Sistemas de recomendação de notícias na Internet baseados em filtragem colaborativa

Allan Panossian Kajimoto - 5122986
Renato Shirakashi de Sousa - 5123027
Sidney Eduardo Serra Zanetti - 5122781
Victor Miranda Cirone - 5123006

Universidade de São Paulo
Instituto de Matemática e Estatística
MAC499 – Trabalho de Formatura Supervisionado
Orientador: João Eduardo Ferreira

Fevereiro 2008

Sumário

1. Introdução

A quantidade de informações hoje disponíveis nos diversos meios de comunicação é enorme. O problema da falta de acesso à informação foi substituído pela necessidade de filtrá-la e apresentá-la de acordo com as necessidades de visualização de conteúdo dos interessados, que, por muitas vezes, se perdem em meio a tanta informação pessoalmente irrelevante. Diante desse contexto, encontrar formas de filtrar conteúdo, oferecendo relevância ao usuário, tornou-se um imenso diferencial competitivo entre as organizações.

A Internet, grande responsável por essa inversão, tem sido também o principal objeto de estudo para encontrar alternativas para classificar e filtrar conteúdo.

Entretanto, a recente implementação de sistemas colaborativos através da Web 2.0 em serviços como Digg [10] – serviço em que os usuários avaliam notícias, vídeos e imagens inseridos pelos próprios usuários –, Myspace [11] – comunidade virtual que permite pessoas encontrarem seus amigos, bandas e artistas –, Via6 [12] – comunidade virtual profissional que tem como objetivo o compartilhamento de conhecimento pela Internet –, Twitter [13] – serviço em que os usuários comunicam-se e mantêm-se conectados por meio da troca de mensagens rápidas – e Wikipédia [14] – enciclopédia livre em que seus verbetes e artigos são criados e alterados pelos próprios usuários – começa a criar uma enorme base de dados de descrições, intenções e classificações de conteúdo, levantando a questão de como utilizar a participação dos usuários sobre o conteúdo para recomendação.

Por exemplo, um usuário pode ser apresentado a um conteúdo não apenas baseando-se em sua idade, sexo ou características pessoais, mas utilizando dados fornecidos por outros usuários para definir o melhor conteúdo a ser oferecido. Dessa maneira, podemos, devido à crescente participação dos usuários sobre o conteúdo, utilizar-se dessas informações para encontrar métodos para filtrar conteúdo relevante.

O propósito desse estudo é, portanto, estudar métodos para recomendar conteúdo através de classificação de usuários tendo em vista uma abordagem prática e adaptativa. O resultado desse estudo é a ferramenta colaborativa Rec6[1], na qual a ação de usuários, combinada a algoritmos descritos nos próximos itens desse estudo, provém notícias de forma relevante. Embora tal ferramenta também inclua outras funcionalidades, esse estudo restringe-se apenas à descrição do compartilhamento e recomendação de notícias como forma de classificação.

O objetivo desse estudo também não é introduzir o leitor à área de classificação, recomendação e recuperação da informação. Tais estudos, principalmente na área de recuperação da informação, podem ser encontrados em diversas outras publicações (BAEZA-YATES, Ricardo; RIBEIRO-NETO, Berthier. *Modern Information Retrieval*. Nova Iorque: Addison Wesley, 1999. SALTON, Gerard; MCGILL, Michael. *Introduction to Modern Information Retrieval*. Nova Iorque: McGraw-Hill Inc., 1986). Esse estudo restringe-se apenas a apresentar métodos específicos de ordenar conteúdo a partir de recomendação de usuários (filtragem colaborativa), detectar problemas e propor soluções.

---

[1] http://www.rec6.com.br

1.1 A Web 2.0

Tim O'Reilly [8] [9] em 2005 denominou uma série de fatores que começavam a mudar o paradigma da interação com a internet como Web 2.0. Esses fatores, principalmente ligados à maior participação do usuário na internet, propiciaram a criação de ambientes colaborativos, nos quais os próprios usuários geravam, compartilhavam e consumiam o conteúdo na rede. De certa maneira, o antigo modelo baseado em um produtor de conteúdo institucional estava sendo substituído por esse novo modelo, que têm se mostrado mais relevante para os usuários, já que não estão ligados diretamente aos interesses de um pequeno grupo de pessoas ou instituições.

Entre os fatores que descrevem a Web 2.0, descritos por O'Reilly, estão:

1. **Conteúdo gerado pelo usuário**

   Cada vez mais os usuários podem gerar conteúdo e disponibilizá-lo na Web, por meio de vídeos, blogs e comunidades virtuais. Esse conteúdo pode, inclusive, sofrer interações de outros usuários, na forma de comentários, edição, citações, cópias ou paródias. Este processo de interação entre usuários e conteúdo gerado por eles tem criado um imenso repositório de informações relevantes para os usuários e esse conteúdo tem ganhado a preferência de muitos usuários, como se pode observar na ascensão de serviços como YouTube, Myspace, Orkut e Wikipedia.

2. **Web como plataforma**

   A Web tem se tornado mais do que uma simples rede de documentos interligados, mas uma plataforma capaz de prover serviços e conectividade. Por esse motivo, a criação de aplicativos e infra-estrutura necessária para executá-los online tem tornado a Web uma plataforma para aplicações que explorem conectividade e compartilhamento, provendo serviços que não poderiam ser oferecidos de forma off-line. A portabilidade de serem executados a partir de um navegador, dividindo processamento com um servidor remoto, garante grande portabilidade as aplicações e acesso a praticamente todos os usuários.

3. **Informação sobre tecnologia**

   A informação tem se tornado o principal ativo de organizações, deixando aspectos que antes eram primários, como tecnologia, a um patamar secundário. A enorme quantidade de informações sendo criadas, distribuídas e compartilhadas é a causadora desse processo, tornando otimizações criadas pela tecnologia terem um menor impacto diante da influência da criação de idéias e do conhecimento.

Essencialmente Web 2.0 pode ser descrita como colaboração. Muitas aplicações, como o Wikipédia, Digg, e redes sociais são frutos desse novo paradigma. Com ele, a criação de conteúdo ganhou proporções globais, aumentando a quantidade de informações disponíveis, e também informações qualitativas e quantitativas sobre as mesmas. Com essa quantidade enorme de informações e classificações sobre elas, por

meio de sistemas de votação, folksonomia[2] e taxonomia, encontrar sistemas que possam utilizar essas informações para recomendar conteúdo para o usuário é um enorme diferencial.

2. Sistema de recomendação

2.1 Introdução

Como resultado da enorme quantidade de informações geradas e disponibilizadas na Internet, por meio de ferramentas colaborativas ou produção da mídia tradicional, um dos maiores problemas de veículos de comunicação deixou de ser deter a informação, mas apresentá-la de forma relevante ao usuário final. Sistemas de recomendação tentam determinar qual conteúdo deve ser apresentado, de forma mais relevante.

Recomendar não é uma tarefa fácil. Normalmente, em nosso dia-a-dia, utilizamos recomendações para diversas tarefas, como realizar uma compra, agendar uma visita ao médico ou mesmo buscar uma referência para alguma informação. Segundo Resnick e Varian [2], sistemas de recomendação auxiliam no aumento da capacidade e eficácia deste processo de indicação já bastante conhecida na relação social entre seres humanos.

As características encontradas nesse tipo de sistema são, basicamente, filtragens baseadas em perfis de usuários ou grupos; manipulação de uma grande quantidade de dados em diversos tipos de mídias (texto, vídeo, imagem, áudio, etc.); e manipulação de informação não estruturada, ao contrário de sistemas de banco de dados que trabalham com informações fortemente estruturadas.

Sistemas de recomendação já têm sido largamente utilizados, principalmente em sites de compra, como Amazon[3] e Submarino[4], na intenção de sugerir produtos que possam ser de interesse do usuário. Também são utilizados em espaços publicitários, para melhor adequar o teor do anúncio ao seu alvo. Por exemplo, uma mulher pode ser apresentada a um anúncio de uma loja de cosméticos a partir de uma escolha de um sistema de recomendação, que assim julgou relevante tal anúncio; ou de forma invertida, na qual o anunciante define qual o público alvo de sua campanha.

Dois principais tipos de entidade podem ser identificados em sistemas de recomendação: usuários e conteúdo. A relação entre ambos é responsável pela recomendação. O problema típico de sistema de recomendação pode ser descrito da seguinte maneira:

Para um usuário u, um conjunto C de conteúdo, $c_i \in C$, recomendamos c tal que $F(u, c) = MAX\ F(u, c_i)$, no qual

F: usuário x conteúdo -> pontuações

---

[2] A folksonomia permite a cada usuário da informação a classificar com uma ou mais palavras-chaves, conhecidas como *tags* (em português, marcadores). [7]

[3] http://www.amazon.com

[4] http://www.submarino.com.br

é uma função que determina a relevância de c a u. Dessa forma, se conhecermos os valores de pontuações para todo usuário x conteúdo pertencente ao domínio, podemos determinar um conteúdo c_k que maximiza F para dado usuário u, portanto é o melhor item a ser recomendado a ele.

Os sistemas de recomendação podem ter diversas abordagens para determinar a função F. Muitos deles utilizam informações sobre o item, traçando e determinando se o perfil do usuário se relaciona com ele. Através de palavras-chave, por exemplo, é possível determinar, através de uma matriz, quais palavras-chave de um item também estão presentes em um perfil de um determinado usuário.

Outra abordagem relaciona itens de interesse do usuário, captados através de alguma entrada prévia e relaciona-os com itens similares e conclui que poderão ser de interesse do usuário em maior ou menor nível. Um bom exemplo desse tipo de abordagem pode ser encontrado comercialmente em lojas de comércio eletrônico, como Amazon e Submarino, onde a compra de um produto fornece dados para que produtos similares sejam ofertados futuramente ao usuário.

Essas abordagens descritas acima são conhecidas como filtragens baseadas em conteúdo, ou cognitivas. Outro tipo de filtragem para sistemas de recomendação é a filtragem colaborativa. Nesse tipo de filtragem, relaciona-se não conteúdo a usuários, mas usuários entre si e determinam-se usuários com perfis similares para então determinar conteúdo a ser recomendado.

