CAIO RAMOS CASIMIRO

Aspectos temporais na recomendação de conteúdo em microblogs

São Paulo

2015

CAIO RAMOS CASIMIRO

Aspectos temporais na recomendação de conteúdo em microblogs

Versão original

Dissertação apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação.

Área de Concentração: Sistemas de Informação

Orientador: Prof. Dr. Ivandre Paraboni

São Paulo
2015

CATALOGAÇÃO-NA-PUBLICAÇÃO
(Universidade de São Paulo. Escola de Artes, Ciências e Humanidades. Biblioteca)

Casimiro, Caio Ramos
Aspectos temporais na recomendação de conteúdo em microblogs / Caio Ramos Casimiro ; orientador, Ivandré Paraboni. – São Paulo, 2015
71 f. : il.

Dissertação (Mestrado em Ciências) - Programa de Pós-Graduação em Sistemas de Informação, Escola de Artes, Ciências e Humanidades, Universidade de São Paulo
Versão original

1. Linguagem natural. 2. Inteligência artificial. 3. Blogs. I. Paraboni, Ivandré, orient. II. Título

CDD 22.ed. – 006.35

Dissertação de autoria de Caio Ramos Casimiro, sob o título **"Aspectos temporais na recomendação de conteúdo em microblogs"**, apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo, para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação, na área de concentração Sistemas de Informação, aprovada em ___ de _____ de _____ pela comissão julgadora constituída pelos doutores:

Prof. Dr. _____
Presidente
Instituição: _____

Prof. Dr. _____
Instituição: _____

Prof. Dr. _____
Instituição: _____

Dedico este trabalho ao meu filho Leonardo.

Agradecimentos

Agradeço a Deus pela oportunidade e capacidade para realizar este trabalho.

Agradeço aos meus familiares por todo apoio que recebi neste período.

Agradeço ao professor e amigo Ivandre Paraboni por toda dedicação, paciência e amizade. Em todos momentos críticos deste período, sua orientação e conselhos fizeram a diferença.

Agradeço também a todos amigos que colaboraram de alguma maneira para a conclusão deste trabalho.

"Imaginação é mais importante que conhecimento."

(Albert Einstein)

Resumo

CASIMIRO, Caio Ramos. **Aspectos temporais na recomendação de conteúdo em microblogs.** 2015. 71 f. Dissertação (Mestrado em Ciências) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2015.

Este documento apresenta um estudo que avalia o uso de informação temporal na tarefa de recomendação de *tweets* no *twitter*. Foram explorados dois aspectos temporais: a vida útil de tópico de informação e a sua versão personalizada para cada usuário. A aplicação destes aspectos temporais foi avaliada utilizando-se três sistemas de recomendação implementados. Também avaliamos dois modelos de tópicos utilizados para representar *tweets*: o modelo *bag of words* e um modelo de tópicos latentes extraídos por LDA (*Latent Dirichlet Allocation*). Além disso, avaliamos o uso de máquinas de vetor de suporte para estimar o perfil de interesses de usuário, comparando esta abordagem com uma outra mais simples. Os experimentos foram executados utilizando-se um conjunto de dados com 414 milhões de *tweets* publicados por 321 mil usuários. Os resultados apresentados demonstram que o uso de vida útil de tópico na tarefa de recomendação melhora a qualidade das recomendações, e o uso da versão personalizada desta informação melhorou ainda mais a qualidade destas.

Palavras-chaves: Recomendação de conteúdo. Aspectos temporais. Microblogs. Twitter.

Abstract

CASIMIRO, Caio Ramos. **Temporal aspects on content recommendation in microblogs.**. 2015. 71 p. Dissertation (Master of Science) – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, 2015.

This document presents a study that evaluates the use of temporal information in the task of recommending tweets on Twitter. Two temporal aspects have been analysed: the lifespan of information topic and its personalized version for each user. The application of such temporal aspects has been evaluated using three recommendation systems implemented in this work. We also evaluated two topic models considered to describe tweets: a bag of words model and a model of latent topics extracted using LDA (Latent Dirichlet Allocation). Furthermore, we evaluated the use of SVM (Support Vector Machines) to estimate the user profile, comparing this approach with a simpler one. The experiments have been executed using a dataset with 414 millions of tweets published by 321 thousands of users. The results show that the use of topic lifespan information increases the quality of recommendation, and the personalized version of this information increases the quality even more.

Keywords: Content recommendation. Temporal aspects. Microblogs. Twitter.

Lista de figuras

Lista de tabelas

Sumário

[1] De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

1 Introdução

Em anos recentes, muitas pessoas aderiram ao uso de redes sociais virtuais para diferentes fins. Dentre as várias redes existentes, destacam-se Twitter[1] e Facebook[2], que somam milhões de usuários. Até junho de 2012, o Facebook acumulou aproximadamente 955 milhões de usuários ativos por mês (FACEBOOK, 2012). Já o Twitter, em março de 2012, contabilizou mais de 140 milhões de usuários ativos e uma média de 340 milhões de *tweets* (publicações) por dia (TWITTER, 2012).

Em geral, essas redes sociais apresentam publicações ao usuário sob a forma de uma *timeline*, i.e., uma lista de publicações ordenada de acordo com a data e hora de publicação, de modo que as publicações mais recentes ocupem o topo dessa lista. Apesar da conveniência em agregar diferentes fontes de informação em um único ponto, a grande quantidade de informação que os usuários dessas redes recebem faz com que seja difícil ler o que realmente importa ou interessa. Para resolver esse problema, tanto academia quanto indústria têm pesquisado e implementado sistemas de recomendação de documentos nessas redes (PUNIYANI et al., 2010; CHEN et al., 2010; DUAN et al., 2010; HANNON; BENNETT; SMYTH, 2010; CELEBI; USKUDARLI, 2012; KIM et al., 2014; PHELAN et al., 2011; YI et al., 2014; PENNACCHIOTTI et al., 2012; OTSUKA; WALLACE; CHIU, 2014; ARMENTANO; GODOY; AMANDI, 2013; KYWE et al., 2012; WU et al., 2015; YU, 2012; HONG; DOUMITH; DAVISON, 2013; LIANG et al., 2012).

Sistemas de recomendação são uma sub-classe de sistemas de filtragem de informação (RICCI et al., 2010). Esses sistemas estão presentes em diferentes domínios da web como o de compras (Amazon[3]), entretenimento (Netflix[4]) e até em sistemas de busca fazendo recomendação de propagandas (Google[5]). Para o caso do Twitter, o objetivo de um sistema de recomendação de conteúdo é tipicamente reordenar a *timeline* do usuário, de modo a exibir no topo da lista de publicações aquelas mais interessantes ao usuário. A entrada para esse sistema seria então um conjunto de publicações utilizadas para aprender os interesses do usuário e um conjunto de publicações a serem ordenadas de acordo com os

[1] www.twitter.com
[2] www.facebook.com
[3] wwww.amazon.com
[4] www.netflix.com
[5] www.google.com

interesses aprendidos. A figura 1 ilustra um trecho da *timeline* de um usuário contendo um *retweet* na primeira posição.

