DESENVOLVIMENTO DE UM ORGANIZADOR AUTOMÁTICO DE MENSAGENS PARA O AMBIENTE DO GMAIL

Igo Amaurí Luz, Larissa Soares, Marcelo Bastos, Charlene Almeida, Angelo Loula

Colegiado de Engenharia da Computação – Universidade Estadual de Feira de Santana
Br 116, KM 03, Avenida Universitária – 41031-460 – Feira de Santana – Ba – Brasil

igoamauri@gmail.com, lari.rsoares@gmail.com,
marcelomirbas@yahoo.com.br, charlene.uefs@gmail.com,
angelocl@ecomp.uefs.br

Abstract*. Electronic mail messages are a major form of communication today, but the growing volume of messages has required tools to organize them. We present a solution for automatic classification of labeled messages, integrated into the web application of one of the most popular e-mail services, Gmail.*

Resumo. *Mensagens de correio eletrônico são uma das principais formas de comunicação atualmente, mas o volume crescente de mensagens tem demandado ferramentas para sua organização. Apresentamos uma solução para classificação automática de mensagens rotuladas integrado ao ambiente web de um dos serviços mais populares de correio eletrônico, o Gmail.*

1. Introdução

O serviço de mensagens de correio eletrônico, atualmente, é uma das principais formas de comunicação entre as pessoas. Isto se deve a rapidez e facilidade na transferência das mensagens, e também ao baixo custo envolvido, gerando assim um aumento no número de usuários do serviço.. O volume crescente de mensagens que cada indivíduo recebe, no entanto, tem demandado cada vez mais tempo para as pessoas lerem e processarem mensagens recebidas.

Existe, portanto, uma demanda pela organização dessas mensagens de forma a separá-las e diferenciá-las, tanto para leitura quanto para armazenamento. Tradicionalmente, essa organização passa pela classificação das mensagens através de pastas ou marcadores, realizada de forma manual pelo usuário ou através de filtros especificados pelo mesmo. O processo de leitura e armazenamento das mensagens passa, então, a ser orientado pelas pastas/marcadores e o processo de recuperação de mensagens já lidas também pode ser feito da mesma maneira.

O objetivo deste trabalho está relacionado ao problema de gestão de mensagens eletrônicas. Particularmente, será apresentada uma solução para auxiliar na organização de mensagens através da inferência automática de critérios de classificação de mensagens por marcadores. Esta solução foi integrada a interface de um dos serviços de e-mail mais populares na atualidade, o Gmail[1]. Existem outros trabalhos que foram desenvolvidos com o intuito de realizar a classificação de e-mails, para isso, esses trabalhos discutem a utilização de diferentes técnicas e baseiam-se sempre na construção de soluções separadas da interface web dos servidores de e-mail [por

[1] Gmail é uma marca registrada de Google Inc.

exemplo, Crawford et al. 2001; Clark et al. 2003]. Sendo assim, este é o primeiro trabalho a buscar criar um serviço de classificação associado a uma interface web de um serviço popular de correio eletrônico.

Este artigo está organizado da seguinte forma: na Seção 2 há uma breve explicação sobre serviço de e-mail e o Gmail; a Seção 3 aborda a técnica de mineração de dados utilizada; a Seção 4 apresenta a metodologia do trabalho, a arquitetura proposta, o desenvolvimento da interface, do script e da *applet*; a seção 5 apresenta o produto final e o seu funcionamento; e, por fim, a Seção 6 aborda as conclusões do trabalho apresentado.

2. O Serviço de E-mail e o GMail

O serviço de e-mail é uma forma de comunicação entre pessoas conectadas a Internet. Este serviço possui várias características que contribuem para sua popularização. A comunicação é assíncrona, não necessita que remetente e destinatário estejam conectados simultaneamente. É rápido, as mensagens podem ser entregues até em menos de um segundo, mesmo que os usuários estejam em continentes diferentes. Permite flexibilidade quanto ao conteúdo, sendo possível transferir textos, vídeos, imagens ou qualquer tipo de arquivo digital. Seu custo é muito baixo e está relacionado em geral ao acesso a Internet pelo usuário, que pode utilizar serviços gratuitos de e-mail.