2.2 Tipos de filtragem em sistemas de recomendação

Malone [18] identificou três tipos de filtragem de informação, duas delas diretamente relacionadas a sistemas de recomendação, a filtragem cognitiva e filtragem colaborativa.

2.2.1 Filtragem Colaborativa

Nesse tipo de filtragem, as recomendações são feitas por meio da previsão das preferências do usuário baseadas em interações de outros usuários, não havendo a necessidade de conhecimento do conteúdo dos itens. Em geral, esse tipo de filtragem oferece um maior grau de surpresa ao usuário com boas recomendações e, em alguns casos, pode oferecer conteúdo de forma totalmente irrelevante. Esse tipo de filtragem foi proposto para suprir os pontos que a filtragem cognitiva não conseguia atender.

Resnick [2] apresenta uma primeira abordagem para filtragem colaborativa na qual determina recomendações baseadas em conteúdo consumido por usuários com mesmo padrão de consumo do usuário atual, conforme ilustrado na figura 1. Essa abordagem é utilizada principalmente em sistemas de comércio eletrônico, como Amazon e Submarino. A figura 3 representa uma tela onde é possível ver esse tipo de implementação.

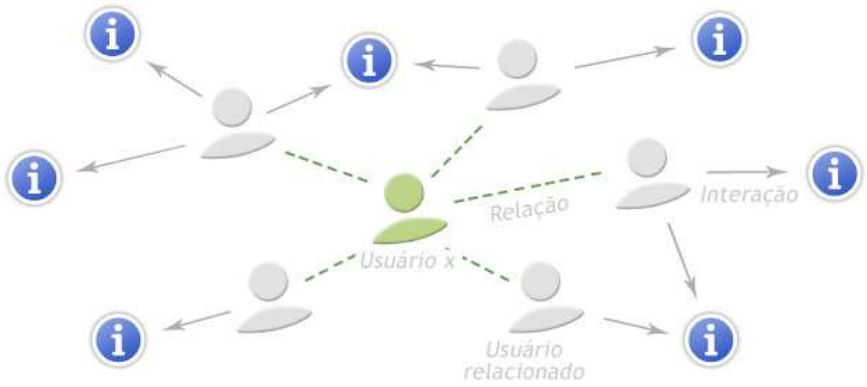


Figura 1: Filtragem colaborativa

Para facilitar o entendimento desse tipo de abordagem, considere um usuário x e xi os i usuários com padrão de compra mais parecido com x (compraram alguns dos mesmos produtos que x comprou). Considere agora os vetores (p, n), nos quais p é um produto e n é o número de vezes em que esse produto foi adquirido por algum xi. Ordenando-se o conjunto dos vetores (p, n) decrescentemente em n, temos uma ordem de recomendação de produtos para x conforme apresentado na tabela 1. Uma variação pode trabalhar com pesos entre usuários xi, baseados em seu nível de relação com x.

Usuário (xi)	**Produto**
1	Livro1
2	CD1
3	CD2
4	Livro1
5	DVD1
6	Livro1
7	CD2

Tabela 1: Filtragem colaborativa

V = {(Livro1, 3), (CD2, 2), (CD1, 1), (DVD1, 1)}

Nessa abordagem, uma questão importante é a coleta de informações para a definição do perfil de um usuário. Existem dois métodos básicos para fazer esse tipo de coleta: o primeiro é através de informações diretas que o próprio usuário fornece ao sistema, e o segundo é através do aprendizado do sistema pelas ações (comportamento) do usuário.

Uma segunda abordagem, utilizada nesse estudo conforme ilustrada pela figura 2, determina recomendações baseadas nas classificações realizadas por outros usuários dentro de um grupo restrito de conteúdo, ordenadas pela soma da relevância de tais classificações como um sistema de votação com pesos. Essa abordagem é utilizada principalmente em sistemas de notícias, como o Digg. Em geral, esse tipo de abordagem oferece recomendações menos pessoais e mais dirigidas a um grupo determinado de usuários, restritos a um tema. Em contrapartida, sua implementação enfrenta menos problemas de escalabilidade e em geral é mais viável.

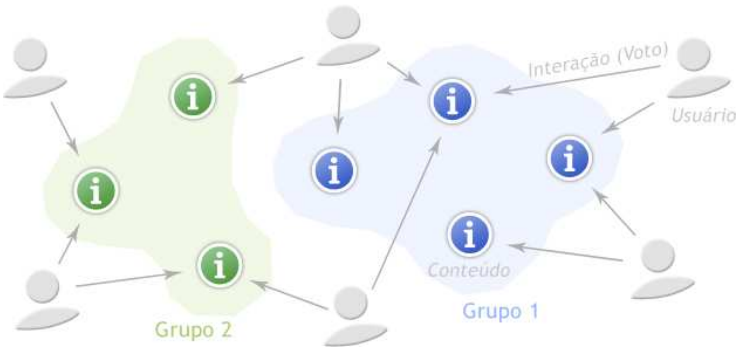


Figura 2: Filtragem colaborativa utilizada

Figura 3: Exemplo de filtragem colaborativa utilizada

Na Filtragem Colaborativa o processo de identificação de um item relevante a um usuário é chamado de "*k-nearest-neighbor*" [6] e segue os seguintes critérios:

- Identificação dos usuários similares ao usuário alvo e seus respectivos níveis de similaridade;

- Ordenação dos níveis de similaridade para determinação de um novo grupo de usuários que possuem os maiores níveis;

- Verificação das avaliações dadas pelos usuários do novo grupo e definição dos itens mais relevantes a serem recomendados.

O exemplo seguinte ajuda a explicar melhor o processo descrito acima.

Sejam três usuários (Marcos, Pedro e Thiago) e cinco itens (Livro1, Livro2, CD1, CD2 e DVD1). Seja a tabela abaixo a relação das opiniões desses usuários com os itens descritos (as opiniões variam de 1 a 10).

	Livro1	Livro2	CD1	CD2	DVD1
Marcos	2	9	4	10	-
Pedro	7	1	6	2	1
Thiago	1	8	5	9	8

Nota-se que Marcos não possui avaliação do item DVD1 e ao mesmo tempo avaliou muito bem o item CD2 e muito mal o item Livro1. Pode-se observar também que há uma similaridade nas avaliações de Marcos e Thiago, pois suas notas estão muito próximas umas das outras. Como a Filtragem Colaborativa indica itens a um usuário baseando-se nas avaliações feitas por outros usuários e sua similaridade, como Thiago avaliou muito bem o item DVD1, esta seria uma boa sugestão de item a ser exibido ao usuário Marcos.

2.2.2 Filtragem cognitiva

Na filtragem cognitiva, também conhecida como filtragem baseada em conteúdo, têm-se informações sobre um determinado conteúdo e informações que possam ser relacionadas a elas sobre um usuário. Podemos então determinar uma relação entre usuário e conteúdo baseado nessas relações. Por exemplo, um artigo sobre programação poderia ser recomendado a estudantes de cursos ligados a computação. Esse é o princípio da recomendação baseada em conteúdo, que ao contrário da filtragem colaborativa, não utiliza relações entre usuários para determinar o conteúdo. Por esse motivo, muitas vezes a recomendação baseada em conteúdo não gera grande surpresa ao usuário, já que a relação e os meios que o sistema usou para

recomendar podem ser inferidos diretamente pelo usuário, mesmo que inconscientemente.

Para viabilizar recomendações baseadas em conteúdo, o uso de taxonomias é necessário para identificar classes de usuários e conteúdos que possibilitem relacioná-los. Uma classe de usuário ou conteúdo pode ser identificada por seu perfil, histórico ou atributos, com a ajuda de algoritmos de classificação. A partir desse ponto, é possível relacionar classes de pessoas e conteúdo de forma compatível.

Para determinar interesses do usuário, pode-se utilizar informações fornecidas de modo ativo pelos próprios usuários, a partir de um cadastro, preenchimento de perfil ou questionários e estruturar essas informações para utilização futura no algoritmo de classificação. Outra opção é obter tais informações de modo passivo, armazenando padrões de navegação, interação, compra e assuntos de interesse do usuário e tratar tais informações através de algoritmos de classificação.

Muitas ferramentas utilizam a técnica de indexação de freqüência de termos Salton [6] para determinar relações na filtragem cognitiva. Nessa técnica, informações sobre conteúdo e usuários são armazenadas em vetores, onde cada palavra corresponde a uma posição. O conteúdo de cada posição armazena o número de ocorrências para determinada palavra. Então, uma comparação entre os vetores pode determinar quão próximo um conteúdo está de um usuário. Outras técnicas que podem ser utilizadas são a busca booleana e filtragem probabilística.

Outra abordagem pode ser descrita como a utilização da folksonomia, na qual os próprios usuários definem palavras-chave para conteúdo. Nessa abordagem o conteúdo não é classificado em grupos pré-definidos, e pode ser mais difícil de definir uma relação entre usuário e conteúdo, devido à esparsabilidade de classes, já que as palavras-chave são definidas de modo livre pelos usuários. Entretanto, o problema da classificação é delegado aos próprios usuários, o que contorna esse problema recorrente na taxonomia.