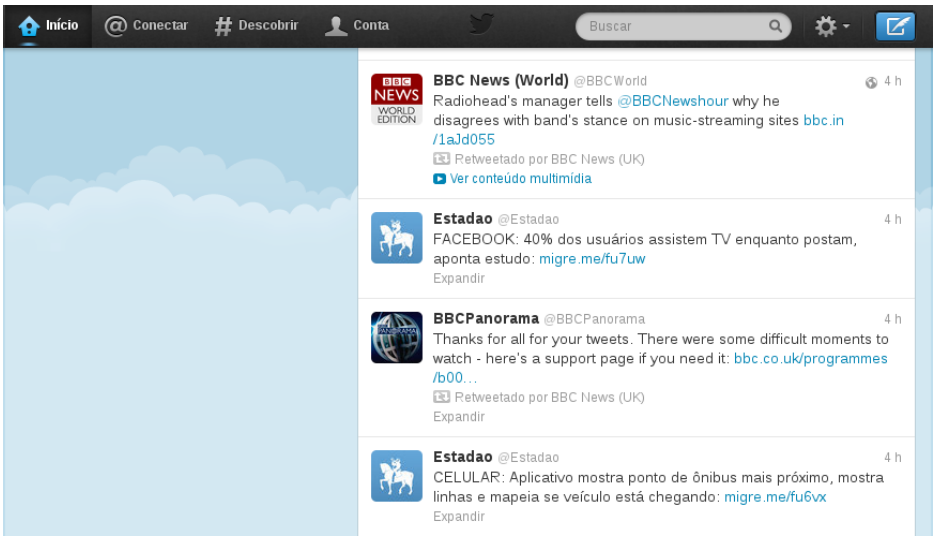

Figura 1 – Trecho da *timeline* de um usuário contendo 3 *tweets* e um *retweet*

Fonte: Twitter, 2015

Apesar de sistemas de recomendação já terem sido aplicados a domínios similares ao Twitter, como o domínio de notícias (RICCI et al., 2010), a recomendação de documentos em redes sociais é um problema com limitações e possibilidades distintas. Para o caso do Twitter, por exemplo, as publicações são limitadas a 140 caracteres, o que motiva o uso de abreviações e desmotiva o uso culto da língua.

Além do problema relacionado ao uso da língua, o Twitter agrega fontes de informação de natureza bastante distinta. Há usuários que compartilham notícias de esportes, há outros que compartilham notícias políticas etc. Ou seja, o Twitter reúne publicações de diferentes tópicos de informação.

Um aspecto fundamental do uso de redes sociais, e que constitui o tema central deste trabalho, é a observação de que tópicos diferentes possuem *tempos de vida útil* diferentes. Por exemplo, parece razoável que uma notícia sobre política permaneça interessante por dias, enquanto uma notícia sobre a situação do trânsito permaneça interessante por poucas horas.

Além disso, é razoável supor que usuários distintos apresentem *padrões distintos de consumo* de informação. Ou seja, além de determinados tópicos terem tempos de vida útil distintos, é possível que estes tempos sejam específicos para cada usuário. Por exemplo,

há usuários que consomem apenas publicações mais recentes, enquanto outros consomem itens mais antigos.

Com base nestas observações, este trabalho explora dois aspectos temporais da recomendação de conteúdo: a informação de vida útil de tópico de informação e sua forma customizada para cada usuário. De forma mais específica, serão investigadas as hipóteses de que considerar a informação de vida útil de tópico melhora a qualidade da recomendação e a hipótese de que usar a forma personalizada desta informação melhora ainda mais a qualidade da recomendação.

1.1 Objetivo Geral

O objetivo da presente pesquisa é explorar o impacto de fatores temporais na tarefa de recomendação de conteúdo no Twitter.

1.2 Objetivos Específicos

Os objetivos específicos desse trabalho são:

- Construir um conjunto de dados anonimizado, extraídos da plataforma Twitter, contendo as informações necessárias para treinar e avaliar um sistema de recomendação de documentos.
- Propor e implementar modelos computacionais que explorem aspectos temporais na tarefa de recomendação de conteúdo no Twitter.
- Verificar, por meio de um experimento controlado, como os aspectos temporais analisados impactam a qualidade da recomendação de conteúdo no Twitter.

O trabalho apresenta três sistemas de recomendação e utiliza dados de 414 milhões de *tweets* e de 321 mil usuários. Tendo em vista os objetivos acima expostos, entretanto, cabe ressaltar que o foco do presente estudo é o uso de aspectos temporais na tarefa de recomendação, sem o objetivo de desenvolver um novo sistema completo que seja de desempenho superior a sistemas existentes.

Esta dissertação está organizada da seguinte forma. O capítulo 2 apresenta conceitos fundamentais para esta pesquisa. O capítulo 3 apresenta trabalhos relacionados. No capítulo 4 apresentamos um estudo preliminar conduzido para obter conhecimento sobre o problema

estudado. O capítulo 5 contém o tema principal desta pesquisa – recomendação de conteúdo com base em informação temporal. O capítulo 6 descreve as hipóteses desta pesquisa, o experimento conduzido para testá-las e apresenta os resultados obtidos. Finalmente, o texto é concluído no capítulo 7 onde sumarizamos o trabalho realizado e apontamos possibilidades para trabalho futuro.

2 Conceitos básicos

Neste capítulo, revisamos uma série de conceitos fundamentais para a presente pesquisa. São discutidos conceitos de classificação de documento (seção 2.1), sistemas de recomendação (seção 2.3) e *Latent Dirichlet Allocation* (seção 2.2).

2.1 Classificação de documentos

O problema de compartilhamento de notícias em redes sociais envolve a classificação de notícias com base no perfil de leitores. Ou seja, para um dado leitor, é preciso classificar uma notícia como sendo recomendável ou não. Com base nisso, nesta seção abordaremos alguns conceitos fundamentais de classificação de documentos.

A atividade de classificar documentos não é algo exatamente novo. Um bibliotecário que organiza os livros de uma biblioteca está classificando documentos. Com o advento da web, a quantidade excessivamente grande de informação gerou oportunidades para a construção de classificadores de documentos, uma vez que a classificação manual é claramente inviável.

Sistemas classificadores podem resolver uma série de problemas: detecção de spam - um classificador pode ser treinado para identificar e-mails indesejados a partir de características do texto; ou, de forma similar, um classificador pode automaticamente separar e-mails em pastas de acordo com seus conteúdos; detecção de sentimento – classificadores podem ser utilizados para detectar sentimentos em sentenças, frases e até mesmo documentos.

Dada essa variedade de aplicações, é importante estabelecer uma definição formal do problema de classificação de documentos. De acordo com (MANNING; RAGHAVAN; SCHUTZE, 2008), um classificador de documentos é um modelo que tem por finalidade mapear documentos para suas respectivas classes:

$$\gamma : X \rightarrow C \tag{1}$$

Onde γ é um classificador e X é um conjunto de documento. Em outras palavras, dada uma a descrição d de um documento e um conjunto fixo de classes $C = \{c_1, c_2, \ldots, c_n\}$ o modelo classificador deverá, com base em experiência adquirida em um processo de

treinamento, atribuir ao documento a classe mais provável levando em consideração a descrição deste.

É importante destacar o processo de treinamento, que permite que um classificador seja capaz de atribuir a classe mais provável de um documento não classificado. Tipicamente, o processo de aprendizado se dá por meio da apresentação de documentos rotulados (documentos com a classe previamente atribuída) ao modelo e posterior teste de desempenho do modelo. Dessa forma, o modelo deverá extrair do conjunto de documentos rotulados, denominado conjunto de treinamento, os padrões ou evidências das classes desses documentos.

O aprendizado acumulado durante a fase de treinamento deverá ser validado posteriormente em um conjunto de documentos diferente do conjunto de teste. Isso ajuda a garantir que o modelo é capaz de generalizar o conhecimento adquirido durante o processo de treinamento e não está limitado a ter bom desempenho apenas com documentos muito parecidos com os contidos no conjunto de treinamento. Esse problema é conhecido como *overfitting*(MANNING; RAGHAVAN; SCHUTZE, 2008).