Atualmente existem muitos serviços de e-mails com interface web, como Hotmail, Yahoo e Gmail. Dentre esses, destaca-se o Gmail (serviço de email do Google), criado em março de 2004. O Gmail, segundo a pesquisa da Complete em 2010 nos Estados Unidos, possui cerca de 15% dos usuários e já está consagrado como o melhor serviço de e-mail [Complete 2010].

O Gmail apresenta como um de seus principais diferenciais o agrupamento de mensagens relacionadas em conversas e também o uso de marcadores ao invés de pastas para organizar as mensagens, permitindo que uma mesma mensagem possa ter vários marcadores. A aplicação de marcadores a mensagens pode ser feita manualmente pelo usuário, que seleciona as mensagens e então aplica determinado marcador, ou então podem ser usados filtros. Um filtro define um critério – especificado pelo usuário – relacionado ao conteúdo de mensagens novas, e ações a serem executadas sobre mensagens que atendam a este critério. Este recurso de filtros auxilia na organização de mensagens permitindo, por exemplo, que seja criado um filtro para aplicar um determinado marcador a mensagens que atendam ao critério definido, categorizando, assim, toda mensagem nova. Há ainda um recurso que tenta auxiliar o usuário na definição dos filtros, o '*Filter messages like these*' que permite inferir um filtro para mensagens pré-selecionadas. Este recurso, no entanto é limitado, pois cria somente critérios de filtragem baseados no remetente da mensagem.

A proposta desse trabalho é desenvolver um novo recurso para inferência automática de critérios de filtragem baseado em marcadores usando informações diversas do conteúdo da mensagem (e não somente sobre o remetente). Para isso, a solução envolve um módulo para interação com o usuário, incluindo elementos na interface gráfica do Gmail, baseada em HTML e em JavaScript , assim como um módulo para comunicação com o serviço de email e para inferência de um critério de filtragem, aplicando um árvore de decisão C4.5.

3. Classificação por Árvore de Decisão C4.5

Classificação é um método de mineração de dados cujo objetivo é classificar elementos de um conjunto de dados em diferentes classes, baseado em propriedades que estes elementos têm em comum (atributos) [Souza 1998].

Um classificador tem como base um algoritmo de aprendizagem e este tem como entrada um conjunto de exemplos, descritos por valores de atributos. Este algoritmo tem como saída um esquema de classificação que irá predizer classes baseadas nos valores de um conjunto de atributos de entrada.

Uma árvore de decisão é um tipo classificador descrito por uma estrutura de árvore. Nesta árvore, as folhas representam classificações para os dados de entrada, os nós internos representam atributos e os ramos são valores possíveis de atributos. Para classificar uma entrada, descrita como um conjunto de atributos, deverá ser feito o percurso da árvore seguindo os valores dos atributos da entrada até encontrar um nó folha com uma classe a ser atribuída à entrada.

Para exemplificar uma árvore de decisão, pode-se considerar um exemplo de classificação de mensagens eletrônicas para determinar se trata-se ou não de mensagens relacionadas a família. Isto poderia ser feito avaliando o atributo de remetente das mensagens em uma árvore de decisão como a representada na figura 1.

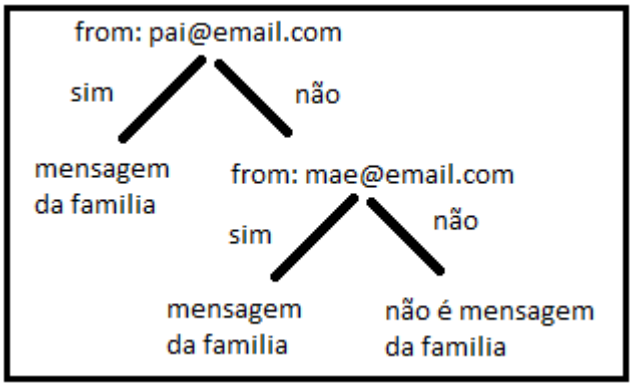

Figura 1. Exemplo de árvore de decisão

Um algoritmo de aprendizagem para construção de árvores de decisão, a partir de exemplos, amplamente utilizado para mineração de dados é o C4.5 [Quinlan 1993]. Este algoritmo foi utilizado neste trabalho para a inferência dos filtros de e-mails. O algoritmo C4.5 constrói uma árvore de decisão utilizando o ganho de informação como critério de escolha dos atributos para os nós internos. O ganho de informação para um atributo indica qual atributo irá separar um maior número de exemplos de classes diferentes em ramos diferentes (para mais detalhes, ver Quinlan 1993).

