um campo perfil e não um campo documento.

Extraindo Representações de Documentos

Um conjunto ou coleção de documentos consiste de todos os documentos recuperados por um perfil.

O algoritmo de indexação de texto foi implementado como um processo de 2 estágios. No primeiro estágio, são calculados os tf 's. A contagem da frequência das palavras é calculada e armazenada. Esse processo é repetido para cada um dos N documentos.

No segundo estágio, são calculados os idf 's. Uma segunda análise sobre os dados armazenados fornece as contagens de documentos para todos os termos, isto é, o número de documentos em que cada termo aparece. O produto de tf e idf dá os pesos dos termos, conforme descrito na equação (1), e o vetor termo para cada documento pode ser gerado.

Avaliação dos Documentos Filtrados

Na clássica representação vetor-espacial, documentos são recuperados encontrando-se vetores na proximidade do vetor consulta. Uma medida de similaridade comumente usada é o cosseno do ângulo entre vetores. Isso pode ser calculado através do produto escalar dos 2 vetores:

$$\mathbf{Similaridade \ (Vi, Vj) = \sum_k Wij \ X \ Wjk \quad (3)}$$

Na aplicação proposta, a similaridade entre um documento e um perfil é uma função da similaridade entre os vetores de termos dos documentos e dos perfis, conforme a equação abaixo:

$$\mathbf{S \ (F \ i^d, F \ i^p) = \sum_k W \ ik^d \ X \ W \ ik^p, \quad (4)}$$

onde "d" indica que o campo pertence a um documento e o "p" indica que o campo pertence ao perfil.

Os perfis são usados para percorrer diferentes partes do banco de dados e pontuar diferentes artigos. Os scores de similaridade de documentos dizem respeito a diferentes perfis. Quando o agente coleta todos os artigos de maior pontuação recuperados pelos perfis, não é possível comparar os scores a menos que eles estejam todos na mesma escala. Portanto, os scores de documentos devem ser normalizados de modo a ficarem restritos ao intervalo fechado [-1,1].

Atribuindo Pontos e Selecionando Documentos

As representações de todos os documentos a serem avaliados estão agora disponível. Os documentos que são finalmente apresentados para o usuário são selecionados dentre aqueles que obtiveram pontuação alta com respeito aos diferentes perfis. O valor máximo de similaridade é 1.0, e só ocorre quando a representação do perfil e do documento são idênticas. Nesse trabalho, foi adotado o limite de 0.2 como o valor de score mínimo que um documento deve ter para ser apresentado ao usuário.

Associando Feedback

O usuário pode se comunicar com o agente fornecendo feedback positivo (+1) ou negativo (-1) para os artigos recuperados por cada perfil. O feedback tem o efeito de modificar o perfil em questão. Na verdade, o perfil tem que incorporar as mudanças geradas pelo feedback do usuário antes que novos documentos possam ser avaliados.

O perfil mantém uma lista dos artigos com seus respectivos status. O status indica se um documento foi ou n"ao avaliado pelo usuário e/ou pelo sistema. Uma vez que o vetor de termos para um documento com feedback está disponível, o perfil é modificado conforme a equação (6) descrita no módulo de aprendizdo abaixo.

4.4 MÓDULO DE APRENDIZADO

Esse módulo é responsável por garantir que o conjunto de perfis reflita os interesses dinâmicos do usuário da melhor forma, recuperando o máximo de documentos relevantes e descartando os irrelevantes .

O método de aprendizagem adotado pelo sistema é o feedback de relevância, no qual o vetor de consulta original (representado pelos perfis) é alterado com base no feedback dado pelo usuário para os documentos recuperados por cada perfil.

Para representações vetor-espaciais, o método para reformulação de uma consulta em resposta ao feedback do usuário é ajuste vetorial. Uma vez que consultas e documentos são ambos vetores, o vetor de consulta é movido espacialmente para mais perto de vetores representando documentos que receberam feedback positivo, e para mais longe dos vetores representando documentos que receberam feedback negativo.