O treinamento descrito acima é conhecido como aprendizado supervisionado, pois o modelo recebe, durante o período de treinamento, os dados classificados por um humano supervisor.

2.1.1 Classificador Naïve Bayes

Dentre os vários classificadores existentes, vamos detalhar o classificador Naïve Bayes. O classificador Naïve Bayes é um importante classificador conhecido pela sua simplicidade e eficiência. Esse classificador se baseia no teorema de Bayes (MANNING; RAGHAVAN; SCHUTZE, 2008) que, de uma forma geral, modela o relacionamento entre as probabilidades de dois eventos distintos $P(A)$, $P(B)$ e suas probabilidades condicionais: $P(A \mid B)$ e $P(B \mid A)$. Esse relacionamento é expresso usando a seguinte equação:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{2}$$

Em termos gerais, o teorema de Bayes mostra como a evidência do evento B altera a probabilidade *a priori* do evento A. Essa probabilidade alterada, ou atualizada, é

denominada probabilidade *a posteriori* de A (BOLSTAD, 2004). Esse teorema é útil para calcular a probabilidade de um documento pertencer a uma classe dada a sua descrição:

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)} \tag{3}$$

Nesta equação, c representa uma classe e d representa a descrição de um documento. Usualmente, a descrição de um documento é definida pelo seu conjunto de *tokens*(palavras). Assim, o documento fictício "chinese chinese tokio japan" possui uma descrição formada pelo conjunto de tokens d = {chinese, chinese, tokio, japan}. Dependendo das características do modelo, pode ser feito algum tipo de pré-processamento nos conjunto de palavras, por exemplo, evitando repetições ou removendo palavras que não servem como evidência para classe alguma (e.g., a conjunção 'e'). Com base nessa descrição, temos o seguinte modelo (MANNING; RAGHAVAN; SCHUTZE, 2008):

$$P(c \mid d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k \mid c) \tag{4}$$

onde $P(t_k \mid c)$ é a probabilidade de um token ocorrer em documentos de uma determinada classe c. Nesse ponto deve-se ressaltar uma característica importante de classificadores bayesianos - a suposição de independência da ocorrência de palavras dada uma classe. Ou seja, a ocorrência de um termo em um documento de uma classe não influencia na probabilidade de ocorrer um segundo termo nesse documento. Essa suposição é claramente errada. Suponha um documento cuja classe seja "notícia sobre a China" a ocorrência do termo 'hong' certamente altera a probabilidade de ocorrer o termo 'kong'. Contudo, essa suposição simplifica muito os cálculos, e o classificador bayesiano tende a ter um desempenho surpreendentemente bom a despeito desta simplificação.

Na equação 4, o produto das probabilidades não está sendo normalizada pela probabilidade do documento $P(d)$, por isso foi utilizado o símbolo de proporcionalidade e não de igualdade. Essa simplificação não altera o resultado do classificador uma vez que essa probabilidade é igual para todas as classes, conforme discutido a seguir.

Ao classificar documentos, queremos atribuir a um documento a melhor classe e, no caso dos classificadores bayesianos, a classe mais provável ou *maximum a posteriori* (MAP) c_{map} é(MANNING; RAGHAVAN; SCHUTZE, 2008):

$$c_{map} = \arg\max_{c \in C} \hat{P}(c \mid d) = \arg\max_{c \in C} \hat{P}(c) \prod_{1 \leq k \leq n_d} \hat{P}(t_k \mid c). \tag{5}$$

Na equação 5 é utilizado $\hat{P}$ ao invés de P uma vez que estes não são os valores reais dessas probabilidades, mas são estimativas extraídas do conjunto de treinamento. Em geral, essa forma de produtório não é muito utilizada. Como estão sendo multiplicadas probabilidades, pode ocorrer estouro de ponto flutuante. Para evitar isso, é possível utilizar o logaritmo das probabilidades, dado que $log(xy) = log(x) + log(y)$. Com isso, ao invés de termos um produtório, teremos um somatório:

$$c_{map} = \arg\max_{c \in C} \hat{P}(c \mid d) = \arg\max_{c \in C} log(\hat{P}(c)) \sum_{1 \leq k \leq n_d} log(\hat{P}(t_k \mid c)). \tag{6}$$

Dessa forma, o classificador bayesiano vai atribuir a um documento a classe que maximiza o resultado da equação 6. Podemos interpretar essa equação como sendo a soma da probabilidade *a priori* da classe c com as evidências de que esse documento pertence à classe c. A primeira refere-se a probabilidade de um documento qualquer ser da classe $c \in C$, sem que levemos em conta o conteúdo do documento. Já a segunda representa a probabilidade de que o documento seja da classe $c \in C$ levando em conta seu conteúdo. Em outras palavras, cada $\hat{P}(t_k \mid c)$ pode ser interpretado como a quantidade de evidência de que o *token* fornece sobre fato de o documento ser da classe c, e a classe a ser escolhida é aquela que possui mais evidências ou as evidências mais representativas.

Uma vez estabelecido o modelo na equação 6, é necessário estimar os parâmetros $\hat{P}(c)$ e $\hat{P}(t_k \mid c)$. Esses parâmetros são tipicamente estimados via *maximum likelihood estimation*(MLE), ou seja, para estimar $\hat{P}(c)$ são utilizadas as frequências das classes no conjunto de treinamento e para estimar $\hat{P}(t_k \mid c)$ são utilizadas as frequências relativas dos termos no conjunto de treinamento. Assim temos que, (MANNING; RAGHAVAN; SCHUTZE, 2008)

$$\hat{P}(c) = \frac{N_c}{N} \tag{7}$$

onde N_c é a quantidade de documentos no conjunto de treinamento pertencentes à classe c e N é a quantidade de documentos no conjunto de treinamento. Já as probabilidades $\hat{P}(t_k \mid c)$ são calculadas usando a frequência do termo t em documentos da classe c (MANNING; RAGHAVAN; SCHUTZE, 2008):

$$\hat{P}(t_k \mid c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}, \tag{8}$$

onde T_{ct} é a frequência do termo t em documentos da classe c.

Há um problema ao estimar as probabilidades usando *MLE*: esparsidade. Ou seja, é muito provável que haja termos nos documentos de teste que não foram vistos

nos documentos de treinamentos, ou não foram vistos para uma determinada classe. Se estivermos utilizando a equação 5, a ocorrência de um zero vai levar a zero a probabilidade $\hat{P}(c \mid d)$ o que é claramente um problema. Para evitar esse problema podemos utilizar, por exemplo, *add-one* ou *Laplace Smoothing* que adiciona 1 a cada probabilidade $\hat{P}(t_k \mid c)$ (MANNING; RAGHAVAN; SCHUTZE, 2008):

$$\hat{P}(t_k \mid c) = \frac{T_{ct} + 1}{\sum_{t' \in V}(T_{ct'} + 1)} = \frac{T_{ct} + 1}{\sum_{t' \in V}(T_{ct'}) + B}, \tag{9}$$

onde $B = |V|$ é igual ao número de termos distintos no vocabulário.

O classificador bayesiano foi revisado por ser utilizado em tarefas de classificação textual. Modelos de extração de tópicos de conjuntos de textos como *Latent Dirichlet Allocation* (BLEI; NG; JORDAN, 2003) e *Probabilistic Latent Semantic Indexing* (HOFMANN, 1999), baseiam-se no teorema de Bayes para extrair tópicos de documentos, ou seja, classificá-los. Tal classificador é relevante para este trabalho porque será proposto o uso de uma ferramenta de extração de tópico para representar o perfil de interesses do usuário.