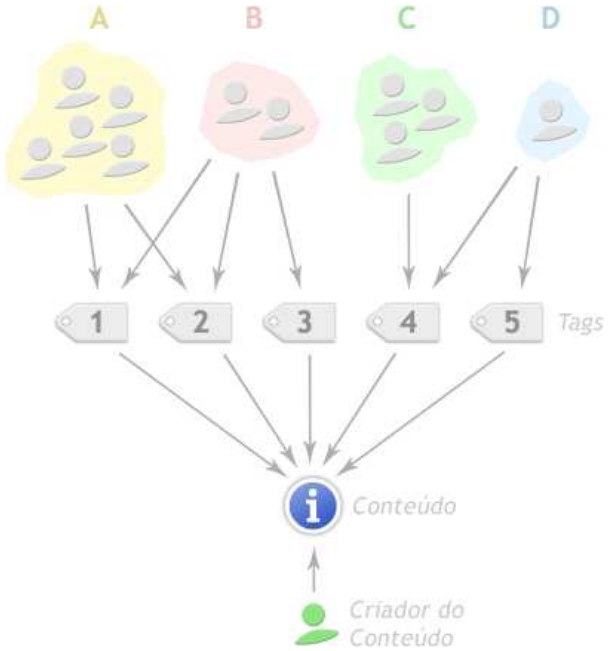


Figura 4: Folksonomia

2.3 Problemas em sistemas de recomendação

Existem problemas conhecidos e comuns nas duas abordagens de sistemas de recomendação, cada um com sua implicação diferente. Abaixo estão listados os principais problemas.

2.3.1 Novo usuário (*Cold Start*)

Quando um usuário é novo no sistema, ainda não existem avaliações realizadas por ele, portanto, seu perfil de avaliações está vazio.

Na filtragem colaborativa, enquanto o novo usuário não executa uma interação com algum item do sistema, não é possível relacioná-lo com outros usuários, portanto torna-se difícil a recomendação de um item. Já na filtragem baseada em conteúdo, a recomendação de itens não depende deste relacionamento inicial, bastando apenas que o sistema tenha informações do perfil do novo usuário.

2.3.2 Ovelha Negra (*Gray Sheep*)

Quando o usuário possui gostos bastante raros fica difícil recomendar algo de seu interesse.

Na filtragem colaborativa, um usuário com este perfil não é facilmente relacionado com outros usuários do sistema, tornando-se difícil recomendar itens a ele. Na filtragem baseada em conteúdo, mesmo que um usuário tenha um perfil raro, a recomendação de itens relacionados a este perfil não é um problema. Por exemplo, um usuário que tenha interesse em assuntos de tecnologia e oceanografia será facilmente recomendado com itens de tecnologia e itens de oceanografia.

2.3.3 Primeira avaliação (*Early-rater*)

Quando um novo item surge no sistema, não é possível recomendá-lo a algum usuário até que outro usuário o avalie.

Este problema é claramente identificado na filtragem colaborativa, já que um item novo não possui recomendações e avaliações dos usuários, não podendo, assim, ser indicado. Enquanto que na filtragem baseada em conteúdo, bastando saber o conteúdo de um item já é possível indicá-lo a um usuário.

2.3.4 Avaliações Esparsas

Quando o sistema possui poucos usuários para muitos itens, as avaliações tornam-se esparsas e fica difícil encontrar usuários similares para serem feitas recomendações.

Na filtragem colaborativa este problema é identificado facilmente. Porém, na filtragem baseada em conteúdo, a recomendação independe do número de usuários e itens, e sim de seus perfis e conteúdos.

2.3.5 Escalabilidade

Quando o volume de usuários, itens e avaliações é muito grande, o sistema que faz o cálculo das relações entre os usuários de forma on-line pode chegar a executá-lo com um tempo de resposta muito elevado.

Este é um problema comum às duas abordagens, porém na filtragem colaborativa este problema é mais evidente, pois os cálculos são feitos entre todos os usuários e todos os itens. Na filtragem baseada em conteúdo os cálculos são feitos entre um usuário somente e todos os itens relacionados.

2.3.6 Super-especialização

Este problema refere-se ao fato do sistema só conseguir recomendar itens muito semelhantes àqueles que o usuário já avaliou. Por exemplo, um usuário do MovieLens [15] que tem o perfil formado basicamente de filmes de guerra, receberá recomendações, em grande parte, de outros filmes de guerra. Isto pode tornar-se um problema, uma vez que os interesses dos usuários tendem a apresentar mudanças com o passar do tempo [16] [17]. Na filtragem baseada em conteúdo, este problema é claramente identificado. Um usuário que já tenha um perfil definido receberá sempre itens relacionados a este perfil, e qualquer mudança de perfil pessoal (fora do sistema) não será acompanhada dentro do sistema. Na filtragem colaborativa, a recomendação de itens não é baseada no perfil inicial do usuário, e sim nas suas ações e relações com outros usuários.

2.3.7 Falta de surpresa na recomendação (*Serendipity*)

Itens que não se relacionam com o perfil de um usuário podem nunca ser recomendados.

Este problema ocorre na filtragem baseada em conteúdo, já que os conteúdos recomendados serão sempre pertencentes a um mesmo grupo relacionado ao perfil do usuário. Enquanto que na filtragem colaborativa, a surpresa ocorre com maior freqüência, já que os usuários relacionados podem ter avalidado itens muito diferentes dos já vistos pelo usuário original.

2.3.8 O conteúdo de alguns tipos de dados ainda não pode ser analisado

O conteúdo de alguns tipos de dados, como vídeo, imagem e som, ainda não pode ser analisado com total precisão. Isso ocorre devido à falta de tecnologia para a análise desses tipos de dados.

Este problema é observado na filtragem baseada em conteúdo, já que a recomendação depende do conteúdo do item que se relacionará com o perfil do usuário. Já na filtragem colaborativa, a análise do conteúdo do item não influência na recomendação que é feita por meio do julgamento dos usuários.

Nota-se que a filtragem colaborativa e a baseada em conteúdo são complementares. As desvantagens que uma possui são supridas pelas vantagens da outra. Deste modo, muitos sistemas utilizam uma abordagem híbrida para sistemas de recomendação, aproveitando, assim, as vantagens de cada uma delas.

2.4 A relevância do recomendador

A popularização de ferramentas que permitem a qualquer usuário criar, opinar e contribuir abertamente para um determinado projeto, idéia ou conhecimento tem sido um grande passo que potencializou como nunca a quantidade de informações disponíveis, mas que também trás a tona a questão da confiabilidade das informações disponíveis.

A confiabilidade do conteúdo gerado pode ser discutida já que especialistas são colocados em mesmo patamar de leigos em determinado assunto, gerando informações imprecisas ou potencialmente incorretas. Considerar então o nível de um usuário na determinação de recomendações passa a ser uma necessidade.

Esse problema fica mais evidente em áreas de conhecimento muito específicas. Imaginando, por exemplo, o caso de uma loja de livros que, colocando especialistas e leigos em mesmo patamar, possa recomendar, para compradores de um ótimo livro, um livro ruim, com diversas incorreções, pois o mesmo havia sido comprado por usuários leigos, em maior quantidade. Nesse caso, os especialistas poderiam ser maior peso na decisão da recomendação, evitando que opiniões de leigos sejam determinantes na escolha.

2.5 Exemplos de aplicações que utilizam sistema de recomendação

1 - Last.fm

O site Last.fm é um serviço gratuito de "rádio personalizada" na internet.

Utilizando um sistema de recomendação chamado *Audioscrobbler,* o sistema armazena o nome de todas as músicas ouvidas pelo usuário em seu computador.

Então, a partir dessas preferências musicais registradas do usuário, o sistema consegue fazer recomendação de outros artistas e músicas utilizando como base para o cálculo os outros usuários de gosto semelhantes (chamados pelo serviço de vizinhos).

Para evitar o problema do novo usuário (*cold start*), o Last.fm só faz a primeira recomendação de músicas ao usuário após o mesmo ter escutado pelo menos cinco músicas de artistas diferentes.

2 – StumbleUpon

StumbleUpon é um serviço na Internet que possibilita as pessoas descobrirem e compartilharem sites, blogs, fotos e vídeos. O sistema faz a recomendação ao usuário, a partir do perfil traçado pelo sistema conforme o usuário avalia as páginas visitadas como boa ou ruim (através de uma extensão instalada no navegador). Dessa maneira, o StumbleUpon consegue fazer a relação dos usuários de gosto parecido e associá-los, tornando possível o sistema de recomendação.

Este sistema foi comprado pelo Ebay em Maio de 2007 por um valor aproximado de 75 milhões de dólares. Em janeiro de 2008, o StumbleUpon conta com mais de 4 milhões e 300 mil usuários.

3. Protótipo Rec6

Em setembro de 2006 foi criado o projeto Rec6 com o intuito de estudar sistemas de recomendação de notícias baseados em votos já em contato direto com o usuário final. Em outubro de 2007, o sistema já dispunha de mais de 5.800 usuários cadastrados e 57.000 notícias inseridas. O modelo conceitual do projeto foi baseado no americano Digg, projeto de muito sucesso nos EUA, que recomenda milhões de notícias baseadas no mesmo tipo de votação.

3.1 Introdução

Para a elaboração desse primeiro protótipo, foi escolhida como objeto de recomendação a filtragem colaborativa, utilizando-se a variante descrita no item 2.2.1.

O Rec6 baseia-se em duas operações principais: inserção e votação de notícias. A inserção de notícias é feita por meio de *hyperlinks*, a partir de um Identificador Unificado de Recurso ou *Uniform Resource Identifier (URI)* válida. Uma vez inserida no sistema, a notícia fica disponível em uma fila de entrada, na qual pode ser votada (recomendada) por usuários para que seja anexada à página principal. Dessa forma, a partir dos votos de usuários, é possível determinar quais as notícias mais relevantes para um determinado público.

Figura 5: Captura da tela do Rec6

A interface do Rec6 é dividida em sete categorias (Tecnologia, Blogosfera, Economia e Negócios, Gestão e Marketing, Acadêmico e Educação, Mundo e Política, Entretenimento) e pode ser descrita em dois grandes grupos, que assim serão referenciados em todo esse trabalho: "capa" e "mais novas". Na "capa", a página padrão de uma categoria, são apresentadas as notícias ordenadas por sua relevância, após a influência de votos dos usuários. Na "mais novas" são apresentadas notícias por ordem de inserção (da mais nova a mais velha), formando uma lista de notícias que poderão então ser promovidas para a capa.

Cada notícia é inserida no sistema por um usuário, que fornece dados para sua identificação (título, *URI* e descrição) e para sua classificação (categoria e palavras-chave). A partir da categoria definida pelo usuário, a notícia é inserida no respectivo grupo. A partir das palavras-chave o sistema será capaz de classificar as notícias em ferramentas externas, como a comunidade virtual Via6.