Nesse trabalho, é utilizado o algoritmo C4.5 para atributos binários (a entrada possui ou não determinada característica para uma mensagem eletrônica) e classificação binária (uma entrada deve ser atribuída ou não a um determinado marcador).

4. Metodologia

Este projeto teve como objetivo a criação de uma solução para classificação automática de mensagens de correio eletrônico integrada a interface web do Gmail. Para isso, era preciso alterar, no navegador do usuário, a interface enviada pelos servidores do Google. Assim foi utilizado o navegador Mozila Firefox, em conjunto com a ferramenta de execução de scripts personalizados GreaseMonkey.

A página do serviço do GMail assim como a ferramenta do Greasemonley utilizam scripts escritos em JavaScript. O JavaScript é uma linguagem de script integrável ao HTML que é executada do lado do cliente [Flanagan 2002]. Dessa forma, promove a capacidade de processamento independente do servidor, deixando-o a cargo do navegador que usa os recursos da máquina do cliente. O JavaScript possui bibliotecas que facilitam a construção dos scripts, dentre elas pode-se destacar a JQuery, desenvolvida para percorrer de forma rápida e concisa documentos HTML [jQuery 2010].

O GreaseMonkey é uma ferramenta que possibilita a execução do scripts que podem alterar localmente o conteúdo de páginas web vindas de servidores remotos. Ele é utilizado como um complemento disponível para o Mozila Firefox que faz a execução de scripts em sites da web. Ele permite que o usuário instale ou crie seus próprios scripts, de modo a modificar o visual ou as funcionalidades das páginas que recebe.

As tecnologias de JavaScript, Greasemonkey e JQuery, assim como o complemento Firebug [Firebug 2010] para depuração de páginas e scripts no Firefox, possibilitaram a execução desse projeto e a implementação da arquitetura proposta.

4.1 Arquitetura proposta

A primeira arquitetura proposta para nossa solução integrada à interface do Gmail está ilustrada na figura 2. Nossa solução personalizaria a interface do Gmail, para que pudéssemos interagir com o usuário, através do GreaseMonkey usando também a API para o Gmail disponibilizada para o Greasemonkey. As mensagens para construção do classificador poderiam ser obtidas também através desta API, que as solicitaria aos servidores do Google. Mas, após pesquisas e testes iniciais, verificou-se a impossibilidade de acessar os e-mails através desta API devido à inexistência de funções que fizessem a comunicação direta com o servidor do Gmail e também devido a API não estar funcional devido a mudanças no serviço do Gmail.

Diante disso, foi elaborada uma nova proposta alternativa para contornar estas dificuldades. A alteração consistiu na utilização de uma *applet* em Java, que fosse incorporada a página do Gmail, sendo responsável pela comunicação direta com os servidores remotos Gmail. A figura 3 ilustra essa alternativa na forma de acesso.