2.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation é um modelo de tópicos probabilístico (BLEI, 2012) proposto por (BLEI; NG; JORDAN, 2003) cujo objetivo é explicar um conjunto de documentos através de um conjunto de tópicos latentes. Neste modelo, cada documento de um córpus pode ser visto como uma mistura de tópicos diversos e cada tópico pode ser visto como uma mistura de palavras do vocabulário do córpus. As subseções seguintes descrevem as suposições feitas a respeito do processo de produção dos documentos (subseção 2.2.1) e o processo de aprendizagem do modelo LDA (subseção 2.2.2).

2.2.1 Produção dos documentos

Em LDA são feitas algumas suposições a respeito dos documentos de um córpus e do processo que os gera. Primeiro, assume-se que os documentos são representados por um modelo *bag of words*. Neste modelo a ordem das palavras do documento não é considerada.

O processo de elaboração dos documentos considerado no modelo LDA é descrito a seguir (BLEI; NG; JORDAN, 2003).

1. Escolhe-se uma quantidade N de palavras.
2. Escolhe-se $\theta \sim Dir(\alpha)$
3. Para cada uma das N palavras w_n:
   a) Escolhe-se um tópico $z_n \sim Multinomial(\theta)$.
   b) Escolhe-se uma palavra w_n a partir de $p(w_n|z_n, \beta)$, que é uma probabilidade multinomial condicionada ao tópico z_n.

O processo geração de um documento inicia-se escolhendo sua quantidade de palavras (passo 1). A seguir (passo 2) extrai-se um vetor de parâmetros θ a partir de uma distribuição Dirichlet (KOTZ; BALAKRISHNAN; JOHNSON, 2000), em que α é um vetor de números reis positivos. Dirichlet é uma distribuição sobre distribuições Multinomiais e aqui é utilizada para extrair o vetor de parâmetros θ da distribuição de tópicos utilizada no passo seguinte.

Com isto, para cada palavra, escolhe-se um tópico z_n (passo 3.a) a partir de uma distribuição Multinomial com parâmetros θ. Finalmente, escolhe-se uma palavra w_n a partir da probabilidade Multinomial $p(w_n|z_n, \beta)$ condicionada ao tópico previamente escolhido. O parâmetro β desta probabilidade é uma matriz $k \times V$, em que k é a quantidade de tópicos e V é a quantidade de palavras do córpus. Nesta matriz, $\beta_(ij) = p(w^j = 1|z^i = 1)$, ou seja, a probabilidade da palavra w^j ser escolhida dado que o tópico z^i foi escolhido.

2.2.2 Aprendizagem

Há mais de uma forma possível para aprender, ou estimar, os parâmetros do modelo LDA. Nesta seção apresentamos uma abordagem de aprendizagem destes parâmetros conhecida como amostragem de Gibbs colapsada (PORTEOUS et al., 2008). Dada uma quantidade k de tópicos que se deseja extrair do conjunto de documentos D, o processo a ser seguido é descrito em alto nível a seguir (BLEI; NG; JORDAN, 2003).

1. Percorra cada documento d em D e atribua aleatoriamente um tópico z a cada palavra w.
2. Para cada documento d em D:
   a) Calcule a probabilidade $p(z = 1|d = 1)$ para cada tópico z.
   b) Para cada palavra w em d:

i. Calcule a probabilidade $p(w = 1|z = 1)$.

ii. Atribua a w um novo tópico z que maximize a probabilidade $p(z = 1|d = 1) * p(w = 1|z = 1)$

O processo é iniciado atribuindo-se aleatoriamente um tópico z a cada palavra w presente em cada documento do córpus (passo 1). Pode-se notar que com esta atribuição aleatória de tópicos às palavras, já tem-se a representação por tópicos de cada documento, assim como a distribuição de palavras para cada tópico. No entanto, estas representações tendem a ser imprecisas dado que os tópicos foram atribuídos aleatoriamente.

Em seguida, deve-se repetir o passo 2 iterativamente. Neste passo, que é executado para cada documento d da coleção, deve-se calcular a probabilidade $p(z = 1|d = 1)$ para cada tópico z, que é a proporção de palavras do documento d atribuídas ao tópico z.

Calcula-se então, para cada palavra w em d, a probabilidade $p(w = 1|z = 1)$, que é a porcentagem de documentos atribuídos ao tópico z e que contém a palavra w. Com estas probabilidades calculadas, deve-se atualizar o tópico atribuído a cada palavra w em d, escolhendo-se o novo tópico z que maximize a probabilidade $p(z = 1|d = 1) * p(w = 1|z = 1)$. Após repetir suficientemente o processo acima, as distribuições de probabilidade se estabilizarão.

Com as palavras dos documentos atribuídas aos seus respectivos tópicos, é possível estimar a mistura de tópicos de cada documento e a distribuição de palavras de cada tópico. A mistura de tópicos de um documento pode ser estimada utilizando-se a proporção de palavras deste documento atribuída a um tópico. Já a distribuição de palavras de um tópico pode ser calculada contando-se as quantidades das palavras atribuídas a este tópico.

2.3 Sistemas recomendadores

O problema de estimação de leitores potenciais para documentos em redes sociais pode ser modelado de forma a se beneficiar dos conceitos e do *framework* de sistemas de recomendação (RICCI et al., 2010). Nesta seção apresentaremos alguns dos conceitos fundamentais desta área.

Atualmente, os sistemas de recomendação estão presentes em vários sítios da web, desde recomendação de propagandas em motores de busca como o Google[1], recomendação

[1] www.google.com

de produtos em lojas virtuais como Amazon[2], recomendação de conteúdo multimídia como ocorre no Netflix[3], até recomendação de pessoas conhecidas no Facebook.

De modo geral, sistemas de recomendação solucionam o seguinte problema: dado um conjunto de itens I e um conjunto de usuários U, estimar quais itens em I são de interesse para um usuário em U. Itens podem ser os mais diferenciados conceitos como livros, notícias, amigos, propagandas etc. Para o caso de compartilhamento de documentos em redes sociais, onde estamos interessados em estimar para quais pessoas da rede social um determinado documento é interessante, os itens de nosso sistema de recomendação serão os próprios documentos.

Existem diferentes tipos de sistemas de recomendação distintos pelo modo que computam uma recomendação e pelo tipo de informação que é utilizada como fonte na computação. Em (RICCI et al., 2010) é apresentada uma introdução abrangente aos conceitos que envolvem os sistemas de recomendação, e uma classificação geral para esses sistemas. No início do desenvolvimento de sistemas recomendadores havia basicamente 3 categorias: baseado em conteúdo, filtro colaborativo e sistemas híbridos. Atualmente, há tipos mais específicos como sistemas de confiança reforçada (*trust-enhanced*) (RICCI et al., 2010) e sistemas baseados em demografia (RICCI et al., 2010). A seguir descreveremos as características dos sistemas baseados em conteúdo por terem maior afinidade com este trabalho, e abordaremos brevemente os sistemas de filtro colaborativo.

2.3.1 Sistemas de recomendação baseados em conteúdo

Os sistemas de recomendação baseados em conteúdo levam em conta o conteúdo textual de itens analisados pelo usuário para sugerir outros itens a esse usuário. Mais especificamente, esses sistemas analisam o conteúdo do conjunto de itens anteriormente consumidos pelo usuário, para então construir um perfil deste usuário (RICCI et al., 2010).

Em geral, esses sistemas compartilham uma arquitetura de alto nível compreendida por três componentes fundamentais: um analisador de conteúdo(*Content Analyser*), um componente para aprender o perfil(*Profile Learner*) e um filtro(*Filtering Component*) (RICCI et al., 2010).