3.2 Tecnologia

3.2.1 Plataforma

O Rec6 foi criado sob plataforma LAMP (Linux, Apache, MySQL, PHP), acrônimo para a solução de infra-estrutura de software, baseada usualmente em software livre:

1. **Linux**, como sistema operacional.

2. **Apache**, como servidor web.

3. **MySQL**, como servidor de banco de dados relacional.

4. **PHP**, como linguagem de programação.

Em 1998, Michael Kunze, no artigo "*Freeware Web Publishing System*" para a revista de computação alemã c't, utilizou o acrônimo LAMP para descrever essa arquitetura. A combinação dessa solução tem sido utilizada por diversas aplicações Web, como solução diante de outros pacotes comerciais, envolvendo aspectos de custo, escalabilidade e documentação, amplamente disponível para essa plataforma. Também é conhecida como a "plataforma de código aberto".

Uma breve comparação com diversos pacotes comerciais permite concluir que a solução LAMP permite menor custo de instalação, manutenção e muitas vezes suporte. Além disso, diversas aplicações, como Livejournal, Digg, Wikipédia, têm obtido com êxito escalabilidade envolvendo essa plataforma. A documentação segue a mesma filosofia aberta, estando disponíveis diversos materiais e cases de sucesso para consulta.

3.2.2 Infra-estrutura

A arquitetura dos servidores é baseada em *webfarms,* uma coleção de computadores que trabalham em conjunto para disponibilizar um ou mais serviços de forma centralizada. A aplicação tem disponível dois servidores web, dois servidores de banco de dados, um servidor de imagem e um servidor virtual (Linux VS).

Por meio de um sistema de balanceamento de carga, disponibilizado por meio do Linux VS, solução de código aberto para construções de clusters de servidores com alta escalabilidade e disponibilidade criada em 1998, as requisições podem ser distribuídas entre diversos servidores de forma proporcional ao seu poder de processamento/armazenamento. Por meio de diversos algoritmos de escalonamento disponíveis, o Linux VS permite monitorar e balancear a carga entre diversos servidores em diversos papéis diferentes. Para o mundo externo, tudo se comporta como se fosse uma única máquina.

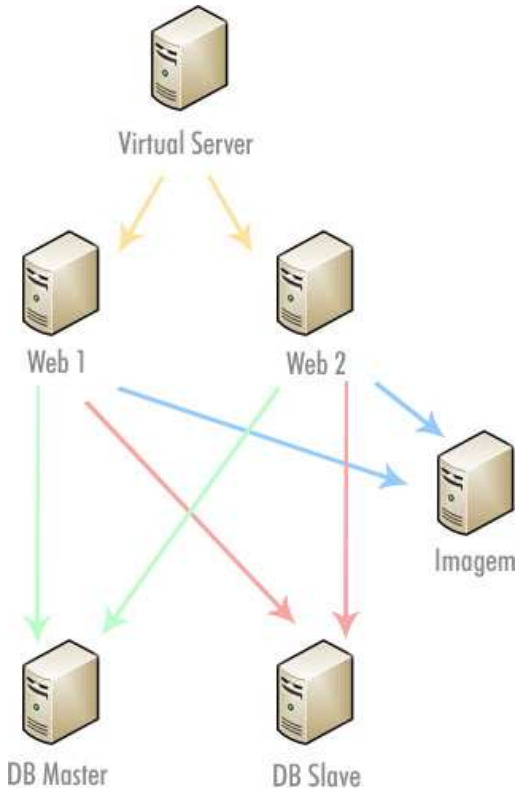


Figura 6: Arquitetura dos servidores

Os dois servidores Web, sob Linux 2.6 e Apache 2, respondem às requisições HTTP de forma compartilhada, recebendo pacotes do Linux VS. Se necessário, fazem requisições ao servidor de imagens, através de uma unidade NFS. Também, acessam o servidor *Master* ou *Slave* do servidor de banco de dados.

Os servidores de banco de dados trabalham na arquitetura *Master-Slave* do MySQL, MySQL Replication. O servidor *Master* é responsável por leituras e gravações em banco de dados, avisam alterações de dados ao *Slave*. O *Slave*, por sua vez, apenas responde a leituras, de modo que está sempre atualizado em relação ao *Master*. Essa solução permite balancear cargas de leituras, normalmente mais custosas entre os dois servidores, sem precisar lidar com uma solução mais complexa de gravação e sincronização de ambos os servidores.

A tabela de versões e distribuições utilizada nos servidores está disponível abaixo:

Responsabilidade	**Software**	**Versão/Distribuição**
S.O.	Linux	Debian 4
Web Server	Apache	2.0
Banco de dados	MySQL	5.1
Interpretador	PHP	5.1
Monitoração	Cacti/MRTG	0.86

Tabela 2: Responsabilidades e softwares da arquitetura de servidores

3.2.3 Padrões de design de software

Para o desenvolvimento do protótipo, utilizamos os seguintes padrões de design de software:

I) MVC

O padrão de design de software MVC (*Model-View-Controller*), utilizada em engenharia de software, foi concebido em 1978, como solução de design para um problema particular. Tem como objetivo dividir a aplicação em três camadas separadas. Em sistemas complexos, que envolvem grande quantidade de dados, estruturas e interfaces, é comum o desejo de se separar aspectos relacionado a dados (*model*) de estruturas de interfaces (*view*). Mudanças de estruturas ou lógica de negócios não necessitam de mudanças na interface do usuário e vice-versa. Para resolver esse problema, o padrão *Model-View-Controller* define camadas responsáveis pelos dados, pela lógica de negócios e interface de interação com o usuário. São elas:

***Model* -** Representa o domínio de uma informação em que a aplicação opera. A classe é responsável pela persistência das informações.

View - Renderiza o *Model*, é responsável pela interface do usuário.

Controller - Processa e responde aos eventos do usuário. Invoca alterações no *Model* e convoca chamadas do *View*.

Dessa maneira, obtém-se uma independência entre a manipulação dos dados e a interface do sistema, tornando possível o desenvolvimento dos mesmos de forma separada.

Esse padrão se tornou popular com utilização massiva em aplicações web e o avanço de linguagens orientadas a objeto. Nessas aplicações, a interface (*view*) é representada pelo código HTML gerado que interage com o usuário. O *model* permite que dados sejam acessados e persistidos e o controller é responsável pelas interações com *model* e *view*.

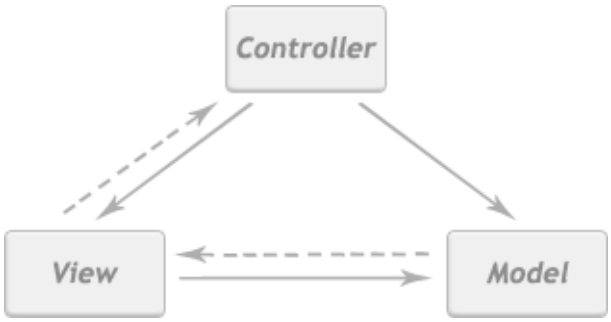


Figura 7: Diagrama que explica a relação entre o Model*, o* View *e o* Controller*. As linhas pontilhadas representam uma ligação indireta, enquanto que as linhas não pontilhadas representam uma ligação direta.*

A implementação da camada *Model* no Rec6 fornece persistência para objetos descritos por ele. Também, uma estrutura de validação de dados que devem ser

configurados junto aos seus atributos. Por isso, é necessário mapear campos dos objetos, seus tipos e chaves primárias.

Exemplo de instanciação e leitura de um *model* do tipo Artigo:

```
$artigo = new Artigo();
$artigo->id_art = 25;
$artigo->read("titulo|subtitulo|texto|autor|fonte");
$artigo->fonte = “Folha de São Paulo”;
$artigo->update("fonte");
```

Implementação do método *update* da superclasse *Model*:

```
public function update($list = null, $condicionais = "") {

      if ($condicionais != "") {
            $atributes = explode("|", $condicionais);

            foreach ($this->atributes as $key => $value) {
                  if (isset ($atributes) && (!in_array($key, $atributes) ||
is_numeric($key)))
                        continue;
                  $sql_data_cond[] = "$key='" . Context :: $dbObj-
>escape($value) . "'";
            }

            $sql_restriction = implode(" AND ", $sql_data_cond);
      }
      else {
            $sql_restriction = $this->getKeyRestriction();
      }

      $serializeAttr = $this->serializeAtributes($list);

      $sql_query = "UPDATE " . $this->tableName .
      " SET " . $serializeAttr .
      " WHERE " . $sql_restriction . " LIMIT 1";

      if($serializeAttr != "")
            Context :: $dbObj->query($sql_query);
}
```

A implementação do *Controller* no Rec6 fornece uma camada comum para captação de eventos da *View*, verificação de permissões e mapeando dados para *models* utilizados por ele.

Exemplo de mapeamento de permissões para os eventos “*edit*” e “*index*” do *Controller* PerfilUsuario:

```
class PerfilUsuario extends ControllerSyxt {

      public $actionMap = Array (
            "edit" => 10,
            "index" => 0
      );

      (...)
}
```

Exemplo de *action* para listagem de histórico de um usuário:

```
function actionList () {

    $hu = new HistoricoUsuario();
    $hu->id_usu = Context::$loggedUser->id_usu;
    $hus = $hu->findAll("id_cur, empresa, cargo, atividades,
"id_usu", "fimAno DESC, fimMes DESC");

    $this->view->assignModelArray('curriculos', $hus);
    $this->view->display("historicoUsuarios_list");
}
```

A implementação da *View* no Rec6 fornece integração e estruturas de controle simples para disponibilizar dados e criar interfaces para interação com o usuário.