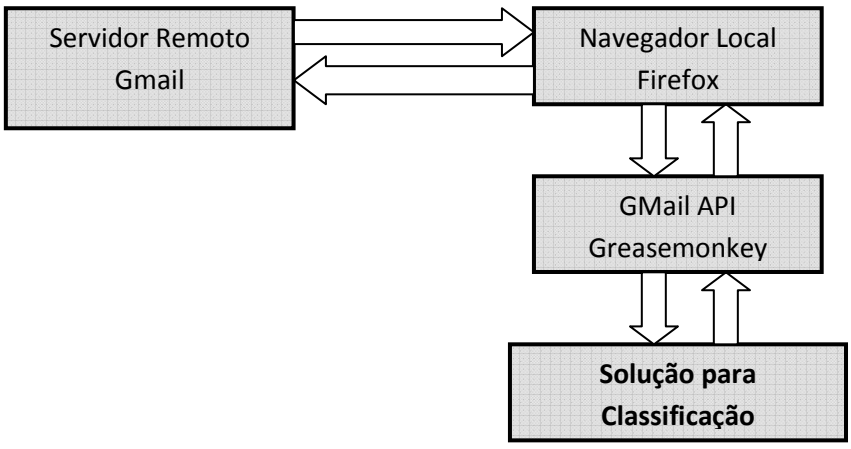

Figura 2. Arquitetura inicial do projeto

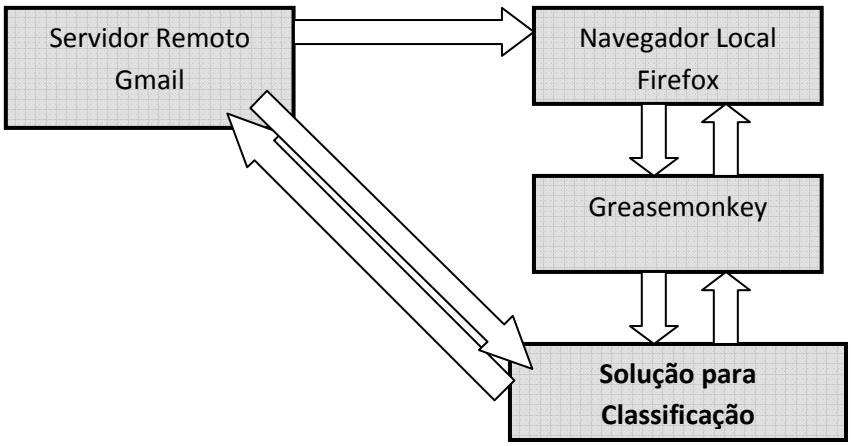

Figura 3. Arquitetura alternativa do projeto

4.2 Desenvolvimento da Interface

Para criar elementos na interface gráfica do Gmail que permitisse ao usuário interagir com nossa solução, buscamos integrar estes novos elementos a elementos já existentes na página deste serviço de email, criando um script executado pelo Greasemonkey. O primeiro novo elemento foi uma nova opção no menu de cada marcador, opção esta pela qual o usuário pode solicitar a criação automática de um filtro para mensagens com o respectivo marcador. O segundo novo elemento é um painel removível situado sobre a janela principal do Gmail, utilizado para solicitar informações e interagir com o usuário.

Para o desenvolvimento da interface, foi necessário conhecer a estrutura HTML da página do Gmail – bem como as suas propriedades -, para que fosse possível inserir elementos nas posições desejadas e definir as características visuais destes. Para isso, a ferramentas Firebug foi aplicada para inspecionar o conteúdo da página.

4.3 Desenvolvimento do Script

O script pode ser dividido em duas partes, a exibição de elementos na interface do Gmail e a comunicação da interface com a *applet*. Quando o usuário fornece a sua senha no painel para conexão com o servidor do Gmail, a *applet* é criada sob demanda, já que a utilização da função de sugerir filtro não é utilizada sempre pelo usuário. Para criar a *applet*, é inserido um código HTML na página que instancia a *applet* no navegador.

Para a comunicação da *applet* com o script foi necessário a criação de funções em JavaScript para que ela utilizasse. Essas funções têm o objetivo de informar o resultado da operação de *login*, mostrar mensagens de processamento ou de erro, e avisar sobre a criação de um critério de filtragem ao final do processamento.

Para permitir ao usuário utilizar o critério de filtragem descoberto, foi necessário também a simulação pelo script de um evento para ativar a opção de criar filtro nativa do Gmail e depois preencher seus campos automaticamente.

Para facilitar a programação em JavaScript foi utilizada a biblioteca JQuery, que abstrai a busca, adição e remoção de elementos HTML e temporização com chamadas de *callback*. Assim torna-se possível localizar as posições da página que iríamos personalizar, assim como vincular chamadas a nosso script.