Considere um perfil P, que contribuiu com um documento D para apresentação ao usuário. O usuário fornece feedback, que é um inteiro positivo ou negativo. Cada termo no perfil é mudado na proporção do feedback recebido:

$$\mathbf{P = P + \alpha * f * D} \qquad (5)$$

isto é, o peso de cada termo é modificado proporcionalmente a taxa de aprendizado e ao feedback. α é a taxa de aprendizagem, que indica a sensibilidade do perfil ao feedback do usuário. Um valor típico para α é de 0.5. Assim, temos:

$$\forall \mathbf{i, k:}\ \mathbf{W}_{ik}^{p} = \mathbf{W}_{ik}^{p} + \alpha * f * \mathbf{W}_{ik}^{d} \qquad (6)$$

O efeito resultante é que, para aqueles termos já presentes no perfil, os pesos dos termos são modificados em proporção ao feedback. Os termos que já não estejam no perfil são adicionados ao perfil.

5 ANÁLISE DOS RESULTADOS

Antes de passar para a análise dos resultados obtidos nessa primeira etapa, é importante definir alguns conceitos usados em sistemas de recuperação de informações, que serão úteis na avaliação desses resultados :

- relevância: segundo Rijsbergen [Rijsbergen], este é o conceito central na área de recuperação de informações. É um relacionamento entre uma fonte de informações e um indivíduo a procura dessa fonte. Um item de informação em uma coleção é *relevante* quando esse item supre uma necessidade de informação de um usuário daquela coleção.
- precisão: é a proporção de documentos relevantes no total de documentos recuperados. Ou seja, é a taxa do número de documentos que são considerados relevantes para uma consulta em particular sobre o número total de documentos recuperados [Silva96].
- lembrança ("recall"): é a proporção de documentos relevantes recuperados, em relação a todos os documentos relevantes para uma consulta.
- "fallout": é a proporção de documentos não relevantes recuperados sobre todos os documentos não relevantes na coleção, para uma consulta [Silva96].

A noção de relevância, como usada em IR é problemática em geral, e, em particular, para o domínio aqui estudado. Estudos sobre o comportamento de usuários mostraram que os mesmos têm dificuldade em fazer julgamentos de relevância consistentes por um período longo de tempo, quando solicitados para avaliar documentos em uma escala de relevância absoluta [Lesk71]. Há, também, considerável discordância entre os julgamentos de diferentes usuários [Saracevic95]. A maneira usual de contornar esse problema é testar sistemas de IR sobre coleções padronizadas de documentos e consultas, e, dessa forma, tornar os valores de precisão e recall comparáveis. Uma vez que o domínio do presente trabalho é a Web, não se pode adotar o uso de uma coleção padronizada.

Em contraste com os resultados negativos dos estudos de usuários mencionados, mostrou-se que juízes humanos são bons para fazer julgamentos relativos dada uma coleção de documentos, e esses julgamentos serão consistentes ao longo do tempo e quando comparadas a outros juízes [Lesk71]. Dessa forma, optou-se nesse trabalho por adotar uma nova medida de performance, a qual requer apenas julgamentos relativos. A medida ndpm ("normalized distance-based performance measure"), proposta por Yao [Yao95], é uma distância, normalizada para estar entre 0 e 1, entre a classificação dada pelo usuário a um conjunto de documentos e a classificação dada pelo sistema. Isso irá fornecer uma medida relativa, que será mais apropriada do que precisão e recall. Além disso, comparar diretamente a classificação do usuário e a do sistema fornece uma medida monovalorada, mais simples de interpretar, do que gráficos de precisão e recall. Em adição a todas essas considerações, torna-se praticamente inviável medir, ou mesmo estimar o valor de recall, quando a coleção de documentos é a Web toda.

O sistema Fenix irá fornecer aos usuários vários documentos considerados relevantes. Com o feedback fornecido, o perfil do usuário vai sendo ajustado para refletir seus interesses atuais. Dessa forma, os documentos efetivamente apresentados ao usuário representam um estreito segmento do conjunto total de documentos recuperados na fase de busca, os quais são próximos uns dos outros em termos de preferência do usuário, tornando-se, assim, inadequados para uso com a medida descrita. Assim, foi adotado um esquema conforme sugerido em [Balabanovic97]. Uma lista especial de documentos é fornecida para um usuário simulado, que deve classificá-la de acordo com seus interesses por um determinado assunto. Essa lista é selecionada aleatoriamente a partir de vários documentos recuperados na Web. O sistema, então, calcula valores de pontuação de acordo com a similaridade dos documentos

em relação a um perfil já construído para aquele usuário. Esses valores de pontuação são usados para prever a classificação de cada documento. A classificação consiste em categorias [Balabanovic97]: Excelente - Bom - Neutro - Pobre - Terrível. Os scores calculados pelo sistema serão convertidos para cada categoria de acordo com uma faixa de valores. O resultado desejado é que a distância ndpm entre as classificações dadas pelo usuário e as do sistema diminua gradualmente com o tempo, conforme o perfil do usuário vai sendo ajustado.