Exemplo:

```
    <td class="noticiaFoto" width="65">
          {if $links[sec].usrSyxt == 1 && !$usuarioList}
          <a
href="{$app.site_root_via6}/{$links[sec].syxt_id}">
                <img src="{getUserImg uid=$links[sec].id_usu
tipo=thumb}" class="foto65" />
          </a>

          <div id="assinatura_{$links[sec].id_link}">
          {if $links[sec].id_usu != $loggedUser.id_usu}
                {if $links[sec].assinado == 1}
                      assinado
                {elseif $integracao_via6 != "0"}
                      &raquo;
                      {html_link mode="ajax"
href="assinaturasDoUsuario.php?action=ajaxInsert&id=`$links[sec
].id_usu`&link_id=`$links[sec].id_link`&control=`$control`"
onSuccess="successAssinatura" onFailure="failureAssinatura"}
                                  Assinar
                      {/html_link}
                {/if}
          {/if}
          </div>
          {/if}
    </td>
```

II) *Factory*

O padrão de design de software *factory* provê uma fábrica de objetos de classes que implementam uma mesma interface. Assim, quando precisamos de uma instância dessa classe, a *factory* retorna a instância da mesma de forma dinâmica. Além disso, outra vantagem desse padrão é a não necessidade do conhecimento da classe de implementação, basta que o desenvolvedor conheça apenas a interface.

III) *Singleton*

O padrão de design de software *singleton* possibilita a garantia da unicidade da instância de uma classe. Assim, um objeto que utiliza o padrão *singleton*, sempre será instanciado através de uma função estática desse mesmo objeto. Essa função estática ficará responsável pela garantia da existência e unicidade da instância da classe.

3.2.2 Disponibilidade e Monitoração

Em sistemas web em produção a necessidade de alta disponibilidade é alta, de modo que os usuários possam ser servidos de forma contínua e transparente a falhas. Para garantir alta disponibilidade do sistema, é necessário que:

1. O sistema permita a detecção de falhas ou níveis de alerta (monitoração)
2. O sistema corrija ou permita corrigir em tempo aceitável tais falhas.

A arquitetura desenvolvida para esse protótipo conta com um servidor virtual (Linux VS) capaz de balancear proporcionalmente a carga dos clientes para os servidores web. O Linux VS é um ponto de entrada único para as requisições, tornando toda a estrutura de servidores transparente para o usuário final. Ele mantém uma tabela de servidores reais e detecta a queda de algum deles, através de um sistema de *heartbeat*. Nesse caso, automaticamente retira esse servidor da tabela até que o mesmo volte a responder. Entretanto, tal estrutura gera um ponto falho no sistema: o próprio servidor Linux VS. Portanto, um novo servidor deve detectar a queda do Linux VS e assumir seu papel em caso de falha, também através de um sistema de *heartbeat*.

O repositório de dados utiliza a arquitetura *master-slave*. Por esse motivo o servidor *master* de banco de dados é um ponto falho no protótipo. A queda de tal servidor causaria deixaria a aplicação fora do ar. Entretanto, o mesmo pode ser contornado rapidamente em tempo hábil através do servidor *slave* de banco de dados, que assumiria seu papel. Uma solução para esse problema seria transformar o servidor *slave* em *master*, adotando a arquitetura *master-master*.

A monitoração de níveis de alerta é feita através da análise de consumo de CPU, memória, transferência de rede e número de processos em fila. Através do Cacti[5], um *frontend* do MRTG, é possível gerar gráficos que acompanhem tais indicadores.

---

[5] http://cacti.net/

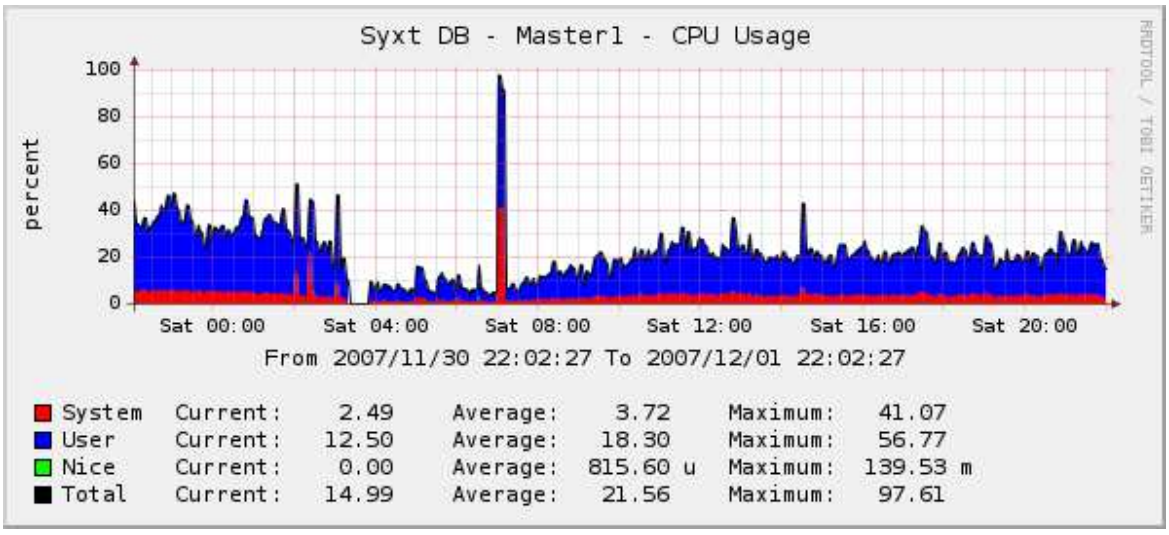


Figura 8: Gráfico de uso do processador do servidor Web 1

3.3 Implementação

O sistema segue três passos básicos:

- **Usuários cadastram notícias**

Um usuário que tenha lido alguma notícia interessante pode cadastrá-la no Rec6. Para isso basta ele ter uma conta no site e clicar no link "Enviar notícia". Ao acessar a página, será solicitado o link da notícia, o título, a descrição, a categoria e três *tags* (palavras-chave). Com essas informações o sistema já tem a habilidade de classificar a notícia na categoria correta, deixá-la disponível para buscas e encontrar notícias relacionadas.

- **Usuários votam em notícias que julgam interessantes**

Um usuário que visita o site com o objetivo de buscar algum conteúdo, lê alguma notícia sugerida pelos outros usuários e caso ache que a notícia é relevante, dá um voto a ela.

- **Sistema filtra e ordena as notícias por relevância**

A partir dos votos dados as notícias, o sistema filtra e ordena as notícias por sua relevância. Com isso, os visitantes têm disponível uma lista de notícias relevantes pronta para ser lida.

Figura 9: Esquema da implementação do sistema

Todas as notícias da ferramenta possuem dois atributos: votos e o valor da sua relevância. Os votos de uma notícia é um número inteiro que representa exatamente o número de votos que ela recebeu. Cada voto aumenta em certo valor a relevância da notícia dependendo do seu peso e suas características, podendo, assim, ocorrer de um voto não aumentar a relevância.

A ordenação das notícias mais relevantes é feita pela relevância e essa listagem é exibida na capa de uma categoria. No "mais novos" são listadas as notícias inseridas recentemente.

3.4 Estatísticas

- Usuários cadastrados: 5.846
- Votos cadastrados: 187.184
- Notícias cadastradas: 57.861
- Média de usuários únicos por mês: 238.741 (referente a Outubro/2007)

Observação: dados referentes ao período: 05/09/2006 a 31/10/2007.

4. Abordagem e desafios da filtragem colaborativa

A abordagem utilizada para esse protótipo, a variação descrita em 2.2.1, levanta alguns desafios a serem enfrentados, comuns a outros tipos de filtragens colaborativas e também próprios dessa abordagem. Em especial, o desafio principal é oferecer um ou mais métodos que ajudem a determinar a relevância de uma notícia x, considerando que a ação do tempo influi diretamente sobre ela: uma notícia muito relevante há um ano pode ter pouca relevância nos dias de hoje. Portanto, para considerar relevância de uma notícia, é essencial considerar o momento t em que ela foi divulgada. Também é necessário considerar fatores como influência social, vantagem cumulativa, votos em grupo com o intuito de promover notícias com interesses pessoais e outros problemas comuns em filtragem colaborativa clássica que podem ser eventualmente aplicáveis a essa variação.

4.1 Normalização Relevância vs. Tempo

Para considerar a relevância de uma notícia é necessário considerar o momento em que ela foi divulgada, além do contexto em que ela se insere. Em geral, uma notícia mais antiga tem menos relevância do que uma notícia mais recente. Isso é uma característica natural de uma notícia, já que ela tem valor jornalístico máximo quando publicada no momento do seu acontecimento e perde valor com o passar do tempo. Considerar essa alteração de relevância é o primeiro passo para normalizar a relevância das notícias inseridas no sistema.

Além disso, em um ambiente com um grande número de notícias sendo inseridas, é interessante que uma notícia perca mais relevância por tempo e, em ambientes com um menor número de notícias sendo inseridas, perca menos relevância; normalizando a rotatividade da capa de modo proporcional ao número de notícias inseridas.

Considerando, por exemplo, um caso de uma página com um grande número de notícias sendo inseridas. Podemos concluir que, se não houver qualquer normalização, as notícias presentes na capa tenderão a permanecer nela por um tempo maior, pois haverá grande concorrência por votos entre as notícias sendo inseridas. Se tivermos 10 usuários votando em 1 entre 100 notícias novas, teríamos em média 0.1 voto por notícia. Se o número de notícias entrando é menor, 10, por exemplo, a média de voto por notícia é 1 e essas notícias ganham mais relevância. Por isso, é necessário normalizar também a relevância das notícias para que intervalos de tempo com maior número de notícias sendo inseridas se comportem como intervalos com menor número.

Essa foi a abordagem utilizada no protótipo do Rec6.