4.4 Desenvolvimento da *Applet*

A *applet* desenvolvida tem o papel de realizar a comunicação com o Gmail. Para isso utilizou-se a API JavaMail, do Java [Oracle 2011], e , como protocolo de comunicação, para obter as mensagens, utilizou-se o Internet Message Access Protocol (IMAP) [Tanenbaum 2001].

Nesta comunicação, os e-mails são buscados no servidor e processados de forma a construir o filtro. São buscadas mensagens rotuladas com o marcador escolhido (exemplos positivos), e também mensagens da caixa de entrada que não possuem este marcador (exemplos negativos). Para que estas mensagens sejam processadas, faz-se necessário, inicialmente, realizar um pré-processamento que tem como objetivo extrair as características contidas no cabeçalho de cada e-mail buscado.

As características do cabeçalho do e-mail que são utilizadas para o nosso filtro são os campos “to”, “from”, “list-id”, “cc” e “subject”. Todos estes campos são analisados e separados de forma a transformar cada mensagem em um vetor de atributos que possa ser utilizado pelo algoritmo de construção de árvores de decisão. Os campos “to”, “from”, “list-id”, “cc” são divididos por endereço contidos em cada um deles. Assim, se uma mensagem contém “to: uefs@uefs.br, uesc@uesc.br”, este campo transforma-se em “to:uefs@uefs.br” e “to:uesc@uesc.br”.

Para “subject”, este processamento é feito por palavra e bi-gramas de palavras em sequência, sendo retiradas todas as conjunções, preposições, artigos e espaços. Por exemplo, para o subject “Curso de computação da UEFS realiza debates” são criadas 9 características: “subject:Curso”, “subject:computação”, “subject:UEFS”, “subject:realiza”, “subject:debates”, “subject:Curso computação”, “subject: computação UEFS”, “subject: UEFS realiza” e “subject: realiza debates”.

5. Produto Final

O produto final desse projeto consistiu na criação de uma solução para classificação automática de e-mails integrada ao Gmail. Com esse script, que é executado na página do serviço de e-mail do Google, o usuário seleciona a opção de criação de um classificador para um marcador (*label*), e então mensagens com e sem este marcador são processadas, apresentando uma sugestão de filtro para o marcador em questão.

Em relação ao funcionamento do script, inicialmente ocorre o carregamento do mesmo na página do Gmail, gerando assim a criação do botão “Automatic Filter”. Após essa fase, o usuário pode informar sua senha, para que a *applet* possa se comunicar com o servidor do Gmail e ter acesso aos e-mails do usuário. Esta senha é verificada, ficando o script à espera que o usuário selecione um marcador. Quando isso é feito, o script executa a *applet*, que cria uma *thread*, que irá buscar mensagens do usuário diretamente do servidor remoto. Após esse processo, a *applet* processa o cabeçalho dos e-mails em busca dos atributos que poderão compor a sugestão de filtro, através do algoritmo da árvore de decisão C4.5. Finalizado o processamento das mensagens, a *applet* gera uma sugestão de filtro que é então repassado para apresentação na tela e disponibilização para o usuário. A Figura 4 apresenta a tela inicial do Gmail, já personalizada por nosso script.

