

Recomendação de Tags a Partir de Métricas de Qualidade de Atributos Textuais em Aplicações da Web 2.0*

Fabiano Belém, Eder Martins, Jussara Almeida, Marcos Gonçalves e Gisele L. Pappa
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
{fmuniz, ederm, jussara, mgoncalv, glpappa}@dcc.ufmg.br

RESUMO

Diversas aplicações populares da Web 2.0 permitem aos usuários a livre atribuição de palavras-chave (tags) ao conteúdo, para prover uma melhor organização e descrição do mesmo. Nessas aplicações, serviços de recomendação de tags podem auxiliar o usuário nessa tarefa, melhorando a qualidade da informação disponível e, conseqüentemente, a eficácia de diversos serviços de Recuperação de Informação (RI) que exploram tags, tais como busca e categorização. Este trabalho trata do problema de recomendar tags de alta qualidade a partir não somente de tags previamente associadas ao conteúdo de um objeto como também de termos presentes em outros atributos textuais (por exemplo, título e descrição) associados ao objeto. Nele, propomos o uso dessas duas fontes de informação para geração de termos candidatos e a combinação de diversas métricas heurísticas em uma função para estimar a qualidade das tags. Para tentar gerar e evoluir novas funções de *ranking*, utilizamos como ferramenta um algoritmo de Programação Genética (PG). A função proposta e as geradas pela PG foram avaliadas em diferentes cenários, com diferentes quantidades de entrada disponíveis, em diferentes aplicações populares da Web 2.0: LastFM, YahooVideo e YouTube. Nossos resultados experimentais indicam que a heurística e as funções geradas pelo GP são mais eficazes que algoritmos de recomendação de tags considerados estado-da-arte.

ABSTRACT

Several popular Web 2.0 applications allow users to assign keywords (tags) to their objects, in order to provide a better description and organization of the shared content. Tag recommendation services may assist users in that task, improving the quality of the available content and, thus, the

***Tag Recommendation Based on Quality Metrics in Web 2.0 Applications.** Esta pesquisa é parcialmente financiada pelo Instituto Nacional de Ciência e Tecnologia para a Web - INCTWeb (MCT/CNPq 573871/2008-6), CNPq, FAPEMIG e CAPES.

effectiveness of tag-based Information Retrieval (IR) services, such as searching and classification. This work addresses the task of recommending high quality tags by exploiting not only previously assigned tags, but also terms extracted from other textual features (e.g., title and description) associated with the target object. These sources are exploited to generate candidate terms for recommendation and to compute heuristic metrics that estimate the quality of a candidate. In particular, we propose a heuristic function that combines multiple metrics to produce a final ranking of the recommended tags. We also investigate the use of Genetic Programming (GP) as a tool to generate and evolve ranking functions. We evaluate both the heuristic and GP-based functions in various scenarios, for three popular Web 2.0 applications, namely, LastFM, YahooVideo and YouTube. Our experiments show that our heuristic and GP-based functions significantly outperform state-of-the-art tag recommendation algorithms.

Categories and Subject Descriptors

H.3.5 [Information Storage and Retrieval]: Online Information Services

General Terms

Algoritmos, Experimentação

Keywords

Recomendação de Tags, Qualidade da Informação

1. INTRODUÇÃO

O uso de tags (palavras-chave associadas a um conteúdo) popularizou-se na Web, principalmente com o surgimento de aplicações Web 2.0, que facilitaram e estimularam a participação dos usuários na criação de conteúdo. Esse conteúdo, tipicamente multimídia (e.g., imagens e vídeos), traz desafios aos atuais métodos de Recuperação de Informação (RI) baseados em conteúdo, devido a questões de escalabilidade e eficiência, particularmente frente à (tipicamente pobre) qualidade do material gerado por usuários. Tags, entre outros atributos textuais como título e descrição, constituem uma boa alternativa para organização, disseminação e recuperação de conteúdo. De fato, estudos recentes demonstram que tags estão entre os melhores atributos para dar suporte a serviços de RI tais como classificação automática [5, 4].

A recomendação de tags é um serviço provido por algumas aplicações com o objetivo de melhorar a qualidade da

informação disponível, por meio da sugestão de termos que, idealmente, descrevam o conteúdo de forma mais precisa e completa. As estratégias de recomendação de tags existentes exploram primordialmente tags previamente associadas a objetos da coleção, expandindo um conjunto inicial de tags \mathcal{I}_o de um objeto o com termos que co-ocorrem frequentemente com os termos de \mathcal{I}_o [8, 19, 6].

Neste trabalho, estendemos a abordagem tradicional para incluir, além de tags previamente utilizadas, o texto contido em outros atributos textuais, tais como título, descrição e comentários de usuários, como fonte de informação para a recomendação. O conteúdo desses atributos é utilizado para extrair termos candidatos à recomendação e para calcular várias métricas. Algumas dessas métricas são heurísticas que estimam a qualidade dos termos recomendados. Qualidade aqui se refere a dois aspectos principais abordados em [5, 4]: (1) *poder descritivo* ou quanto um termo está relacionado ao conteúdo a que foi associado, descrevendo-o de forma precisa, e (2) *poder discriminativo*, ou seja, a capacidade de um termo de distinguir objetos em níveis de relevância quando eles são retornados por uma consulta ou em categorias para serviços como classificação automática de objetos, recomendação e propaganda. Tais métricas, que podem ser computadas automaticamente, foram propostas anteriormente, mas a sua aplicação ao problema de recomendação de tags é nova e adequada pois permite a um recomendador a sugestão dos termos de mais alta qualidade.

Assim, modelamos o problema de recomendação de tags como um problema de ordenação de múltiplos termos candidatos por relevância para recomendação. Em outras palavras, objetiva-se desenvolver uma função que atribua valores aos termos candidatos, possibilitando a sua ordenação de tal forma que termos que representem recomendações de mais alta qualidade (mais apropriadas para o conteúdo) apareçam primeiro que termos que representem recomendações de menor qualidade. A hipótese na qual esta proposta se baseia é que termos de maior qualidade (conforme capturado pelas métricas utilizadas) são mais relevantes para a tarefa de recomendação de tags. Logo, a solução para este problema é a função (cujas variáveis são as métricas propostas) que melhor estime a relevância das candidatas à recomendação e, conseqüentemente, sugira as tags mais adequadas ou mais relevantes a um conteúdo.