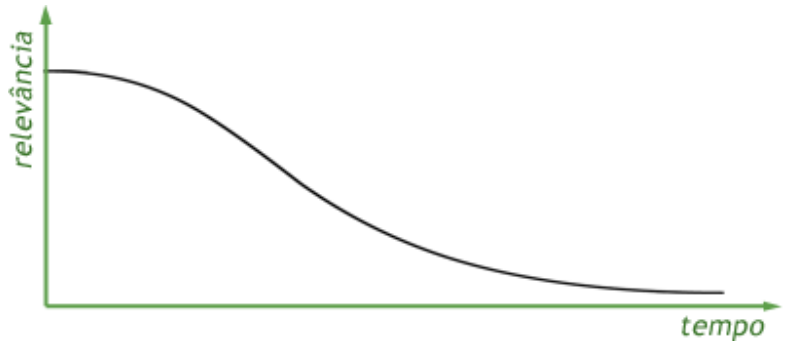


Figura 10: Gráfico da relevância de uma notícia através do tempo

Dessa forma, podemos definir que a relevância rx de uma notícia x é multiplicada por um índice m (0 < m < 1) a cada 24 horas. Ou seja, a cada 24 horas:

```
rx = rx * m
```

Se considerarmos essa execução automática, realizada em intervalos fixos de 10 minutos teremos:

```
m = u^((24*60)/10)  =>  u = raiz(m, 144)

rx = rx * (u)
```

O desafio, portanto é definir o índice m adequado a cada contexto (figura 11).

Para definir m, primeiramente foi definido o índice m0 = 0,5, correspondente a 100 notícias inseridas em 24 horas. A partir daí, o comportamento sugerido para m seria o definido pela reta vermelha da figura 11.

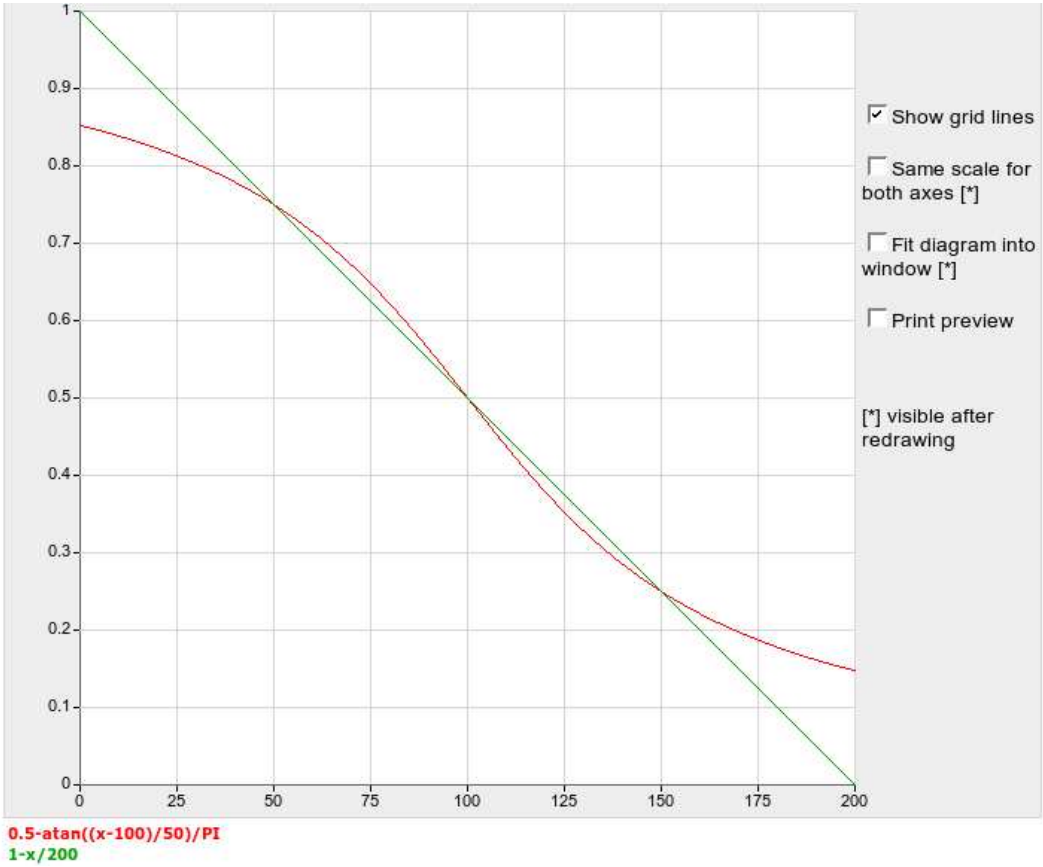


Figura 11: Normalização Relevância vs. Tempo

4.2 Vantagem cumulativa e influência social

Quando um usuário comum entra no Rec6 ele é direcionado para a seção das notícias "Mais Populares" de uma categoria específica. Nesta seção, encontram-se as notícias mais relevantes. Para que outras sejam visualizadas, é necessário que o usuário navegue pelas outras páginas da seção ou selecione outro tipo de visualização: o das notícias "Mais Novas". Verificamos que esta navegação ocorre em um percentual baixo dos usuários (tabela 3).

Seção	Número médio de visitas à seção (por dia)
Mais Populares (capa)	6750
Mais Novos	1322

Tabela 3: Número médio de visitas por seção

Porcentagem de usuários que acessam a Capa e terminam a navegação*:* 84%

Fonte: Google Analytics[6]

Por este motivo, a quantidade de votos feitos na seção "Mais Populares" é maior do que a feita nas na seção "Mais Novos", diminuindo, assim, a chance de uma notícia considerada de boa qualidade que não se encontra na seção "Mais Populares" ter votos suficientes para chegar a ela.

Este problema também sofre influência do caso "primeira avaliação" da abordagem clássica da filtragem colaborativa baseada no usuário, que no nosso caso, é tratada de um modo diferente (ver item 4.4.3). Neste caso, ela fica limitada a uma página de pouca visualização do site, o que acaba sendo prejudicial.

Podemos também analisar outro problema menos impactante que ocorre quando um usuário sofre influência de uma notícia muito votada. Notícias com muitos votos têm maior chance de receber votos do que notícias com poucos votos. Este é um problema de vantagem cumulativa devido à influência social. Deste modo, como as notícias "Mais Populares" são as que possuem maior relevância e em geral possuem mais votos, elas acabam sofrendo esta influência e recebem mais votos.

Figura 12: Influência social

Para amenizar esta situação, cada voto dentro do sistema deve ter o peso proporcional ao peso da página em que ele for feito.

No Rec6, implementamos um sistema que calcula quantos votos uma notícia recebe em média na seção "Mais Populares" e na seção "Mais Novos". A partir destas proporções, conseguimos verificar quantos votos a mais uma notícia da seção "Mais Populares" recebe, fazendo com que um voto em uma notícia fora dela tenha um peso proporcionalmente maior ou menor. Abaixo podemos ver os cálculos efetuados.

6 http://www.google.com/analytics/

`vmp` = Votos que uma notícia recebe em média por dia na seção "Mais Populares"

`vmn` = Votos que uma notícia recebe em média por dia na seção "Mais Novos"

Os votos de uma notícia que está na capa pode ser calculado da seguinte maneira:

```
vcapa = vmp + vmn
```

Isso ocorre, pois uma notícia que está na seção "Mais Populares", também pode ser encontrada na seção "Mais Novos".

Agora, os votos de uma notícia que está nos mais novos é calculado da seguinte maneira:

```
vncapa = vmn
```

Uma notícia que se encontra na seção "Mais Novos", não está obrigatoriamente na seção "Mais Populares". Existe um valor mínimo de relevância que possibilita uma notícia ser exibida na capa. Abaixo deste valor, ela é apenas visualizada na seção "Mais Novos".

Por fim, a proporção de uma categoria é calculada assim:

```
Proporção = (vmp + vmn) / vmn = 1 + (vmp / vmn)
```

Observação: Os cálculos acima não são feitos em tempo real, ou seja no momento do voto. Existe uma rotina que efetua os cálculos das proporções de cada categoria a cada 24 horas. Ela foi criada para evitar o problema de Escalabilidade (ver item 4.4.5) e deixar o processo de votação mais ágil.

Quando um voto é efetuado na seção "Mais Novos", ao invés de incrementarmos a relevância da notícia com o valor exato de um voto, incrementamos seu valor com o número obtido no cálculo da proporção multiplicado pelo valor do voto. Dessa maneira, normalizamos os pesos de cada voto em relação a quantidade de acessos no espaço em que o conteúdo está inserido.

4.3 Métodos para inviabilizar votos inválidos

Para que a variação desenvolvida extraia o máximo de relevância dos votos, é necessário que todo voto passe por um processo de validação e se o mesmo for aprovado, então, sua relevância é computada.

Definição: Um voto da notícia x pelo usuário u (cadastrado no sistema) é válido se e somente se o mesmo não é duplicado ou forjado, ou seja, um voto é válido se e somente se todas as afirmações abaixo são verdadeiras para ele:

a) não existe outro voto de x feito por u no sistema;
b) não existe outro voto de x partindo do mesmo IP de u;
c) IP de u não pertence a uma lista de IPs banidos e/ou representa um *proxy* público;
d) u não está agindo em grupo para promover notícias com interesses pessoais.

Caso uma das afirmações acima não seja válida, a relevância do voto é descartada automaticamente pelo sistema.

Figura 13: Votos inválidos

Da definição acima, temos que o sistema não irá computar votos duplicados (a e b) ou falsos (c) ou de usuários que estão agindo em grupo para promover notícias com interesses pessoais (d).

No caso dos votos duplicados, basta uma verificação rápida na base de dados para a verificação das afirmações a e b. Quanto aos votos falsos, basta gerar uma base com os principais *proxies* existentes na Internet e verificar se a recomendação vem do IP de algum deles. Já para afirmação d, temos um problema um pouco mais complexo.

4.3.1 Usuários agem em grupo para promover notícias com interesses pessoais

Um desafio da variação desenvolvida é o problema dos usuários que agem em grupo para promover notícias com interesses pessoais nessa promoção. Esse problema de confiabilidade da recomendação, que também é informalmente conhecido como "panelinha", pode ser exemplificado da seguinte maneira:

Suponha que um usuário u1 deseja promover a notícia x1. Então, o usuário u1 irá convidar todos os seus contatos (u2, u3, u4, ..., ui) para recomendar sua notícia no sistema. Com isso, x1 poderá vir a ser promovida mesmo não tendo relevância.