[2] www.amazon.com
[3] www.netflix.com

O analisador de conteúdo é responsável por dar uma estrutura bem definida ao conteúdo que chega até o sistema recomendador sob diversas formas e sem representação padronizada. Tomando-se o exemplo do sistema recomendador de documentos, o conteúdo para esse sistema pode vir de páginas da web, de documentos em texto puro, e-mails, atualizações de redes sociais etc. Nesse caso, a função do analisador de conteúdo é transformar o conteúdo sem estrutura em uma representação que possa ser utilizada pelos próximos componentes do sistema. A representação dos itens retornada por esse componente é a entrada para os outros dois componentes do sistema.

O *Profile Learner*, ou componente para aprendizagem do perfil do usuário, é o módulo responsável por coletar os dados representativos para as preferências do usuário e para generalizar esses dados a fim de construir o perfil do usuário. Nesse módulo é comum a aplicação de técnicas de aprendizado de máquina como estratégia de generalização da informação coletada. De modo geral, esse módulo é responsável por construir e atualizar um modelo que represente o gosto do usuário.

Finalmente, o componente de filtragem é responsável por comparar a representação do perfil do usuário, expressa no modelo construído pelo *Profile Learner*, com a representação dos itens retornada pelo analisador do conteúdo. O resultado dessa comparação pode ser binário ou contínuo, ou seja, o filtro pode determinar que um dado item é recomendado ou não, ou estabelecer que um dado item é recomendado com um certo grau de relevância (RICCI et al., 2010).

O funcionamento de um sistema de recomendação baseado em conteúdo pode ser descrito pelos seguintes passos. O primeiro passo ocorre no analisador de conteúdo, onde a informação dos itens provenientes de diferentes fontes é transformada em representações estruturadas desses itens. A representação dos itens é então utilizada pelo componente de filtragem para que este possa gerar a lista de recomendações. Por último, as interações do usuário com os itens da lista de recomendações são passadas como entrada para o componente de aprendizado de perfil em conjunto com a representação dos itens, a fim de atualizar o modelo do perfil do usuário.

Sistemas de recomendação baseados em conteúdo padecem do problema de *cold-start*, ou início frio (RICCI et al., 2010). Quando um usuário começa a utilizar o sistema, não há itens com os quais o usuário tenha interagido e portanto não há um modelo que reflita as preferências do usuário.

2.3.2 Sistemas de recomendação baseados em filtragem colaborativa

Sistemas de recomendação baseados em filtro colaborativo, ou *Colaborative Filtering Recommender System*, diferentemente dos recomendadores baseados em conteúdo, fazem uso não apenas das interações de um usuário u, mas também das interações de outros usuários dentro do sistema. Sistemas desse tipo partem do princípio de que a interação (e.g., o fato de consumir ou não a informação) de um usuário u sobre um item i será similar àquela de um usuário v sobre o mesmo item, se u e v interagiram de forma parecida com itens do sistema no passado (RICCI et al., 2010). Tomando-se o exemplo do recomendador de documentos, supõe-se que um usuário u que tenha lido os mesmos documentos que um usuário v tende a ter a mesma interação sobre um novo item i.

Os sistemas de filtro colaborativo possuem algumas vantagens em relação aos baseados em conteúdo. Por exemplo, eles não dependem do conteúdo de um item para fazer uma recomendação. As recomendações são feitas segundo uma medida de qualidade expressa nas qualificações dos usuários. Dessa forma, os sistemas colaborativos podem recomendar itens totalmente diferentes dos que foram interessantes ou positivamente qualificados para o usuário no passado. Além disso, nos casos onde o conteúdo que descreve os itens é pobre ou até mesmo inexistente, os filtros colaborativos ainda conseguem fazer recomendações.

2.3.3 Sistemas de Recomendação de Confiança Reforçada

Trust Enhanced Recommender Systems, ou sistemas de recomendação de confiança reforçada, são sistemas que utilizam uma estrutura de rede social a fim de gerar recomendações mais precisas a um determinado usuário (RICCI et al., 2010). A inspiração para esses sistemas vem da forma que pedimos recomendações a pessoas em nosso dia-a-dia. Considere o exemplo de uma pessoa chamada João que acaba de chegar em uma cidade desconhecida e deseja encontrar um bom lugar para comer. Em geral, se um amigo de confiança lhe recomendar algum restaurante, essa recomendação lhe soará mais promissora do que uma vinda de um desconhecido. No caso de o amigo não conhecer nenhum restaurante, mas ter um amigo de confiança que conheça algum lugar, esse poderá usar a recomendação do amigo para indicar o restaurante ao João.

No caso de recomendação de produtos e serviços, o uso do conceito de confiança é bastante lógico. Contudo, tratando-se de recomendação da documentos talvez o conceito de confiança não desempenhe o mesmo papel. Uma possível diferença entre os dois casos seria de que o custo/risco em seguir uma recomendação pouco confiável é baixo no caso de documentos, mas relativamente alto em outros casos (como o de um filme ou restaurante ruim).

Pensando-se no ambiente de rede social, é fácil imaginar o caso onde uma pessoa se interessa em um documento publicado por uma outra pessoa sem a necessidade de se estabelecer uma relação de confiança entre elas. Sistemas de recomendação de confiança reforçada são distintos pela forma como o conceito de confiança é modelado e propagado através de uma rede de confiança. Dessa forma, é possível que os conceitos provenientes dos sistemas de confiança reforçada não sejam adequados ao problema de compartilhamento de documentos em redes sociais.

2.3.4 Avaliação de sistemas de recomendação de conteúdo

De forma geral, sistemas de recomendação de conteúdo são avaliados de acordo com a posição dos itens recomendados. Por exemplo, em um sistema de recomendação de conteúdo que gera uma lista de itens recomendados, espera-se que os itens de interesse de um usuário ocupem as primeiras posições da lista. Este é justamente o caso dos sistemas implementados nesta pesquisa em que a saída destes consiste em uma lista de *tweets* recomendados.

Desta maneira, a avaliação destes sistemas é realizada utilizando-se métricas que baseiam-se na posição que o item interessante ao usuário ocupa na lista produzida pelo sistema. Neste trabalho foram utilizadas três métricas que capturam esta informação: MRR (*Mean Reciprocal Rank*), S@5 e S@10 (ABEL et al., 2011a). MRR indica em qual posição média o item mais importante para o usuário ocorre na lista de itens recomendados pelo sistema. S@k indica a probabilidade do item de interesse ao usuário estar nas k primeiras posições da lista.

Este capítulo apresentou conceitos fundamentais de classificação e representação de documentos, e uma visão geral da tarefa computacional de recomendação de conteúdo. Estes conceitos – em especial o modelo LDA e o sistema de recomendação baseado em conteúdo – serão explorados na pesquisa descrita nos capítulos 5 e 6.

3 Trabalhos relacionados

Neste capítulo apresentamos estudos relacionados ao problema de recomendação de conteúdo no Twitter. O capítulo inicia-se descrevendo algumas características do Twitter relevantes à tarefa de recomendação (seção 3.1). Cada uma das seções seguintes discute individualmente um trabalho relacionado ao problema estudado nesta pesquisa.

3.1 Características da plataforma Twitter

A presente pesquisa, e alguns dos trabalhos apresentados a seguir, baseia-se em dados extraídos da rede social Twitter, e portanto é importante deixar claro suas principais características. O Twitter é uma das grandes redes sociais da web, com milhões de usuários ativos. Como a maioria das redes sociais, cada usuário do Twitter possui uma página usualmente denominada *timeline* onde são listadas as publicações, em ordem de data de publicação, dos usuários seguidos pelo usuário dono da *timeline*.