Para resolver esse problema, implementamos um método considerado estado-da-arte em recomendação [19], aqui referenciado por Sum^+ . Este método gera tags candidatas e as ordena explorando apenas métricas relacionadas a co-ocorrência e frequência de tags. A nossa primeira contribuição é uma extensão da linha de base, aqui referenciada por Sum^+E , para incluir não somente métricas relativas à co-ocorrência de tags, como também métricas de qualidade baseadas no conteúdo dos outros atributos textuais associados ao objeto para o qual a recomendação será feita.

Nós também investigamos o potencial de utilizar a Programação Genética (PG) [2] como arcabouço de solução para o problema de recomendação de tags. A PG permite explorar o espaço de possíveis soluções rumo às direções mais promissoras, i.e., uma função que melhor estime a relevância de um termo para a recomendação.

Avaliamos tanto a heurística Sum^+E quanto a solução de PG, comparando os resultados com a heurística linha de base Sum^+ . Os experimentos foram realizados com dados de aplicações populares da Web 2.0, a saber, os sites de

compartilhamento de vídeos YouTube¹ e YahooVideo² e a rádio online LastFM³. A função proposta Sum^+E obteve resultados significativamente melhores (até 73% de ganho em precisão) em relação a Sum^+ . O algoritmo de PG obteve resultados tão bons quanto ou até 4% melhores que Sum^+E .

O restante deste artigo está organizado como segue. A Seção 2 discute trabalhos relacionados enquanto a Seção 3 descreve o problema e as principais métricas utilizadas para recomendação de tags. Os três métodos analisados são descritos na Seção 4 enquanto a avaliação experimental feita é apresentada na Seção 5. Conclusões e trabalhos futuros são apresentados na Seção 6.

2. TRABALHOS RELACIONADOS

Há uma grande diversidade de trabalhos relacionados a conteúdo gerado por usuários, particularmente tags, e a sua aplicação para melhorar serviços como busca, recomendação, agrupamento e indexação [12, 17, 3, 14, 9]. Particularmente no contexto de recomendação de tags, a maioria dos trabalhos anteriores têm foco em medir padrões de co-ocorrência de termos para expandir um conjunto inicial de tags associado a um objeto [19, 6, 8, 21, 22, 11]. Para isso, Heymann *et al.* [8] utilizam regras de associação, selecionando as regras mais importantes a partir de um limiar de confiança, embora nenhuma ordenação das tags recomendadas por relevância seja produzida. Sigurbjornsson e Zwol [19] propõem métricas globais de co-ocorrência de termos como o coeficiente de Jaccard e a confiança. De certa forma, os autores consideram métricas de qualidade, mas apenas relacionadas à frequência das tags na coleção. Garg e Weber [6] propõem um método de recomendação de tags personalizado que explora o conhecimento sobre o comportamento passado de cada usuário, combinando informação local (relativas ao histórico de *tagging* de um usuário específico) e global (relativa ao histórico de todos os usuários do sistema). Krestel *et al.* [11] utilizam *Latent Dirichlet Allocation* (LDA) para expandir um conjunto pequeno de tags previamente associadas a um objeto. Os autores concluem que, em comparação com o método proposto em [8], o uso de LDA provê maior acurácia e recomendações mais específicas.

Além da co-ocorrência de tags, outras evidências também já foram exploradas para recomendação de tags. Wu *et al.* [21] acrescentam atributos extraídos do conteúdo de imagens do Flickr em um algoritmo de *ranking* de tags. Lipczak [13] considera como candidatas as palavras do título de um objeto e palavras geradas pelo usuário para o qual a recomendação será feita. Song *et al.* [20] utilizam um grafo bipartido de tags, documentos e palavras, e aplica um *Two-Way Poisson Mixture Model* em agrupamentos obtidos sobre estas três entidades. Dado um documento como consulta, o algoritmo computa suas probabilidades posteriores sobre os agrupamentos e em seguida ordena as tags por relevância considerando essas probabilidades. Uma abordagem baseada em conteúdo para expandir um conjunto de tags é descrita em [18]. Os autores utilizam conteúdo duplicado em vídeos para criar ligações em um grafo e propagar tags entre vídeos similares. Outros trabalhos exploram redes sociais entre usuários [7, 10, 15]. Guan *et al.* [7] produzem recomendações personalizadas a partir de um grafo bipartido de

¹<http://www.youtube.com>

²<http://www.yahoo.com/video>

³<http://www.last.fm>

usuários, documentos e tags. Konstas *et al.* [10] exploram os relacionamentos de amizade entre usuários do LastFM para melhorar os métodos tradicionais de recomendação de objetos baseados apenas no conteúdo dos mesmos.

Embora haja uma diversidade de estratégias de recomendação na literatura, nosso trabalho é o primeiro a adotar métricas da qualidade da informação para explorar atributos textuais de um objeto visando amenizar, nas recomendações, o impacto causado pelo conteúdo de baixa qualidade gerado por usuários. Outra contribuição desse artigo é a utilização dos diferentes atributos textuais para estimar a qualidade dos termos candidatos, visto que trabalhos anteriores ou não exploram todos os atributos disponíveis [13] ou não produzem um *ranking* dos termos recomendados [8]. Por fim, este é o primeiro trabalho a aplicar um algoritmo de PG ao problema de recomendação de tags, obtendo sucesso na geração e evolução de funções para ordenar termos candidatos por relevância.

3. RECOMENDAÇÃO DE TAGS ASSOCIATIVA E CONTEXTUAL

O problema de recomendação de tags pode ser definido da seguinte maneira: dado um conjunto \mathcal{I}_o de tags previamente associadas por usuários a um objeto o , e o conjunto de atributos textuais (exceto tags) $\mathcal{A}_o = \{\mathcal{A}_o^1, \mathcal{A}_o^2, \dots, \mathcal{A}_o^n\}$, onde cada elemento \mathcal{A}_o^i é formado pelo conjunto de termos contidos no atributo textual i do objeto o , gerar um conjunto de termos candidatos \mathcal{C}_o , e sugerir os k termos mais relevantes desse conjunto, resultando em um conjunto de tags ordenadas por relevância $C_{k,o}$.

Muitas estratégias de recomendação de tags, em particular as propostas neste trabalho, exploram padrões de co-ocorrências de tags através da mineração de relações entre tags associadas a um mesmo objeto em uma coleção de objetos. O processo de aprendizagem de tais padrões é definido a seguir. Temos um conjunto de dados de treino $\mathcal{D} = \{\mathcal{I}_d\}$, onde \mathcal{I}_d contém todas as tags associadas ao objeto d . O conjunto de teste \mathcal{O} contém objetos da forma $o = \langle \mathcal{I}_o, \mathcal{A}_o, \mathcal{Y}_o \rangle$, onde ambos \mathcal{I}_o e \mathcal{Y}_o são conjuntos de tags associados ao objeto o . Porém, enquanto as tags em \mathcal{I}_o são conhecidas com antecedência (fornecidas ao recomendador), as tags em \mathcal{Y}_o são desconhecidas, e utilizadas como gabarito para avaliação automática do recomendador. Isso simula a situação em que “novas” tags (\mathcal{Y}_o) são recomendadas para um objeto com base em tags já presentes naquele objeto (\mathcal{I}_o), além de permitir uma avaliação automática das recomendações, como veremos na Seção 5.2. O conjunto \mathcal{A}_o reflete o *contexto* em que o objeto está inserido, composto por título, descrição, entre outros atributos textuais.