Agora, suponha que o usuário u2 deseja promover a notícia x2. Então, o usuário u2 irá convidar todos os seus contatos (u1, u3, u4, y1, y2, ..., yi) para recomendar sua notícia no sistema. Com isso, x2 poderá vir a ser promovida mesmo não tendo relevância.

Não encontramos referência para esse problema em outros estudos e com isso desenvolvemos nosso próprio método para diminuir a ocorrência de notícias sem relevância sendo promovidas.

Método desenvolvido:

Para mostrar o método desenvolvido, inicialmente, precisamos definir o grau de relacionamento entre dois usuários.

Definição: Seja o grau de relacionamento av aplicado a dois usuários (u1 e u2) determinado da seguinte maneira:

$$av = A/B * 100$$

No qual A é o número de notícias votadas por u2 que também foram votadas por u1 nos últimos z dias e; B é o número de notícias votadas por u1 nos últimos z dias.

Em outras palavras, o grau de relacionamento entre u1 e u2 será determinado pela porcentagem de notícias votadas por u1 que também foram votadas por u2 (note que av aplicada a u1 e u2 pode ser diferente do valor de av aplicada a u2 e u1).

Assim, seja u o usuário votante, x a notícia a ser votada, V o grupo dos usuários que já votaram em x e Amin um limitante inferior do grau de relacionamento para que a relevância do voto seja descartada.

- Então, para cada v pertencente a V, determina-se a relação av entre u e v;
- Se av > Amin, para qualquer v, então, o voto de u não atua sobre a relevância de x.

Explicando com outras palavras, temos que no momento da recepção de um voto, o método analisa o grau de relacionamento entre o usuário votante e os usuários que já votaram na notícia. E caso, o grau de relacionamento seja maior que um limitante inferior quando calculado entre o usuário votante e algum votante prévio, a relevância do voto do usuário atual é descartada.

Problemas de desempenho

Com a implementação do método desenvolvido para impedir o problema dos usuários que agem em grupo para promover notícias com interesses pessoais, obtém-

se um problema de desempenho do sistema. Como o sistema é obrigado a calcular o grau de relacionamento do usuário votante com todos os outros usuários que já votaram na notícia a cada voto, então, é necessário que o algoritmo seja otimizado.

Para isso, uma solução viável, é recalcular e armazenar em intervalos de tempo definidos, por exemplo, 24 horas, os graus de relacionamento entre os usuários. Dessa maneira, restringimos a falta de desempenho a um horário programado, que pode ser agendado para o horário em que o sistema é menos requisitado, e que, portanto, não afetará diretamente a maioria dos usuários.

Com essa otimização, nosso método sofreria uma pequena mudança:

Seja u o usuário votante, x a notícia a ser votada, V o grupo dos usuários que já votaram em x e Amin um limitante inferior do grau de relacionamento para que a relevância do voto seja descartada.

- Então, para cada v pertencente a V, lê-se a relação av entre u e v armazenada;
- Se av > Amin, para qualquer v, então, o voto de u não atua sobre a relevância de x.

Penalizações

Todo usuário, que tem um voto invalidado por estar agindo em grupo para promover notícias com interesses pessoais, recebe uma punição. Esta punição refere-se a perda de 10% da relevância de seu voto a cada voto invalidado por esse motivo.

Nota: Todo usuário do sistema, que não possui 100% da relevância de seu voto, recebe um acréscimo diário de 3% da relevância de seu voto. Essa abordagem permite que usuários penalizados voltem gradativamente a ter relevância para o sistema com o tempo, desde que não sejam penalizados novamente.

4.3.2 Problemas resolvidos

Com a validação de um voto feita dessa maneira, reduzimos drasticamente os seguintes problemas do sistema:

- Criação de usuários falsos por uma mesma pessoa para votar em sua própria notícia;
- Criação de scripts/robôs que utilizam um mesmo IP ou IP de *proxies* mais conhecidos para efetuar votos em uma notícia;
- Criação de "panelinhas" entre os usuários para promoção das próprias notícias.

4.3.3 Problemas que ainda precisam ser resolvidos

Com o bloqueio dos votos a partir de um mesmo IP, acabamos por bloquear o voto de pessoas diferentes, mas que estão "atrás" de um mesmo IP, por exemplo, em uma empresa onde existe apenas um computador (um IP) que serve Internet aos outros computadores. Além disso, em alguns casos, o usuário ainda pode conseguir IPs diferentes desconectando-se e conectando-se novamente a seu provedor.

Também, é praticamente impossível ter uma lista completa de todos os *proxies* existentes na Internet e, portanto, não podemos garantir que todos os *proxies* estão bloqueados com esse método.

4.4 Outros desafios da filtragem colaborativa

Abaixo serão listados desafios comumente encontrados na filtragem colaborativa clássica que também podem ser analisados na abordagem utilizada na ferramenta Rec6 e como esses problemas foram abordados. A descrição de cada desafio pode ser encontrada na seção 2.3 deste documento.

4.4.1 Novo usuário (*Cold Start*)

Na ferramenta Rec6, a recomendação de conteúdo relevante não é baseada nas avaliações prévias de um usuário somente, mas sim, de vários usuários dentro de uma categoria. Deste modo, o desafio enfrentado diante de um novo usuário é encontrar a categoria adequada a ser exibida, com suas notícias mais relevantes.

Uma possível solução para este problema seria exigir no cadastro alguma informação do usuário que auxilie na identificação de uma categoria adequada para ser exibida no acesso inicial, por exemplo, sua profissão, interesses pessoais, idade, etc.

Outra solução é a criação de uma seção que englobe todas as categorias da ferramenta, com as notícias mais relevantes de cada uma. Esta seria a seção visualizada pelos novos usuários.

Uma funcionalidade já implementada no sistema que auxilia na exibição de uma categoria relevante para o usuário é de sempre direcioná-lo para a última categoria visitada por ele. Assim, tentamos garantir que a categoria exibida no acesso inicial à ferramenta seja de seu interesse.

4.4.2 Ovelha Negra (*Gray Sheep*)

No Rec6 o usuário está condicionado a visualizar notícias de um mesmo tipo quando está dentro de uma categoria. Não existe uma solução prática para usuários com gostos raros, já que a ferramenta abrange somente as notícias das categorias que ela possui.

4.4.3 *Early-rater*

Na ferramenta Rec6, quando uma notícia é enviada ela já recebe uma primeira avaliação do usuário que a enviou. Porém, esta situação é tratada de outra maneira no Rec6: as novas notícias que surgem no sistema ficam limitadas à seção "Mais Novas" aguardando por avaliações, enquanto a maior parte dos usuários acessa somente a seção "Mais Populares" da ferramenta.

Uma maneira de amenizar este problema é incentivar o acesso à seção "Mais Novas" da ferramenta, aumentando a possibilidade de notícias receberem votos

suficientes para serem exibidas nas primeiras páginas da seção "Mais Populares" e, logo, serem recomendadas aos usuários.

4.4.4 Avaliações Esparsas

No Rec6, em caso de avaliações esparsas, a lista de notícias recomendadas ("Mais Populares") fica menos rotativa, já que poucos usuários estão votando nas notícias do sistema. Este problema é amenizado por meio de duas funcionalidades implementadas no Rec6 (Normalização Relevância vs. Tempo; Vantagem Cumulativa e Influência Social) que são explicadas nas seções 4.1 e 4.2 deste documento.

4.4.5 Escalabilidade

A natureza do sistema de recomendação baseado em votos diminui a necessidade de processamento, pois não precisa realizar cálculo de relações entre usuários, já que essa relação é definida pelo próprio usuário ao visitar uma categoria. Isso torna o sistema computacionalmente mais viável e escalável, embora possa tornar maior o grau de incerteza da recomendação.

4.4.6 Super-especialização

O Rec6 utiliza um sistema de filtragem colaborativa baseado no grupo a que o usuário pertence e não baseado no usuário isoladamente. Isso significa que a recomendação de notícias é separada por categorias, por exemplo, Tecnologia. Deste modo, ao entrar em uma categoria, o usuário está ciente de que verá somente notícias relacionadas àquela categoria. Qualquer mudança de interesse de um usuário requer o acesso a outra categoria compatível com seu novo interesse.

4.4.7 Falta de surpresa na recomendação (*Serendipity*)

Na ferramenta Rec6, um usuário que possui um perfil e acessa categorias específicas nunca receberá recomendações de notícias de outras categorias. Neste caso, ele nunca poderá ser surpreendido com uma notícia que eventualmente seja de seu interesse. Isso ocorre devido à variação desenvolvida para o Rec6, que utiliza categorias para agrupar notícias do mesmo assunto.

4.4.8 O conteúdo de alguns tipos de dados ainda não pode ser analisado

Um método para facilitar a análise do conteúdo de um tipo de dado é através de descrições textuais, como por exemplo, palavras-chave. Na ferramenta Rec6, para toda notícia inserida é exigido o preenchimento de três palavras-chave (*tags*) que facilitam a definição do conteúdo da notícia. Apesar desta funcionalidade, este problema não afetaria diretamente as notícias inseridas no sistema, já que elas são compostas apenas de textos que são facilmente analisados.

Abaixo, os problemas listados foram identificados na variação desenvolvida para o Rec6 e podem não ser aplicados a outros sistemas de recomendação.

4.4.9 Inserção de conteúdo impróprio ou inválido

Em sistemas em que o conteúdo é gerado principalmente por usuários, é muito comum o surgimento de itens que não correspondem com o objetivo do sistema ou que são inválidos. No caso do Rec6, itens como pornografia e violência não são permitidos.

Para isso, foi criado um sistema de banimento de domínios e denúncia de notícias. Qualquer usuário do sistema pode denunciar uma notícia como duplicada, *spam*, duvidosa, inadequada, com link quebrado ou que viola direitos autorais. As notícias denunciadas ficam marcadas para que outros usuários sejam desencorajados a votar nelas.