O Twitter possui duas características fundamentais: o texto de uma publicação não pode exceder 140 caracteres e o relacionamento entre usuários da rede é direcionado, ou seja, um usuário A pode seguir um usuário B sem que B siga A, o que implica A receber em sua *timeline* publicações de B. Um usuário também pode republicar *tweets* de outros usuários. Essa republicação é denominada *retweet*. Quando um usuário republica um *tweet*, ele está propagando na rede aquela informação, que será recebida pelos seus seguidores.

3.2 Analisando modelagem de usuário no Twitter para recomendação de notícias personalizadas

O trabalho apresentado em (ABEL et al., 2011a) avalia diferentes abordagens para a construção do perfil de interesses de um usuário do Twitter no contexto de recomendação de notícias. Em (ABEL et al., 2011a) foi desenvolvido um *framework* que gera perfis de acordo com o ajuste de três parâmetros: tipo de perfil, enriquecimento e restrições temporais. O tipo do perfil pode ser baseado em *hashtags* - termos iniciados com "#", baseado em tópicos ou baseado em entidades. O parâmetro de enriquecimento define se, na construção do perfil, será utilizado o conteúdo de notícias relacionadas aos *tweets* ou se apenas o conteúdo destes. O parâmetro referente a restrições temporais permite que o

conjunto de dados utilizados na construção do perfil seja restrito a *tweets* em finais de semana ou *tweets* publicados no último mês.

Em (ABEL et al., 2011a) o perfil de interesses pessoais do usuário é definido como um conjunto ponderado de conceitos:

$$P(u) = \{(c, w(u, c))|c \in C, u \in U\} \tag{10}$$

Onde C é o conjunto de conceitos dentro do sistema e U é o conjunto de usuários dentro do sistema. Aqui os conceitos podem ser *hashtags*, tópicos ou entidades nomeadas. Os autores utilizaram como função peso $w(u, c)$ a quantidade de *tweets* que referenciam um conceito c. Por exemplo, em um perfil do tipo baseado em entidades, $w(u, USP) = 5$ significa que o usuário u publicou 5 *tweets* que referenciam a entidade USP. Além disso, os autores normalizam os pesos de modo que a soma dos pesos de todos os conceitos para um determinado usuário seja igual a 1.

Em (ABEL et al., 2011a) a extração de tópicos e entidades do córpus de *tweets* foi realizada utilizando-se a ferramenta OpenCalais[1]. O enriquecimento dos *tweets* com conteúdo de notícias foi realizado baseando-se em associações explícitas, ou seja, nos casos em que há um link para a notícia no conteúdo do *tweet* e em estimativas de associação. Essas associações estimadas foram computadas utilizando-se em um método proposto pelos próprios autores em um trabalho anterior (ABEL et al., 2011b), que se baseia em entidades extraídas dos *tweets*, das notícias e também de registro de horário de ambos. De acordo com Abel et al., esse método de associação apresentou uma acurácia superior a 70% e, apesar do ruído inserido no perfil do usuário, essas associações exercem um efeito positivo na modelagem do usuário.

O estudo do comportamento temporal dos perfis mostrou que para todos os tipos de perfis há uma evolução com o tempo. Para capturar essa evolução, foi utilizada uma medida de distância denominada *distância-d1*, que mede a diferença entre perfis em suas representações vetoriais (LIU; DOLAN; PEDERSEN, 2010):

$$d_1(\hat{p}_x(u), \hat{p}_y(u)) = \sum_i |p_{x,i} - p_{y,i}| \tag{11}$$

Quanto maior $d_1(\hat{p}_x(u), \hat{p}_y(u)) \in [0..2]$, maior a diferença entre os perfis. De acordo com Abel et al., perfis baseados em *hashtags* sofrem maiores mudanças ao longo do tempo,

[1] http://www.opencalais.com

enquanto que perfis baseados em entidades são os que menos sofrem. Ainda assim, todos os perfis mudam ao longo do tempo. O estudo também apontou que para 24,9% dos usuários avaliados, perfis gerados com dados de finais de semana diferem significativamente, a uma *distância-d1* ≈ 2, de perfis gerados com dados de dias úteis. Essa diferença pode indicar que alguns usuários demonstram interesses distintos em finais de semana.

Para construir e avaliar os diferentes modelos de usuário, Abel et al. coletaram um conjunto de dados composto por *tweets* e por notícias de mais de 20 mil usuários. Além disso, foram monitorados mais de 60 RSS *feeds* de importantes fontes de notícia como BBC[2], CNN[3] e The New York Times[4], resultando em 2.316.204 *tweets* e 77.544 artigos de notícia (ABEL et al., 2011a).

Cada um dos perfis foi avaliado no contexto de recomendação seguindo um mesmo algoritmo de recomendação. Como o foco do trabalho não era experimentar estratégias de recomendação, mas sim avaliar a qualidade dos perfis, foi aplicado um algoritmo simples de recomendação por conteúdo que recomenda itens de acordo com sua similaridade cosseno com um determinado perfil de usuário. Dessa forma, os itens também são representados como o perfil descrito na equação 10. Nessa estratégia, o problema de recomendação é abordado como um problema de recuperação e classificação onde o perfil do usuário exerce a função de *query* de busca.

As medidas de avaliação utilizadas em (ABEL et al., 2011a) foram *MRR (Mean Reciprocal Rank)* e *S@k* (Sucesso na posição k) discutidas na seção 2.3.4.

$$sim_{cosine}(\hat{p}(u), \hat{p}(n_i)) = \frac{\hat{p}(u).\hat{p}(n_i)}{||\hat{p}(u)||.||\hat{p}(n_i)||} \tag{12}$$

De acordo com (ABEL et al., 2011a), perfis do tipo baseado em entidades geraram os melhores resultados no contexto de recomendação, enquanto que os perfis baseados em *hashtags* apresentaram os piores resultados. O enriquecimento dos *tweets* com conteúdo de notícia também se mostrou favorável tanto nos perfis baseados em entidades quanto nos baseados em tópicos. Quanto ao fator temporal, os resultados mostram que, para perfis baseados em entidades, a limitação para os *tweets* presentes nas últimas 2 semanas anteriores à semana de recomendação degradou a qualidade das recomendações. Ou seja, perfis construídos com todo histórico de *tweets* tiveram desempenho melhor que aqueles

[2] www.bbc.co.uk/
[3] www.cnn.com
[4] www.nytimes.com

construídos apenas com os *tweets* mais recentes. Entretanto, esse padrão se inverte para os perfis baseados em tópicos, onde perfis construídos apenas com *tweets* recentes geraram melhores recomendações.

O estudo apresentado em (ABEL et al., 2011a) contribui para a presente pesquisa por avaliar o desempenho de perfis construídos a partir de abordagens distintas. Abel et al. demonstram que um perfil baseado em entidades aproxima melhor os interesses de leitura dos usuários e que restrições temporais podem impactar positivamente esta aproximação.

3.3 Combinando modelagem de tendências e de usuário para recomendações personalizadas de notícias

Em (GAO et al., 2011) é apresentado um estudo comparando três perfis de usuário construídos a partir de dados do Twitter: um perfil representando os interesses pessoais do usuário, outro representando tendências globais da rede e o terceiro sendo uma combinação dos anteriores. Além disso, é aplicado um fator temporal ao modelo de tendências globais com a finalidade de capturar itens que possuem popularidade pontual. Cada um desses modelos é discutido individualmente abaixo.