Assim, definimos métodos de recomendação de tags associativa e contextual como aqueles que estimam a relevância de tags a um objeto com base nos elementos descritos acima. A seguir apresentamos várias métricas que podem ser exploradas para recomendação de tags. Algumas delas, como *Sum* e *Estabilidade*, já foram aplicadas em funções de recomendação previamente propostas [19], e são apresentadas na Seção 3.1 e 3.2. Outras, como o IFA e o Espalhamento, foram propostas em [5, 4] para avaliação de qualidade de atributos textuais na Web 2.0.

3.1 Co-ocorrência de termos

Para definir as métricas relacionadas à co-ocorrência de

termos, recorremos ao conceito de regra de associação, uma implicação $\mathcal{X} \rightarrow y$, onde o antecedente \mathcal{X} é um conjunto de tags e o consequente y é uma tag candidata. As principais medidas da importância de uma regra de associação são: (1) **suporte** (σ): frequência com que o conjunto $\mathcal{X} \cup \{y\}$ aparece em \mathcal{D} ; e **confiança** (θ): probabilidade condicional de y estar associado a um objeto $d \in \mathcal{D}$ dado que $\mathcal{X} \subseteq \mathcal{I}_d$.

É comum utilizar limiares de *suporte* σ_{min} e *confiança* θ_{min} para esses valores para selecionar apenas as regras mais frequentes e/ou mais confiáveis, o que pode prover ganhos em eficiência e na qualidade ao recomendador. Geramos esse conjunto seletivo de regras \mathcal{R} eficientemente com o *Apriori* [1], um algoritmo clássico de geração de regras de associação que realiza podas na busca por conjuntos frequentes a partir da premissa de que se um conjunto S for infrequente, então todos os superconjuntos de S também serão infrequentes.

Por fim, é necessário combinar as diversas regras geradas para uma mesma tag candidata. Uma possível estratégia para isso é somar os valores de confiança. Dessa forma, a relevância de uma tag candidata t para um objeto o é calculada como:

$$Sum(t, o) = \sum_{\mathcal{X} \subseteq \mathcal{I}_o} \theta(\mathcal{X} \rightarrow t), \quad (\mathcal{X} \rightarrow t) \in \mathcal{R} \quad (1)$$

Para fins práticos, utilizamos apenas as regras em que $|\mathcal{X}| \leq \ell$, ou seja, o conjunto \mathcal{X} contém até ℓ tags. Como em [19], mantemos $\ell = 1$, pois o número de regras cresce exponencialmente, até certo ponto, com o aumento de ℓ . Além disso, com $\ell = 1$ já é possível obter bons resultados.

Além das equações que medem a co-ocorrência de tags, outras métricas podem ser utilizadas para estimar a relevância de uma tag para recomendação, dando maior peso àquelas que têm maior qualidade em termos de poder descritivo e poder discriminativo. Além disso, outros atributos textuais de um objeto podem conter termos relevantes. Tais aspectos são abordados nas Seções 3.2 a 3.4.

3.2 Frequência de termos

Termos muito frequentes como a palavra “vídeo” na coleção de vídeos do Youtube são muito genéricos e pouco discriminativos. Por outro lado, termos muito raros podem ser palavras com erros de ortografia, neologismos e nomes próprios raros, que também não contribuem para tarefas como classificação e recomendação por terem baixa capacidade de generalizar conceitos. Sigurbjornsson *et al.* [19] propõem e aplicam ao problema de recomendação a métrica *Estabilidade*, que dá um peso maior para tags que possuem um valor intermediário de frequência:

$$Est(t, k_s) = \frac{k_s}{k_s + |k_s - \log(f_{t,tag})|} \quad (2)$$

onde $f_{t,tag}$ é o número de objetos de \mathcal{D} em que o termo t aparece como tag e k_s é um parâmetro que representa a “frequência ideal” de um termo e deve ser ajustado à coleção de dados. Diferentemente de [19], aplicamos esta métrica para os termos de todos os atributos textuais considerados.

Por outro lado, pode ser desejável recomendar alguns termos mais raros, visto que eles têm maior capacidade de discriminar objetos (importante aspecto para ordenação dos resultados de uma busca, por exemplo). Esse aspecto é melhor capturado pela métrica Inverso da Frequência no Atributo (IFA) [5, 4], uma adaptação do tradicional IDF utilizado em RI que considera a frequência de um termo em um atributo

textual. Aqui, utilizamos o IFA de um termo no atributo textual tags, definido como:

$$IFA(t, tag) = \log \frac{|\mathcal{D}| + 1}{f_{t,tag} + 1} \quad (3)$$

Vale ressaltar que o termo t pode vir de outros atributos textuais além de tags, embora a frequência considerada seja neste último. O valor 1 somado ao numerador e denominador da Eq. 3 trata dos casos em que o termo não apareceu em \mathcal{D} , apresentando frequência nula, e IFA máximo. Note que, embora isso possa privilegiar termos que aparecem em outros atributos textuais, o peso relativo dessa métrica pode ser ajustado na fórmula final de recomendação pelo próprio processo evolutivo da PG, como veremos mais adiante.

3.3 Poder descritivo

Quando são disponibilizados outros atributos textuais para descrever o conteúdo de um objeto, tais como título e descrição, podemos explorar a métrica heurística *Espalhamento* [5, 4] para estimar o poder descritivo dos termos associados aos atributos de um objeto. O *Espalhamento* $E(t, o)$ de um termo t em um objeto o é dado pelo número de atributos textuais (exceto tags, no contexto de recomendação) de o que contêm t , ou seja:

$$E(t, o) = \sum_{\mathcal{A}_o^i \in \mathcal{A}_o} j, \text{ onde } j = \begin{cases} 1 & \text{se } t \in \mathcal{A}_o^i \\ 0 & \text{caso contrário} \end{cases} \quad (4)$$

A hipótese por traz de $E(t, o)$ é que quanto maior o número de atributos de um objeto o que contêm o termo t , maior a chance de que t seja relacionado ao conteúdo de o . Por exemplo, se o termo “Sting” aparece em todos os atributos de um vídeo do YouTube (tags, título, descrição e comentários), há uma grande probabilidade de que o vídeo seja relacionado a esse cantor. No caso das aplicações que consideramos, aqui o espalhamento máximo de um termo t candidato à recomendação será 3, quando consideramos título, descrição e comentários, já que os termos candidatos obviamente não podem pertencer a \mathcal{I}_o .