Figura 14: Denúncia de uma notícia

Outro método desenvolvido foi o banimento de domínios. Qualquer usuário que tentar inserir uma noticia de um domínio que tenha sido banido devido a uma grande quantidade de denúncias ou inserção de conteúdo impróprio, não conseguirá fazê-lo.

4.4.10 Usuário comum não entende facilmente um sistema de recomendação

É muito comum um usuário novo não entender como funciona um sistema de recomendação, porém é fundamental que os usuários do sistema entendam-no e colaborem para torná-lo mais relevante a todos.

No Rec6, existe um documento chamado "Instruções de Uso" com uma explicação de como funciona a ferramenta, como inserir notícias e como votar corretamente.

5. Conclusão

A busca por informação relevante na internet está deixando de ser uma tarefa fácil e corriqueira, devido ao enorme aumento de informações disponíveis, potencializado pelo advento da Web 2.0. Além dos veículos de mídia tradicionais, os próprios usuários criam e compartilham conteúdo por meio de blogs, vídeos, compartilhamento de *links*, perfis em redes sociais, comentários, fóruns e e-mails.

Sistemas de recomendação têm o objetivo de otimizar ferramentas de busca e sugerir conteúdo durante a navegação do usuário. O objetivo desse trabalho foi estudar esses sistemas e propor um sistema de recomendação de notícias baseado em votações dos usuários.

Como resultado, um sistema protótipo, o Rec6, foi criado e colocado em produção, com mais de 5 mil usuários. Com ele, pudemos implementar um sistema de recomendação de notícias baseado em votos dos usuários, descobrir e combater seus principais problemas.

Esse tipo de abordagem de filtragem colaborativa consegue desviar de problemas encontrados em outras abordagens, como o *early-rater*; mas traz outros desafios, principalmente ligados à influência de grupos de usuário com interesses próprios. O grande desafio foi minimizar esses problemas.

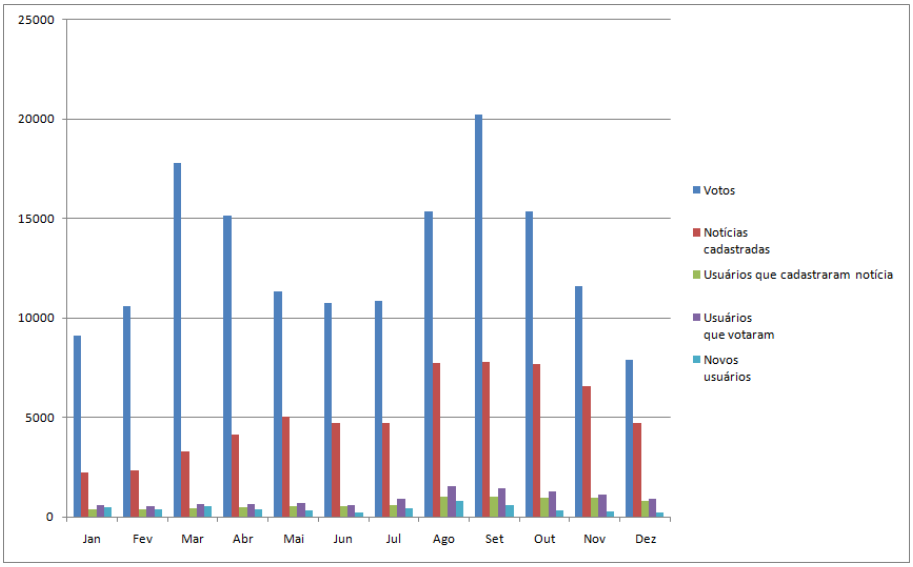
Entretanto, dentre todos os trabalhos realizados, pesquisados e constatados em sistemas já estabelecidos, não foi possível encontrar um método definitivo para resolver esses problemas. É possível minimizá-los de forma eficiente para a maioria dos casos, que é o que se propôs e foi apresentado neste trabalho.

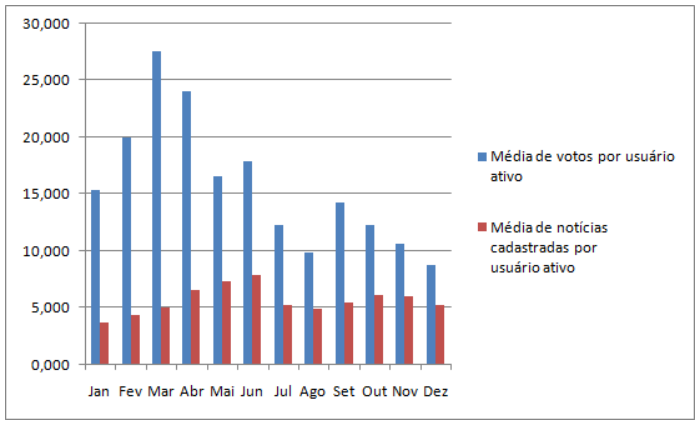
Portanto, o desafio futuro é trabalhar nesses problemas e expandir o universo de trabalho para um ambiente aberto, como em um sistema de busca. Para isso, é necessário um estudo aprofundado de mineração e classificação de dados.

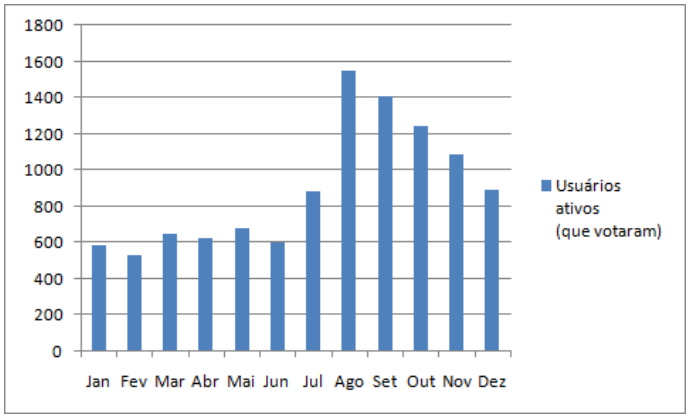
Dados referentes ao ano de 2007 sobre o Rec6

	Votos	Notícias cadastradas	Usuários que cadastraram notícia	Usuários que votaram	Novos usuários
Jan	9086	2201	367	589	481
Fev	10592	2331	350	531	389
Mar	17823	3269	450	646	521
Abr	15146	4155	502	628	372
Mai	11321	5007	537	682	332
Jun	10756	4724	518	600	235
Jul	10859	4692	598	884	402
Ago	15347	7732	1011	1552	796
Set	20207	7800	999	1412	583
Out	15338	7679	975	1245	337
Nov	11619	6585	937	1090	241
Dez	7884	4723	788	892	225
Total	155978	60898	8032	10751	4914

	Média de votos por usuário ativo	Média de notícias cadastradas por usuário ativo	Usuários ativos (que votaram)
Jan	15,426	3,737	589
Fev	19,947	4,390	531
Mar	27,590	5,060	646
Abr	24,118	6,616	628
Mai	16,600	7,342	682
Jun	17,927	7,873	600
Jul	12,284	5,308	884
Ago	9,889	4,982	1552
Set	14,311	5,524	1412
Out	12,320	6,168	1245
Nov	10,660	6,041	1090
Dez	8,839	5,295	892







6. Bibliografia

[1] CAZELLA, Sílvio César. *Applying user's opinion relevance in a Recommender System to researchers* <http://www.inf.unisinos.br/%7Ecazella/papers/VersaoFinal2006TeseSilvioCazellahomologacao.pdf>.

[2] RESNICK, Paul; VARIAN, Hal R. *Recommender Systems*. Commun. ACM, Março 1997 - <http://portal.acm.org/citation.cfm?id=245108.245121>.

[3] VOSS, Jakob (2007). *Tagging, Folksonomy & Co-Renaissance of Manual Indexing?*. 2007 - <http://arxiv.org/abs/cs/0701072>.

[4] LERMAN, Kristina. *Social Networks and Social Information Filtering on Digg*. 2006 - <http://arxiv.org/abs/cs/0612046>.

[5] GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John M. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

[6] HERLOCKER, Jonathan Lee. *Understanding and Improving Automated Collaborative Filtering Systems*. 2000. 220 f. Tese (Doutorado em Ciência da Computação) - University of Minnesota, Minnesota.

[7] Site: *Folksonomia – Wikipedia* <http://pt.wikipedia.org/wiki/Folksonomia>.

[8] O'REILLY, Tim (30/09/2005). *Site: What's Web 2.0* <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html?page=1>. Visitado em 8 de Agosto de 2006.

[9] O'REILLY, Tim (17/07/2006). *Site: Levels of the Game: The Hierarchy of Web 2.0 Applications* <http://radar.oreilly.com/archives/2006/07/levels_of_the_game.html>. Visitado em 8 de Agosto de 2006.

[10] Site: *Digg / About Us* <http://www.digg.com/about>.

[11] Site: *Sobre a Empresa – Myspace.com* <http://www.myspace.com/index.cfm?fuseaction=misc.aboutus>.

[12] Site: *Via6 – Sobre* <http://www.via6.com/sobre.php>.

[13] Site: *Twitter – What are you doing?* <http://twitter.com/>.

[14] Site: *Wikipedia: Sobre a Wikipedia* <http://pt.wikipedia.org/wiki/Wikipedia:Sobre>.

[15] Site: http://www.movielens.org/

[16] ADOMAVICIUS, Gediminas; TUZHILIN, Alexander. *Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions*. IEEE Transactions on Knowledge and Data Engineering, Nova Iorque, v. 17, n. 6, p. 734-749, Junho de 2005.

[17] BALABANOVIC, Marko; SHOHAM, Yoav. *Fab: Content-Based, Collaborative Recommendation*. Communications of the ACM, Nova Iorque, v.40, n.3, p. 66-72, Março de 1997.

[18] MALONE, Thomas W; GRANT, Kenneth R; TURBAK, Franklyn A; BROBST, Stephen A; COHEN, Michael D. *Intelligent information-sharing systems*. Communications of the ACM, p. 390–402, Maio de 1987.