O perfil de interesses pessoais do usuário em (GAO et al., 2011) é definido de forma análoga ao perfil descrito na equação 10 da seção 3.2, porém definindo-se C, o conjunto de conceitos, como o conjunto de entidades dentro do sistema. Uma entidade pode ser uma pessoa, organização, local, evento etc. As entidades foram associadas aos *tweets* usando-se a ferramenta OpenCalais[5]. Em (GAO et al., 2011) foi utilizada como função peso $w(u, c)$ a quantidade de *tweets* que referenciam uma entidade c. Posteriormente os pesos foram normalizados de modo que a soma dos pesos de todos os conceitos para um determinado usuário seja igual a 1.

O segundo perfil, que representa as tendências, é modelado de forma semelhante ao perfil de interesses pessoais em (ABEL et al., 2011a):

$$T(I_j) = \{(c, w(I_j, c)) | c \in C\} \tag{13}$$

Nesta equação, C representa o conjunto de conceitos candidatos a partir dos quais as tendências serão extraídas em um intervalo de tempo I_j. Para o perfil de tendências, os autores propuseram para a função peso w uma função TFxIDF (MANNING; RAGHAVAN;

[5] http://www.opencalais.com

SCHUTZE, 2008) modificada por um fator temporal que garante mais peso a conceitos que apresentam picos de popularidade, e não uma popularidade constante. Para um dado intervalo de tempo I_j, a frequência de termo TF é calculada como (MANNING; RAGHAVAN; SCHUTZE, 2008):

$$w_{TF}(I_j, c) = \frac{n_{c,j}}{\sum_{c \in C} n_{c,j}} \tag{14}$$

Nesta equação, $n_{c,j}$ representa o número de *tweets* que referenciam o conceito c durante o intervalo de tempo I_j.

A frequência de documento inversa é calculada, para um determinado intervalo de tempo I_j, da seguinte forma (MANNING; RAGHAVAN; SCHUTZE, 2008):

$$w_{TFxIDF}(I_j, c) = w_{TF}(I_j, c).log(\frac{|I|}{1 + |\{I_i : n_{c,i} > 0\}|} n_{c,j}) \tag{15}$$

Nesta equação, $|I|$ denota a quantidade de intervalos temporais e $|\{I_i : n_{c,i} > 0\}|$ representa a quantidade de intervalos que o conceito c fora referenciado pelo menos uma vez.

O fator temporal para um conceito c é calculado usando-se o desvio padrão dos instantes dos *tweets* que o referenciam (GAO et al., 2011).

$$\sigma(c) = \sqrt{\frac{\sum_{k=1}^{N} (ts_k - \bar{ts})^2}{N - 1}} \tag{16}$$

Nesta equação, ts_k representa o instante do k-ésimo *tweet* que faz referencia ao conceito c, $\bar{ts}$ representa a média dos instantes dos *tweets* que referenciam c e N denota o total de *tweets* que referenciam c. Quanto menor o desvio padrão de um conceito, mais concentrados estarão os *tweets* que o referenciam. Isto permite lançar mão desse termo para destacar conceitos que possuem picos de popularidade em detrimento daqueles que possuem popularidade constante.

A função peso proposta em (GAO et al., 2011) para o perfil de tendências, denominada TFxIDF sensível ao tempo, é calculada da seguinte forma para um determinado conceito c em um dado intervalo I_j:

$$w_{t-TF}(I_j, c) = w_{TF}(I_j, c).(1 - \hat{\sigma(c)}) \tag{17}$$

Nesta equação, $\hat{\sigma(c)}$ representa o desvio padrão $\sigma(c)$ normalizado.

Finalmente, o terceiro perfil avaliado é modelado usando uma ponderação simples (GAO et al., 2011):

$$\vec{m}(I_j, u) = d * \vec{p}(u) + (1 - d) * \vec{t}(I_j) \tag{18}$$

Nesta equação, $\vec{p}(u)$ é a representação no espaço vetorial de $P(u)$, sendo que seu i-ésimo elemento representa $w(u, c_i)$ e $\vec{t}(I_j)$ representa $T(I_j)$ no espaço vetorial onde seu i-ésimo elemento denota $w(I_j, c_i)$.

De acordo com (GAO et al., 2011), o modelo que teve o melhor desempenho foi o modelo misto, com um valor de d igual a 0.6.

O estudo apresentado em (ABEL et al., 2011a) contribui para a presente pesquisa por demonstrar uma forma de exploração de aspectos temporais no processo de recomendação.

3.4 Personalização de notícias direcionada por mídia social

Em (O'BANION; BIRNBAUM; HAMMOND, 2012) foi desenvolvido um sistema de recomendação de notícias aplicado ao jornal *The Hunffington Post*[6]. O sistema de recomendação foi utilizado para gerar versões personalizadas da página principal do jornal. Nesse sistema, foram utilizadas informações de *tweets* e de artigos de notícias do jornal para modelar o usuário.

Em (O'BANION; BIRNBAUM; HAMMOND, 2012) seguiu-se uma abordagem voltada à organização de artigos de notícias em jornais. Nessa abordagem, foram treinados dois classificadores textuais: um que atribui uma categoria a *tweets*, e outro que atribui *tags* a *tweets*. Cada artigo do jornal tem atribuído uma categoria e um conjunto de *tags*. Uma categoria fornece à notícia uma classificação alto nível (e.g., Economia, Política, etc.), enquanto que *tags* são, em geral, pessoas, locais ou tópicos relevantes à notícia.

Estando os classificadores treinados, cada *tweet* de um determinado usuário é coletado e associado a uma categoria e um conjunto de *tags*. Dessa forma, o perfil do usuário é composto por categorias e *tags* que foram associadas ao seu conjunto de *tweets* e pode ser descrito pela equação abaixo (O'BANION; BIRNBAUM; HAMMOND, 2012):

$$P(u) = \{(c, w(u, c))|c \in C, u \in U\}, \{(t, w(u, t), maxdate(t))|t \in T, u \in U\} \tag{19}$$

[6] www.thehuffingtonpost.com

Nesta equação, U é o conjunto de todos os usuários, C é o conjunto de categorias de notícia que aparecem no conjunto de *tweets* do usuário u, T é o conjunto de *tags* que aparecem no conjunto de *tweets* do usuário e $maxdate(t)$ representa o registro no tempo do último *tweet* em que a *tag* t foi atribuída.

O termo $w(u,t)$ representa a quantidade de *tweets* que estão relacionados à *tag* t posteriormente normalizada para que, para um dado usuário, a soma de $w(u,t)$ seja igual a 1. O parâmetro *maxdate* é utilizado no algoritmo de recomendação para conferir maior peso àquelas *tags* atribuídas a *tweets* recentes. Para cada *tag* $t \in T$ é calculado um valor que representa o nível de interesse do usuário naquela *tag*. Esse valor é posteriormente utilizado para estimar o interesse do usuário nas categorias. Seu cálculo é descrito na equação abaixo (O'BANION; BIRNBAUM; HAMMOND, 2012):

$$TS(u,t) = w(u,t) * 9^d \tag{20}$$

Nesta equação, d é a diferença em dias entre a data atual e a data registrada em $maxdate(t)$. O interesse de um usuário em uma categoria, representado por $w(u,c)$, é estimado utilizando-se a equação abaixo (O'BANION; BIRNBAUM; HAMMOND, 2012):

$$w(u,c) = \sum_{t \in TC} TS(u,t) \tag{21}$$

Nesta equação, TC é o conjunto de todas as tags que foram atribuídas aos *tweets* classificados como sendo da categoria c.