3.4 Entropia

Um outro aspecto de qualidade de um termo ou atributo textual útil na prática para recomendação de tags é a sua previsibilidade, ou seja, a facilidade com que um termo pode ser recomendado para um determinado objeto. Heymann *et al.* [8] quantificam essa característica através da Entropia. Entretanto, nosso trabalho é o primeiro a utilizar essa métrica como componente de uma função para estimar a relevância de termos candidatos. A Entropia $H(t, tag)$ de um termo t no campo tag é definida como:

$$H(t, tag) = - \sum_{(\{t\} \rightarrow i) \in \mathcal{R}} \theta(\{t\} \rightarrow i) \log \theta(\{t\} \rightarrow i) \quad (5)$$

Se um termo co-ocorre consistentemente com determinadas tags, ele é mais previsível (dado que ele ocorreu, é relativamente fácil saber quais tags vão ocorrer), e tem menor Entropia. Por outro lado, o fato de um termo ocorrer com igual probabilidade com determinadas tags torna-o mais imprevisível, com maior Entropia. Numa outra interpretação, a Entropia de um termo t mede a concentração da distribuição dos valores de Confiança de todas as regras de associação em \mathcal{R} tendo t como antecedente. Quando um termo

t ocorre em um objeto de teste, mas não está presente no treino (\mathcal{D}), o mesmo recebe um valor de Entropia arbitrariamente grande, visto que o resultado da Eq. 5 nesse caso não é um número real.

4. ESTRATÉGIAS DE RECOMENDAÇÃO DE TAGS

Esta Seção descreve a estratégia de recomendação proposta em [19], que considera a co-ocorrência de tags para recomendação, bem como a extensão proposta considerando métricas de qualidade. Finalmente, uma estratégia de recomendação baseada em programação genética é introduzida.

4.1 Estratégia Baseada em Co-ocorrências

O método base de comparação utilizado neste trabalho é a melhor função proposta em [19], denominada Sum^+ . Esta função, em acréscimo à métrica de co-ocorrência da Eq. 1, pondera os valores de Confiança pela Estabilidade das tags do antecedente e do consequente de cada regra de associação que tem uma das tags de \mathcal{I}_o como antecedente. Seja c o termo candidato a recomendação ao objeto o . A função Sum^+ é definida como:

$$Sum^+(c, o, k_i, k_c, k_r) = \sum_{i \in \mathcal{I}_o} \theta(\{i\} \rightarrow c) \times Est(i, k_i) \times Est(c, k_c) \times Rank(c, i, k_r) \quad (6)$$

onde $\theta(\{i\} \rightarrow c)$ é a Confiança da regra de associação $(\{i\} \rightarrow c) \in \mathcal{R}$. Os parâmetros k_i , k_c e k_r devem ser ajustados para otimizar o resultado das recomendações. A função $Rank(c, o, k_r)$ é utilizada como uma heurística para dar maior peso a tags candidatas que estão no topo das listas de candidatas, com o objetivo de suavizar o decaimento dos valores de co-ocorrência. Ela é definida como $Rank(c, o, k_r) = k_r / (k_r + p(c, o))$, onde $p(c, o)$ é a posição de c entre as tags candidatas ordenadas por Confiança.

Este método foi escolhido como linha de base devido à sua semelhança com o nosso (ambos fazem *ranking* de termos com base em co-ocorrências e métricas de qualidade) e por ser o estado-da-arte em recomendação associativa.

4.2 Estratégia Baseada em Co-ocorrências e Métricas de Qualidade

Propomos uma função que consiste na combinação convexa entre a Equação 6 e o Espalhamento do termo candidato. Novamente, seja c um termo candidato a recomendação ao objeto o . Sum^+E é definida como:

$$Sum^+E(c, o, k_i, k_c, k_r, \alpha) = \alpha Sum^+(c, o, k_i, k_c, k_r) + (1 - \alpha)E(c, o) \quad (7)$$

O parâmetro $0 \leq \alpha \leq 1$ pondera os dois fatores, Sum^+ e E . Quanto maior o valor de α , maior será o peso dado à função Sum^+ , enquanto valores menores de α dão maior peso ao valor de $E(c, o)$ de um termo nos atributos textuais de um objeto. O fator Sum^+ é calculado sobre os termos candidatos gerados a partir de co-ocorrências, enquanto o fator E também é calculado sobre termos de outros atributos textuais.

4.3 Estratégia Baseada em PG

A PG é uma técnica de aprendizado de máquina iterativa que tem obtido sucesso nas mais diversas aplicações por ser independente de domínio e implementar um mecanismo de

busca global. A PG evolui uma população de indivíduos representados por árvores, e criados inicialmente a partir de um conjunto de terminais e funções associados ao problema alvo. A cada geração, os indivíduos são avaliados de acordo com uma métrica de qualidade também relacionada ao problema em questão, e calculada por uma função de *Fitness*. Nesta Seção, definimos a representação de um indivíduo (Seção 4.3.1), a função que avalia a qualidade de uma solução (Seção 4.3.2) e os operadores e parâmetros utilizados na PG (Seção 4.3.3).

4.3.1 Representação

A representação de um indivíduo é uma árvore, composta de terminais (folhas) e não-terminais (nós internos). Cada terminal representa uma constante ou uma variável associada ao problema. Os não-terminais são operadores, tais como adição e multiplicação. A árvore representa uma função qualquer que utilize tais elementos. Escolhemos como não-terminais o conjunto de operadores $\{+, -, \times, /, \log\}$. Os terminais consistem de variáveis e constantes (valores entre 0 e 1 gerados aleatoriamente com distribuição uniforme). As variáveis utilizadas incluem as métricas das equações 1-5, a frequência do termo t em \mathcal{D} ($f_{t,tag}$) e os fatores das funções apresentadas em [19], denominados *Sum* (Eq. 1), Sum_{est} , *Vote* e $Vote_{est}$. $Vote(c, o)$ nada mais é do que o número de regras de associação que têm como consequente o termo candidato c e como antecedente um subconjunto de \mathcal{I}_o , representando quantos “votos” o termo c recebeu como relevante. $Vote_{est}$ é semelhante a *Vote*, mas pondera cada voto pela *Estabilidade* do antecedente. A Tabela 1 mostra todas as métricas utilizadas como variáveis na PG.