Foram realizadas cinco sessões simuladas de interação "usuário"- agente. Após uma busca inicial, em cada sessão o agente classificava os documentos recuperados segundo as categorias acima, o "usuário" avaliava os documentos, dando seu valor de feedback e sua classificação. Com o feedback, o perfil do agente era ajustado para as busca posteriores, e com a classificação, era calculada a distância ndpm. Observou-se uma diminuição progressiva dessa distância ao longo das sessões, indicando que o agente foi se adaptando às preferências do "usuário", aumentando, assim, a probabilidade de recuperar um maior número de documentos relevantes, e descartar os irrelevantes.

6 CONCLUSÕES

Um Sistema de Filtragem de Informação deve ser capaz de especializar-se conforme os interesses do usuário, adaptar-se às mudanças e explorar o domínio de informações potencialmente relevantes.

O resultado inicial foi bastante promissor. O agente proposto demonstrou ser uma poderosa ferramenta de filtragem de informações. Entretanto, há muito trabalho a ser feito a fim de torná-lo um programa mais amigável e completo.

Resultados de testes simulados em ambientes controlados foram bastante satisfatórios. Eles mostraram que o uso da técnica de "relevance feedback" é bastante eficiente para o agente especializar-se nos interesses específicos de um usuário .

Nos testes foram obtidos valores altos de similaridade para documentos semelhantes e baixos para documentos distintos. A partir da obtenção do "feedback" do "usuário" para um determinado número de documentos fornecidos, foi gerado o arquivo de perfil, contendo a representação vetorial dos documentos que receberam "feedback" positivo. Ao se apresentar um novo documento ao "usuário" e receber seu "feedback", o perfil já criado foi ajustado, aumentando ou diminuindo os pesos de cada termo já existente, e incluindo os termos novos.

A seguir foram testados documentos não lidos pelo "usuário" e calculada sua similaridade em relação ao perfil. Verificou-se que os documentos selecionados pelo agente para apresentação ao "usuário", ou seja, os que obtiveram maior valor de similaridade correspondem efetivamente àqueles que potencialmente seriam de maior interesse. A seleção de documentos relevantes foi crescentemente melhorada ao longo das sessões de feedback do usuário.

Vale a pena ressaltar que sistemas cuja busca de informação é baseada apenas em palavra-chave apresentam uma deficiência. Alguns artigos são difíceis de serem expressados por "keyword", como por exemplo: artigos de humor, sátira, retórica, etc. O Agente Fenix não se baseia apenas em palavra-chave, ele recupera artigos através de um perfil formado para cada interesse do usuário.

O impacto de sistemas como Fenix em uma empresa, poderá ser altamente significativo. Muitas das tarefas de busca de informações poderiam ser automatizadas, pois o sistema é capaz de filtrar e priorizar diversos assuntos interessantes, reduzindo, assim o consumo de tempo gasto nessas atividades.

Agentes de Filtragem de Informação são uma grande promessa para o gerenciamento das inúmeras informações disponíveis .

7 REFERÊNCIAS BIBLIOGRÁFICAS

[Allan95] Allan, J. 1995. Relevance feedback with too much data. In Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

[Armstrong95] Armstrong, R.; Freitag, D.; Joachims, T.; and Mitchell, T. 1995. WebWatcher: A learning apprentice for the World Wide Web. In Proceedings of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Resources.

[Balabanovic97] Balabanovic , M. An Adaptive Web Page Recomendation Service. Stanford Universal Digital Libraries Project Working Papers SIDL - WP. 1997.

[Buckley95] Buckley, C.; Salton, G.; Allan, J; and Singhal, A. 1995. Automatic query expansion using SMART: TREC-3. In Proceedings of the 3rd Text Retrieval Conference.

[Caglayan97] Caglayan, Alper & Harrison, Collin. Agent Sourcebook - A Computer Guide to Desktop, Internet, and Intranet Agents. Wiley Computer Publishing - 1997.