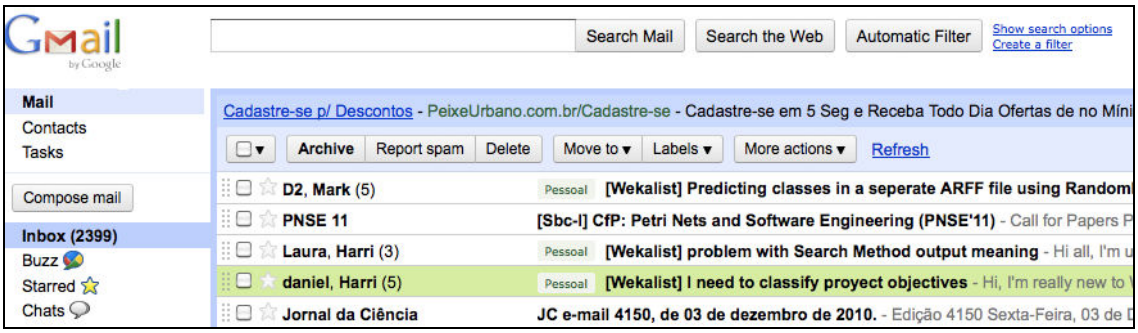

Figura 4. Tela inicial do Gmail com script instalado

Na figura 4, pode-se visualizar a adição do botão "Automatic Filter" referente ao script, o mesmo localiza-se ao lado do botão "Search the Web". Esse novo elemento adicionado permite a visualização do novo painel para nossa solução. Ao clicar no mesmo, pela primeira vez, será solicitada senha do usuário. Nesse momento, o usuário efetua o *login* e o script retornará se o mesmo foi feito corretamente ou ocorreram erros.

Depois de feito o *login* corretamente, o script fica a espera que seja escolhido um marcador para que seja feita a sugestão de filtro. Para isso, o usuário deve ir até as opções de um determinado marcador e selecionar a opção "Suggest Filter". A Figura 5 ilustra um exemplo mostrando esta nova opção.

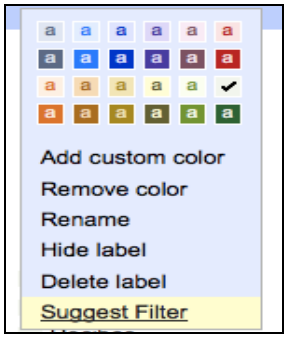

Figura 5. Opção do *Suggest Filter*

Depois de selecionado o marcador, inicia-se o processamento das mensagens. Durante esta etapa, o usuário poderá ver mensagens de progresso deste processamento, por exemplo, depois obter-se as mensagens do servidor de e-mail do Gmail e iniciar o processamento das mesmas utilizando a mineração de dados, aparecerá a mensagem "Processing Messages...". A Figura 6 ilustra esse momento.

Figura 6. Tela processamento de mensagens

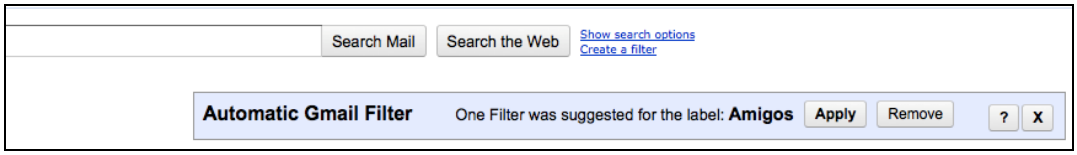

Figura 7. Tela de aviso de sugestão de filtro

Finalizado o processamento das mensagens, o aplicativo então avisa ao usuário que existe uma sugestão de filtro a ser apresentada. A figura 7 ilustra esta sugestão.

Para que a *applet* construa a sugestão do filtro "Amigos" da figura 7, inicialmente, ela é carregada, por exemplo, com os e-mails da figura 9 e com os exemplos negativos. Exemplos negativos são os outros e-mails que não estão classificados com o marcador "Amigos". Após o processamento dessas mensagens o algoritmo da Árvore C4.5, presente na *applet*, gera uma árvore como ilustrado na figura 8.

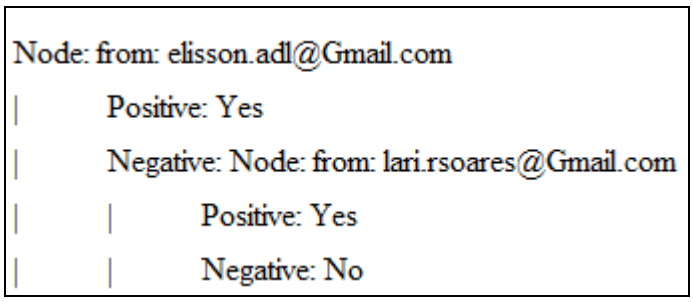

Figura 8. Árvore exemplo gerada

Assim, percorrendo a árvore, o filtro que é devolvido para o Gmail corresponde a: "(from: elisson.adl@Gmail.com OR from: lari.rsoares@Gmail.com)".