Tabela 1: Métricas utilizadas na PG.

Símbolo	Fórmula ou Descrição
$f_{t,tag}$	Frequência do termo t como tag
$IFA(t, tag)$	Eq. 3
$H(t, tag)$	Eq. 5
$E(t, o)$	Eq. 4
$Est(t, tag)$	Eq. 2
$Sum(t, o)$	Eq. 1
$Sum_{est}(t, o, k_i, k_r)$	$\sum_{i \in \mathcal{I}_o} \theta(\{i\} \rightarrow t) \times Est(i, k_i) \times Rank(t, i, k_r)$
$Vote(t, o)$	$\sum_{i \in \mathcal{I}_o} j$, onde $j = \begin{cases} 1, & \text{se } (\{i\} \rightarrow t) \in \mathcal{R} \\ 0, & \text{c.c.} \end{cases}$
$Vote_{est}(t, o, k_i, k_r)$	$\sum_{i \in \mathcal{I}_o} j \times Est(i, k_i) \times Rank(t, i, k_r)$, onde $j = \begin{cases} 1, & \text{se } \{i\} \rightarrow t \in \mathcal{R} \\ 0, & \text{c.c.} \end{cases}$

4.3.2 Fitness

A *Fitness* de um indivíduo neste contexto representa a qualidade das recomendações produzidas pela função de recomendação, estimada em termos da precisão entre as primeiras k tags retornadas. Seja \mathcal{Y}_o o conjunto de tags relevantes para o objeto o , C_o as recomendações para o , ordenadas pela função sendo avaliada, e $C_{k,o}$ os primeiros k elementos de C_o . A precisão é calculada como:

$$P@k(C_o, \mathcal{Y}_o) = \frac{|C_{k,o} \cap \mathcal{Y}_o|}{\min(k, |\mathcal{Y}_o|)} \quad (8)$$

Note que tratamos também os casos em que o gabarito tem tamanho inferior a k , quando o denominador deve ser igual a $|\mathcal{Y}_o|$. O valor da *Fitness* de uma função f é dado pela média de $P@k$ sobre as recomendações geradas por f em uma amostra de objetos passada como entrada.

4.3.3 PG: Funcionamento e Parâmetros

O valor da *Fitness* é utilizado como critério para selecionar os melhores indivíduos, que transmitirão suas características para as próximas gerações através de operações de cruzamento e mutação, entre outras. Utilizamos a **seleção por torneio**, que sorteia, com reposição, k indivíduos da população e em seguida escolhe o melhor (com maior valor de *Fitness*) dentre eles. Enquanto o tamanho da nova população não atinge um determinado número de indivíduos n , dois indivíduos são obtidos pelo método de seleção descrito anteriormente e, com uma probabilidade p_c , o seu “material genético” é trocado no processo de **cruzamento**, no qual escolhe-se aleatoriamente um nó de cada árvore, e as subárvores abaixo de cada um desses nós são trocadas. O papel do cruzamento é combinar soluções boas, rumo às soluções mais promissoras no espaço de busca.

Por fim, é aplicado o operador genético **mutação**, para introduzir soluções novas na população, gerando maior diversidade de soluções. Um indivíduo sofre mutação com probabilidade p_m . Nesse trabalho, a mutação é feita selecionando-se um nó da árvore escolhido aleatoriamente e trocando-o por uma nova subárvore, gerada aleatoriamente, respeitando-se a profundidade máxima permitida.

O processo é repetido iterativamente até obter um valor de *Fitness* dado como meta, ou até atingir um número g de gerações passado como parâmetro ao algoritmo.

5. AVALIAÇÃO EXPERIMENTAL

Nesta Seção apresentamos, primeiramente as bases de dados utilizadas para avaliar os três métodos de recomendação considerados (Seção 5.1). A seguir, descrevemos a metodologia de avaliação aplicada (Seção 5.2), bem como os resultados mais relevantes obtidos (Seção 5.3).

5.1 Bases de Dados

Para avaliar os métodos de recomendação foram coletados os atributos textuais título, tags, descrição e comentários de usuários associados a objetos das aplicações LastFM, YouTube e YahooVideo. A coleta, realizada em outubro de 2008, consistiu de dados de 99161 artistas do LastFM, 180778 vídeos do YouTube e 160228 vídeos do YahooVideo. A estratégia de coleta empregada foi a *snowball sampling*, que parte de um conjunto de objetos tomados como sementes, e segue recursivamente os *links* que apontam para objetos relacionados. Foram selecionados como sementes os artistas associados às tags mais populares do LastFM e os vídeos mais populares do YouTube e do YahooVideo.

Para cada atributo coletado, removemos os afixos das palavras com o algoritmo de Porter [16] para evitar recomendações triviais como plurais e outras pequenas variações de uma mesma palavra. Também foram filtrados *stopwords* (artigos, preposições e vocábulos com pouca semântica), termos muito frequentes (frequência absoluta maior que 10000 na amostra coletada) e termos muito raros (frequência menor que 30), uma vez que tanto termos muito raros quanto termos muito frequentes podem não ser boas recomendações [19].

5.2 Metodologia de Avaliação

Como em [6], utilizamos uma metodologia de avaliação dos métodos de recomendação completamente automática. A idéia é usar parte das tags de um objeto como *gabarito*,

Tabela 2: Subconjuntos de Objetos.

Coleção	Intervalo Menor		Intervalo Médio		Intervalo Maior	
	#tags/ objeto	#objs	#tags/ objeto	#objs	#tags/ objeto	#objs
LastFM	2-6	29622	7-16	30215	17-152	31492
Yahoo	2-6	42847	7-11	60280	12-52	44009
YouTube	2-5	56721	6-9	53284	10-74	59285

isto é, como as tags relevantes para aquele objeto. Estas tags são desconsideradas pelas funções de recomendação no cálculo das métricas apresentadas na Seção 3. Esta metodologia foi adotada pois a avaliação manual da relevância das tags recomendadas é um processo bastante caro, além de estar sujeito à subjetividade dos julgamentos humanos. Note que, na metodologia aplicada, os resultados correspondem a limites inferiores para a relevância das recomendações, dado que algumas das tags recomendadas, apesar de não pertencerem ao gabarito (sendo consideradas irrelevantes) podem de fato ser relevantes para o dado objeto.