[Fischer91] Fischer, G., Stevens, C., Information access in complex, poorly structured information spaces. Human Factors in Computing Systems CHI'91 Conference Proceedings, 1991, pp. 63-70.

[Foltz92] Foltz, P. W., and Dumais, S. T. 1992. Personalized Information Delivery: An Analysis of information filtering methods. Communications of ACM 35 (12):51-60.

[Guralmile70] Guralmile, D. (.ed) (1970). Webster's new World Dictionary of the American Language, Second College Edition. New York: The World Publishing Company.

[Harman95] Harman, D. 1995. Overview of the third Text Retrieval Conference (TREC-3). In Proceedings of the 3rd Text Retrieval Conference.

[Ketchpel94] Steven P. Ketchpel, Michael R. Genesereth,; Software Agents; Communications of the ACM Volume 37, Number 7, July 1994 (também disponível na WWW em http://logic.stanford.edu/sharing/papers/agents.ps)

[Lancaster69] Lancaster, F. W. 1969. "MEDLARS: Report on the Evaluation of Its Operating Efficiency." American Documentation, 20 (2) 119-48.

[Lang,95] Lang, K. 1995. NewsWeeder: Learning to filter netnews. In Proceedings of the 12th International Conference on Machine Learning.

[Lesk71] Lesk, M. E., and Salton, G. 1971. Relevance assessments and retrieval system evaluation. In The Smart System - Experiments in Automatic Document Document Processing. Prentice Hall Inc. 506-527.

[Nissen95] Nissen, M. et al.. Intelligent Agents: a Technology and Business Aplication Analysis, BA248D: Telecomunications and Distributed Processing, Intelligencia, Inc, November, 1995.

[Nwana96] Nwana, H. S.; Software Agents: an Overview; Knowledge Engineering Review Volume 11, Number 3, September 1996 (também disponível na WWW em http://www.cs.umbc.edu/agents/introduction/ao/)

[Rijsbergen] Rijsbergen, C. J. "Information Retrieval". The Information Retrieval Group. Disponível em: http://dsc.glasgow.ac.uk/preface.html

[Rocchio71 Rocchio, J.J. Relevance feedback in information retrieval. In: The Smart Retrieval System - Experiments in automatic Document Processing, p. 313-323, Englewood Cliffs: Prentice-Hall, 1971.

[Salton83] Salton, G., McGill, M. J., Introduction to Modern Information Retrieval, McGraw-Hill, 1983.

[Salton89] Salton, G., Automatic Text Processing. Addison-Wesley Publishing Company, Inc., Reading, MA, 1989.

[Saracevic95] Saracevic, T. 1995. Evaluation of evaluation in information retrieval. In Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

[Schutze95] Schutze, H.; Hull, D. A.; and Pedersen, J. O. 1995. A comparison of classifiers and documents representations for the routing problem. In Proceedings of the 18 th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

[Sheth] Sheth, Beerud. NEWT: A learning approach to personalized information filtering. Dissertação.[s.l.:199?]. Disponível em: http:// agents.www.media. mit.edu/groups/agents/papers/ newt-thesis/ tableofcontents2_1.html.

[Sheth93] Sheth, B., and Maes , P. 1993. Evolving agents for personalized information filtering In Proceedings of the 9 th IEEE Conference on Artificial Intelligence for Applications.

[Silva96] Silva, E. B. BSETI - Uma Ferramenta de Auxílio à Busca e Recuperação de Documentos - 1996. Disponível em: http://www.cos.ufrj.br/~bezerra/pf/PF. html

[Shoham 93] Shoham , Y.; Agent Oriented Programming; Artificial Intelligence 60; 1993

[Twidale95] Twidale, M. B., Nichols, D. M. and Paice, C. D.. Browsing is a Collaborative Process.Tecnical Report - CSEG/1/96-Computing Department, Lancaster University, 1996. Disponível em: http:// www.comp. lancs.ac.uk/computing /research/cseg/projects/ariadne/docs/bcp.html.

[Rijsbergen86] van Rijsbergen, C. J. 1986. "A New Theoretical Framework For Information Retrieval". Paper apresentado na ACM Conference on Research and Development in Information Retrieval, Pisa, Italy.

[Yao95] Yao, Y. Y. 1995. Measuring retrieval effectiveness based on user preference of documents. Journal of the American Society for Information Science 46(2):133-145.