Figura 9. Exemplo de e-mail

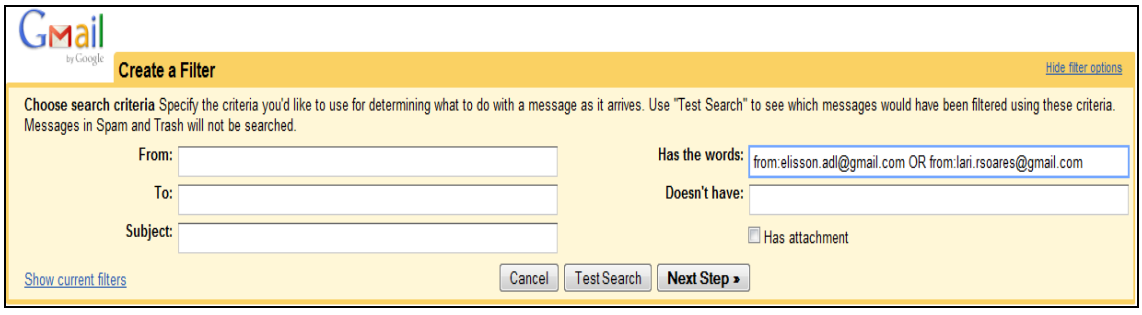

Figura 10. Filtro gerado

No momento em que o usuário seleciona a opção "Apply", a janela de definição de novos filtros do próprio Gmail é aberta. O usuário poderá visualizar o novo filtro sugerido e então utilizá-lo ou não. A figura 10 ilustra a tela do filtro sugerido e pode-se

observar que a partir de uma quantidade de e-mails de determinado marcador foi gerado um filtro coerente com as características dos e-mails

6. Conclusão

Esse artigo apresentou o desenvolvimento de uma ferramenta para classificação automática de mensagens de correio eletrônico integrada à interface do web do serviço de e-mail do Google, o Gmail, que provê funcionalidades para auxiliar na organização de mensagens.

O uso de um classificador automático permite que o usuário não necessite identificar manualmente critérios de filtragem adequados para classificar suas mensagens. Ao invés disso, os filtros para classificação podem ser induzidos por algoritmos com base nas mensagens existentes em pastas/marcadors do próprio Gmail. A ferramenta desenvolvida objetiva ainda oferecer mais facilidade na busca e recuperação de mensagens, seja pela classificação já realizada, ou através do auxílio à busca por agrupamento de mensagens.

7. Referência

Clark, J., Koprinska, I., Poon, J. (2003) "A neural network based approach to automated e-mail classification". In: Proceedings IEEE/WIC International Conference on Web Intelligence, 2003. 13-17 Oct. 2003. p. 702 – 705.

Complete Pulse. (2010) "Gmail's buzz - much bigger then its bite?", http://blog.compete.com/2010/11/11/Gmails-buzz-much-bigger-than-its-bite/, Dezembro.

Crawford, E., Kay, J., McCreath, E. (2001) "Automatic induction of rules for email classification". In: Proceedings of the 6th Australasian document computing symposium, Coffs Harbour, Austrália. p. 13-20.

Flanagan, David. (2002) JavaScript - O Guia Definitivo, O'Reilly & Associates, Inc. Bookman.

Firebug. (2010) "Firebug" , http://getfirebug.com/, Dezembro.

Greasemonkey. (2010) "Greasemonkey-GreaseSpot" , http://wiki.greasespot.net/Greasemonkey, Dezembro.

Hegaret P.; Wood L.; Robie J. (2010) "What is the Document Object Model?", http://www.w3.org/TR/DOM-Level-2-Core/introduction.html, Dezembro.

jQuery Project. (2010) "jQuery JavaScript Library", http://docs.jquery.com/, Dezembro.

Oracle. (2010) "JavaMail API", http://www.oracle.com/technetwork/java /javamail/index.html, Dezembro.

Quinlan, J. Ross. (1993) "C4.5: Programs for Machine Learning". Morgan Kaufmann.

Souza, M.; Mattoso, M.; Ebecken, N. (1998) Data Mining: a database perspective. In: EBECKEN, N. Data Mining.Boston:WIT Press.

Tanenbaum, Andrew S. (2001) Computer Networks, Editora Campus, 4a edição.