A relevância das recomendações de cada método é avaliada pela métrica $P@k$, com $k=5$, definida na Seção 4.3.2. Assim, $P@k$ é utilizada para avaliar tanto os indivíduos em cada geração do processo evolucionário da PG, quanto a recomendação produzida a partir da aplicação de cada função adotada, seja ela Sum^+ , Sum^+E ou a função final gerada pela PG. Dividimos igualmente as tags associadas a cada objeto de teste (ou de validação) $o = \langle \mathcal{I}_o, \mathcal{A}_o, \mathcal{V}_o \rangle$: metade das tags (selecionadas aleatoriamente) foi incluída em \mathcal{I}_o e a outra metade em \mathcal{V}_o , como gabarito. Como conjunto de atributos textuais \mathcal{A}_o utilizamos o título e a descrição dos objetos. Optamos por não incluir comentários em \mathcal{A}_o porque, conforme mostrado em [5, 4]: (1) comentários apresentam grande quantidade de ruído (palavras inexistentes, termos vagos, etc) que não contribuem para descrever o conteúdo; (2) eles estão ausentes em uma parcela significativa dos objetos nas aplicações estudadas; (3) quando presentes, eles contêm grandes quantidades de termos, o que diminui a eficiência de um recomendador, visto que ele terá que considerar uma quantidade muito maior de candidatos. Mais ainda, verificamos, em experimentos iniciais, que a retirada dos comentários de \mathcal{A}_o não afeta significativamente a eficácia dos métodos de recomendação estudados, confirmando sua pequena contribuição para recomendação.

Para analisar a eficácia dos métodos para diferentes quantidades de tags fornecidas como entrada, dividimos cada coleção em três subconjuntos definidos a partir do número de tags disponíveis no objeto. O intervalo do número de tags por objeto que define cada subconjunto foi selecionado de forma a manter aproximadamente o mesmo número de objetos em cada subconjunto, conforme mostra a Tabela 2.

Em seguida, amostramos 20000 objetos selecionados aleatoriamente de cada um dos subconjuntos. Cada uma dessas amostras foi dividida em 5 partes de 4000 objetos, sobre as quais aplicamos validação cruzada para avaliar a qualidade das recomendações. Três partes são utilizadas como conjunto de treino (conjunto \mathcal{D}), que por sua vez é utilizado para geração das regras de associação, cálculo da frequência de tags e entropia, conforme descrito nas seções 3.1, 3.2 e 3.4. Uma parte é utilizada como conjunto de validação, usado para avaliar a qualidade das funções de recomendação geradas durante o processo evolucionário da PG e também para realizar ajuste de parâmetros. A última parte é usada para teste e comparação das estratégias. As 5 partes de cada amostra são rotacionadas 5 vezes (5 folds).

5.3 Resultados

A seguir apresentamos os resultados mais relevantes da comparação das funções Sum^+ , Sum^+E e da melhor função gerada pela PG ao final do processo evolucionário, como recomendadores de tags. Todos os resultados são valores médios de 5 conjuntos de validação (no ajuste de parâmetros feito na Seção 5.3.1) ou de 5 conjuntos de teste (resultados apresentados na Seção 5.3.2). Intervalos de confiança de 95% foram calculados, indicando um erro máximo de apenas 2% da média.

5.3.1 Impacto dos Parâmetros

Primeiramente, realizamos uma bateria de experimentos para determinar os melhores valores dos parâmetros utilizados nas funções Sum^+ e Sum^+E , bem como os limiares de suporte mínimo (σ_{min}) e confiança mínima (θ_{min}). Note que estes dois últimos têm impacto direto no número de regras candidatas geradas a partir da co-ocorrência de termos e, portanto, na eficiência (tempo de processamento) das funções de recomendação. Quanto menor for σ_{min} (ou θ_{min}), maior o número de regras geradas e maior o tempo de processamento. Logo, buscaremos valores que atinjam um bom compromisso entre eficiência e precisão da recomendação. Começamos fixando $k_r=1$, $\sigma_{min}=1$ e $\theta_{min}=0$, e variando k_i e k_c (Eq. 6) no intervalo entre 5 e 50, conforme [19]. Os resultados (omitidos) mostram pouco impacto desses parâmetros na precisão da recomendação, sendo $k_i=5$ e $k_c=5$, a melhor escolha na maioria dos testes. Fixados estes valores, nós avaliamos o impacto de variar k_r de 1 a 10, concluindo que $k_r=1$ era de fato o melhor valor.

Em seguida, avaliamos o impacto dos parâmetros σ_{min} e θ_{min} na precisão da recomendação obtida com a função Sum^+ . Os resultados são mostrados na Tabela 3 para as três aplicações e subconjuntos de objetos. Na maioria dos casos, a precisão cai quando aumentamos qualquer um dos limiares, principalmente se menos tags são fornecidas como entrada, pois o número de candidatas geradas já é pequeno. Entretanto, quando o número de tags fornecidas é maior, a escolha desses limiares pode ser mais agressiva sem prejudicar significativamente a precisão. Note que, no LastFM, para o subconjunto definido pelo maior intervalo, o melhor resultado obtido foi para o maior limiar de Confiança testado. Isto pode ser devido a um grande número de regras geradas para pequenos valores de σ_{min} e θ_{min} , que acabam por confundir o recomendador. De fato, verificamos que nesses casos há uma tendência de recomendar tags muito gerais, como “music” e “listen”, pois há uma maior quantidade de regras envolvendo tags desta natureza, embora essas regras geralmente apresentem baixa Confiança.

Assim, para obter um bom compromisso entre eficiência e precisão, nós escolhemos, para cada subconjunto de objetos, os maiores valores de σ_{min} e θ_{min} para os quais a perda de precisão, em relação aos resultados para limiares iguais a 0, foi menor que 3%. A exceção foi o subconjunto do YouTube com menor intervalo, para o qual uma escolha mais agressiva, com perda de 6,2% levou a um melhor compromisso de eficiência, particularmente para a fase de treino da solução baseada em PG. Os valores escolhidos são mostrados em negrito na Tabela 3.

Nós então aplicamos os valores escolhidos para σ_{min} e θ_{min} na função Sum^+E , variando o parâmetro α . Os resultados, mostrados na Tabela 4, indicam que a melhor configuração tipicamente varia em torno de $\alpha=0,8$ e $\alpha=0,9$. Note que quando $\alpha=1$, Sum^+E se reduz a Sum^+ , exceto

Tabela 3: P@5 de Sum^+ . Em negrito, melhores resultados (melhor compromisso entre eficiência e precisão).

σ_{min}	θ_{min}	LastFM			YahooVideo			YouTube		
		2-6	7-16	17-152	2-6	7-11	12-52	2-5	6-9	10-74
1	0,00	0,355	0,290	0,358	0,430	0,398	0,495	0,351	0,283	0,322
1	0,01	0,355	0,290	0,358	0,430	0,398	0,495	0,351	0,283	0,322
1	0,10	0,348	0,291	0,357	0,424	0,392	0,492	0,345	0,283	0,320
1	0,20	0,294	0,300	0,355	0,405	0,390	0,485	0,315	0,289	0,325
1	0,30	0,223	0,296	0,374	0,371	0,391	0,482	0,271	0,274	0,334
1	0,40	0,181	0,270	0,458	0,333	0,385	0,485	0,237	0,247	0,334
1	0,50	0,153	0,244	0,521	0,298	0,364	0,488	0,215	0,219	0,324
2	0,00	0,349	0,291	0,357	0,419	0,396	0,495	0,329	0,283	0,321
2	0,01	0,349	0,291	0,358	0,419	0,396	0,495	0,329	0,283	0,321
2	0,10	0,339	0,292	0,357	0,413	0,391	0,492	0,320	0,283	0,320
2	0,20	0,280	0,301	0,355	0,390	0,389	0,484	0,284	0,286	0,325
2	0,30	0,208	0,296	0,374	0,353	0,389	0,482	0,241	0,267	0,334
2	0,40	0,167	0,268	0,458	0,316	0,382	0,485	0,213	0,239	0,333
2	0,50	0,139	0,241	0,521	0,279	0,359	0,488	0,190	0,208	0,322

pele fato de que continuam sendo consideradas as palavras de A_o como candidatas, embora o valor de relevância estimado para elas seja nulo, deixando-as no final da lista de candidatas. De fato, comparando os valores para $\alpha=1$ com os valores em negrito da Tabela 3, nota-se que as diferenças são marginais na maioria dos casos. A exceção é para o primeiro subconjunto do YouTube, no qual, simplesmente considerar as palavras de A_o já contribui para um aumento de 17,6% na precisão.

Quanto à solução baseada em PG, a escolha dos valores de parâmetros foi feita como segue. Realizamos experimentos preliminares variando o tamanho da população em $n=50$ e $n=100$. Uma população maior leva a resultados significativamente melhores, por permitir uma maior cobertura do espaço de busca de soluções. Por isto, fixamos $n=100$. Para este tamanho de população, o algoritmo converge em menos de 200 gerações, valor fixado para g . Foi variado o número de indivíduos no torneio em $k=2$ e $k=10$, não havendo diferença significativa no resultado de $P@5$. Logo, optamos por utilizar $k=2$, por exigir menos cálculos de *Fitness*, operação computacionalmente mais cara da PG. Fixamos $p_c=0,6$ e $p_m=0,1$, valores comumente utilizados na literatura [2].

Como o custo de computar a *Fitness* durante o processo evolucionário sobre os 4000 objetos do conjunto de validação pode ser proibitivo, estudamos a possibilidade de calcular a precisão sobre apenas uma amostra de tamanho a desses objetos. Consideramos $a=100$ e $a=500$. Também consideramos uma amostra variável dos 4000 objetos que muda a cada 5 gerações. Esta escolha pode ser mais adequada em relação a uma amostra fixa, visto que evita *overfitting*, ou seja, que as funções geradas pela PG sejam especializadas nos conjuntos de validação e treino considerados, não sendo suficientemente gerais para lidar com novos objetos. Os resultados (omitidos) da comparação entre a eficácia da PG para ambos os tamanhos de amostra e estratégia de amostragem variável mostraram pouca diferença entre eles. Logo, escolhemos a amostragem variável com $a=100$, visto que esta configuração leva a um tempo de treinamento 5 vezes menor que quando $a=500$.

5.3.2 Comparando os Métodos de Recomendação

Definidos os melhores parâmetros da PG, o processo evolucionário foi repetido 5 vezes (5 sementes) utilizando, em cada uma delas, a validação cruzada descrita na Seção 5.2. Cada função gerada ao final do processo foi aplicada em cada um dos conjuntos de teste, produzindo 25 resultados, cuja média é mostrada na Tabela 5, junto a intervalos de confiança de 95%. Também mostramos os resultados das funções

Sum^+ e Sum^+E nesses mesmos conjuntos de teste.

Comparando os resultados de Sum^+ e Sum^+E , nota-se que a inclusão da métrica $E(t,o)$ melhorou significativamente os resultados, particularmente no YahooVideo e YouTube, para os quais a precisão aumentou em até 72-73%. No LastFM, a contribuição dessa métrica é apenas marginal. Isso se deve à maior diversidade de conteúdo existente nos diferentes atributos textuais do LastFM, se comparado ao YouTube e YahooVideo, implicando que título, descrição e tags têm muitos menos termos em comuns [5, 4]. Isto leva a uma distribuição de valores de Espalhamento dos termos de um objeto muito mais concentrada em pequenos valores (0 e 1), dificultando a distinção do *rank* de muitos termos.

Considerando a melhor função gerada pela PG em cada subconjunto de cada aplicação, observa-se que as métricas que mais aparecem são o Espalhamento E e pelo menos uma das métricas de co-ocorrência (Sum , Sum_{est} , $Vote$, $Vote_{est}$). Isso era esperado visto que tais métricas são os elementos-chave para estimar a relevância de um termo. As métricas de qualidade *IFA* e *Estabilidade* também foram incluídas em várias das melhores funções, indicando que essas métricas são úteis ao recomendador. A exceção foi o subconjunto do LastFM com menor número de tags por objeto, para o qual a melhor função gerada diminui com o aumento do Espalhamento. Conforme discutido, o Espalhamento não é muito útil quando há grande diversidade de conteúdo entre os atributos textuais, como é o caso do LastFM.

Apesar do uso destas várias métricas, a Tabela 5 mostra que a solução baseada em PG apresenta ganhos de precisão modestos (até 4%) quando comparada à função Sum^+E . Isso indica que esta função, mais simples que algumas das funções geradas pela PG, é bastante eficaz para recomendar tags, e ressalta mais uma vez o benefício de se explorar termos de outros atributos textuais através de uma métrica que expressa a qualidade destes termos, particularmente o seu poder descritivo. De qualquer maneira, vale ressaltar que, embora os ganhos sejam pequenos, a aplicação da função aprendida com a PG após o treinamento (feito *offline*, periodicamente) não apresenta custos adicionais em relação à aplicação de Sum^+E . Além disto, os resultados mostrados devem ser interpretados como limites inferiores, devido à metodologia de avaliação. Mais ainda, eles demonstram a viabilidade de aplicação do arcabouço de PG, o qual pode ser facilmente estendido para incorporação de novas métricas de qualidade, para o problema de recomendação de tags.

6. CONCLUSÕES E TRABALHOS FUTUROS

Neste artigo, nós propusemos e avaliamos dois novos mé-

Tabela 4: P@5 de Sum^+E . Em negrito, melhores resultados.

α	LastFM			YahooVideo			YouTube		
	2-6	7-16	17-152	2-6	7-11	12-52	2-5	6-9	10-74
0,0	0,070	0,054	0,078	0,508	0,400	0,383	0,463	0,316	0,242
0,1	0,226	0,172	0,232	0,640	0,516	0,509	0,526	0,394	0,325
0,2	0,226	0,172	0,235	0,640	0,516	0,510	0,526	0,394	0,325
0,3	0,226	0,173	0,252	0,640	0,518	0,523	0,526	0,394	0,332
0,4	0,227	0,178	0,293	0,646	0,529	0,550	0,527	0,397	0,346
0,5	0,231	0,196	0,361	0,664	0,548	0,548	0,531	0,403	0,368
0,6	0,245	0,226	0,453	0,692	0,572	0,618	0,541	0,416	0,394
0,7	0,275	0,271	0,530	0,715	0,592	0,639	0,553	0,434	0,422
0,8	0,310	0,310	0,556	0,726	0,593	0,624	0,568	0,453	0,433
0,9	0,350	0,328	0,557	0,717	0,535	0,553	0,574	0,440	0,406
1,0	0,357	0,321	0,553	0,436	0,408	0,509	0,387	0,302	0,354

Tabela 5: P@5 e Intervalos de Confiança de 95%. Em negrito, melhores resultados.

Coleção	#tags/objeto	Sum^+	Sum^+E	PG
LastFM	2-6	0,355 ± ,005	,357 ± ,005	,370 ± ,004
	7-16	0,321 ± ,001	,328 ± ,001	,337 ± ,004
	17-152	0,553 ± ,003	,558 ± ,002	,573 ± ,004
Yahoo	2-6	0,422 ± ,006	,727 ± ,003	,721 ± ,008
	7-11	0,407 ± ,001	,593 ± ,004	,613 ± ,006
	12-52	0,509 ± ,005	,647 ± ,005	,673 ± ,004
YouTube	2-5	0,332 ± ,005	,575 ± ,004	,575 ± ,004
	6-9	0,301 ± ,005	,453 ± ,003	,463 ± ,003
	10-74	0,349 ± ,003	,433 ± ,002	,437 ± ,008

todos de recomendação de tags que incluem, além de estimativas de co-ocorrência de tags, métricas de qualidade da informação aplicadas a termos extraídos de outros atributos textuais. Nossos experimentos mostraram que, em comparação com explorar apenas co-ocorrência de tags, a inclusão da métrica Espalhamento, que captura o poder descritivo de um termo, aplicada a termos extraídos do título e da descrição do objeto, pode melhorar em até 73% a precisão da recomendação. Apesar de modestos, os resultados da PG indicam que ela é um arcabouço promissor para o problema de recomendação de tags, sendo flexível e podendo ser entendido com a inclusão de novas métricas de qualidade.

Ressaltamos que os resultados obtidos são limites inferiores. No futuro, pretendemos avaliar os métodos manualmente e explorar o uso de outras métricas de qualidade, particularmente junto com a PG, bem como dos relacionamentos entre usuários para recomendações personalizadas.

7. REFERENCES

- [1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *VLDB*, 1994.
- [2] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone. *Genetic Programming – An Introduction; On the Automatic Evolution of Computer Programs and its Applications*. Morgan Kaufmann, 1998.
- [3] L. Chen, P. Wright, and W. Nejdl. Improving music genre classification using collaborative tagging data. In *WSDM*, 2009.
- [4] F. Figueiredo, F. Belem, H. Pinto, J. Almeida, M. Gonçalves, D. Fernandes, E. Moura, and M. Cristo. Caracterizando o uso e a qualidade de atributos textuais em aplicações da web 2.0. In *WebMedia*, 2009.
- [5] F. Figueiredo, F. Belem, H. Pinto, J. Almeida, M. Gonçalves, D. Fernandes, E. Moura, and M. Cristo. Evidence of quality of textual features on the web 2.0. In *CIKM*, 2009.
- [6] N. Garg and I. Weber. Personalized, interactive tag recommendation for flickr. In *RecSys*, 2008.
- [7] Z. Guan, J. Bu, Q. Mei, C. Chen, and C. Wang. Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. In *SIGIR*, 2009.
- [8] P. Heymann, D. Ramage, and H. Garcia-Molina. Social tag prediction. In *SIGIR*, 2008.
- [9] F. Khron Jr. and S. C. Cazella. Framework para recomendação de novos relacionamentos em uma rede social a partir do uso de técnicas de folksonomia. In *WebMedia*, 2009.
- [10] I. Konstas, V. Stathopoulos, and J. Jose. On social networks and collaborative recommendation. In *SIGIR*, 2009.
- [11] R. Krestel, P. Fankhauser, and W. Nejdl. Latent dirichlet allocation for tag recommendation. In *RecSys*, 2009.
- [12] X. Li, L. Guo, and Y.E. Zhao. Tag-based Social Interest Discovery. In *Proc. WWW*, 2008.
- [13] M. Lipczak. Tag recommendation for folksonomies oriented towards individual users. In *ECML/PKDD*, 2008.
- [14] G. R. Lopes, M. A. M. Souto, L. K. Wives, and J. P. M. de Oliveira. A personalized recommender system for digital libraries. In *WebMedia*, 2008.
- [15] H. Ma, I. King, and M. Lyu. Learning to recommend with social trust ensemble. In *SIGIR*, 2009.
- [16] M. F. Porter. An algorithm for suffix stripping. In *Readings in information retrieval*, pages 313–316, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.
- [17] D. Ramage, P. Heymann, C.D. Manning, and H. Garcia-Molina. Clustering the tagged web. In *WSDM*, 2009.
- [18] S. Siersdorfer, J. San Pedro, and M. Sanderson. Automatic video tagging using content redundancy. In *SIGIR*, 2009.
- [19] B Sigurbjornsson and R. van Zwol. Flickr Tag Recommendation Based on Collective Knowledge. In *WWW*, 2008.
- [20] Y. Song, Z. Zhuang, H. Li, Q. Zhao, J. Li, W.C. Lee, and C.L. Giles. Real-Time Automatic Tag Recommendation. In *SIGIR*, 2008.
- [21] L. Wu, L. Yang, N. Yu, and X. Hua. Learning to tag. In *WWW*, 2009.
- [22] Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the semantic web: Collaborative tag suggestions. In *WWW*, 2006.