A hipótese em (O'BANION; BIRNBAUM; HAMMOND, 2012) é a de que a ponderação do termo $w(u,t)$ por 9^d (nove elevado à d-ésima potência) ajudará a capturar mudanças nos interesses de notícias do usuário. Como o sistema foi modelado especificamente para um portal de notícias, a avaliação utilizada é ligada ao caso de uso do sistema.

De acordo com (O'BANION; BIRNBAUM; HAMMOND, 2012), o modelo proposto apresentou um desempenho consideravelmente melhor que o método usado para comparação: as notícias presentes na seção "Populares"do jornal.

O trabalho apresentado em (O'BANION; BIRNBAUM; HAMMOND, 2012) é de interesse para a presente pesquisa por demonstrar uma abordagem de recomendação utilizando um classificador treinado com artigos de notícias. O modelo proposto também demonstra uma abordagem para conferir peso a noticias recentes.

3.5 Recomendações baseadas em comentários escritos por usuários em mídias sociais

Em (MESSENGER; WHITTLE, 2011) é apresentada uma proposta de modelagem do usuário a partir de comentários de notícias registrados no portal britânico de notícias The Guardian[7]. Em (MESSENGER; WHITTLE, 2011) são avaliados dois métodos de construção de perfis: um combinando técnicas de processamento de língua e heurísticas para extrair categorias semânticas dos comentários, e outro usando apenas técnicas de processamento de língua para extrair termos multi-palavras do conjunto de comentários.

Na abordagem que extrai categorias dos comentários foi utilizada uma ferramenta de processamento de língua chamada Wmatrix[8]. Esta ferramenta rotula cada termo de um texto com sua classe gramatical e um conjunto de categorias semânticas (e.g., *staff*, de acordo com a ferramenta, faz parte das categorias *Work and Employment* e *People*). Para cada comentário, são atribuídas até duas categorias usando-se os primeiros dois substantivos ou caso não haja substantivos, usando-se adjetivos.

Na segunda abordagem em (MESSENGER; WHITTLE, 2011), é utilizado um conjunto de ferramentas de processamento de língua. Dentre estas destaca-se um extrator de termos valor-C derivado de (ZHANG JOSE IRIA; CIRAVEGNA, 2008). Esta ferramenta é utilizada para extrair termos do conjunto de comentários e atribuir a cada termo um valor-C que indica a importância do termo dentro do conjunto de comentários.

No sistema proposto em (MESSENGER; WHITTLE, 2011), os 10 termos ou conceitos mais relevantes ao conjunto de comentários de um artigo de notícia são utilizados para construir uma *query* de busca para o sistema Google News[9]. Os 10 primeiros resultados retornados são entregues ao usuário como recomendações.

De acordo com (MESSENGER; WHITTLE, 2011), a abordagem usando extração de termos demonstrou um desempenho superior à abordagem de extração de categorias. A diferença de desempenho foi justificada pelo fato de uma categoria ser um conceito abstrato e pouco específico. Termos extraídos, por outro lado, são conceitos específicos que, quando utilizados para construir a *query* de busca, são capazes de capturar notícias mais relevantes ao conteúdo dos comentários.

[7] www.theguardian.co.uk
[8] http://ucrel.lancs.ac.uk/wmatrix/
[9] news.google.com

O modelo apresentado em (MESSENGER; WHITTLE, 2011) é de interesse para a presente pesquisa por demonstrar o uso de um classificador gramatical combinado com heurísticas para extrair categorias de pequenos documentos, que no caso são comentários.

3.6 De conversas a manchetes: explorando a *web* em tempo real para recomendação personalizada de notícias

No trabalho apresentado em (MORALES; GIONIS; LUCCHESE, 2012) é proposto um sistema de recomendação de notícias que modela o perfil do usuário a partir de dados de sua rede social no Twitter e do conteúdo de artigos de notícia. A abordagem em (MORALES; GIONIS; LUCCHESE, 2012) se diferencia das abordagens apresentadas nas seções anteriores pelo uso das relações sociais existentes entre usuários do Twitter. O uso das relações entre usuários do Twitter permite, por exemplo, modelar o perfil de um usuário novo que não possui nenhuma publicação mas segue outros usuários, resolvendo assim o problema do início frio (RICCI et al., 2010).

O modelo proposto captura a relação entre notícias e usuários com base em três componentes distintos: similaridade baseada no conteúdo, similaridade baseada em relações sociais e popularidade de notícias baseada em entidades. Cada um destes componentes é descrito individualmente abaixo.

A similaridade baseada em conteúdo captura relações entre um conjunto de notícias e o conjunto de *tweets* publicados por um usuário. Essas relações são intermediadas por um conjunto de entidades. Na proposta de (MORALES; GIONIS; LUCCHESE, 2012), entidades são artigos da Wikipedia[10] que são relacionados aos *tweets* e às notícias por meio de uma ferramenta chamada Spectrum (PARANJPE, 2009). Esta ferramenta atribui um valor à relação de acordo com a significância da entidade no *tweet* ou notícia. A relação entre uma notícia e um *tweet* é dada pela soma dos produtos de suas relações com as entidades comuns entre si. Em outras palavras, seja E o conjunto de entidades comuns à notícia n e ao *tweet* t, a relação entre n e t é dada por (MORALES; GIONIS; LUCCHESE, 2012):

$$r(n,t) = \sum_i w(e_i,n) * w(e_i,t), e_i \in E \tag{22}$$

[10] www.wikipedia.org

Nesta equação, $w(e_i, n)$ representa a relação entre a notícia n e a n-ésima entidade e $w(e_i, t)$ representa a relação entre o *tweet* t e a n-ésima entidade. A relação entre uma notícia n e um usuário u é dada pela soma do termos $r(n, t)$, para todo *tweet* t publicado por u.

A similaridade baseada nas relações sociais funciona de maneira semelhante à similaridade baseada em conteúdo. Entretanto, na primeira não são usados apenas os *tweets* do usuário para modelar seu interesse na notícia, mas também as publicações de usuários conectados diretamente a ele. Ou seja, os *tweets* dos usuários que um determinado usuário u segue são utilizados para compor o conjunto de *tweets* que serão conectados às notícias, porém sofrendo uma ponderação conforme a equação abaixo (MORALES; GIONIS; LUCCHESE, 2012):

$$rs(n, ft) = \sum_i w(e_i, n) * w(e_i, ft) * 0.85 * f, e_i \in E \tag{23}$$

Nesta equação, ft representa um *tweet* de um usuário seguido, n representa um artigo de notícia, f é igual a 1 divido pelo número de usuários seguidos, 0,85 é um valor de amortização arbitrário e E representa o conjunto de entidades comuns aos *tweets* dos usuários seguidos e às notícias. O componente de similaridade social entre uma notícia e um usuário é dado pela soma dos termos $r(n, ft)$ e $rs(n, t)$, para todo *tweet* t publicado por u e todo *tweet* ft publicado por um usuário seguido por u.

Finalmente, o componente que captura a popularidade de notícias por meio de entidades tem por objetivo permitir que o sistema recomende notícias que, embora muito divulgadas, não seriam capturadas pelos outros componentes do sistema. Neste componente a popularidade de entidades se dá pela contagem das associações destas aos *tweets* e às notícias em uma determinada janela de tempo. Em (MORALES; GIONIS; LUCCHESE, 2012) é aplicado ainda um fator de decaimento exponencial que atribui maior peso à contagem de entidades recentes em detrimento de contagens passadas.

Seja cada um dos componentes descritos denominados por $sbc(u, n)$, $sbs(u, n)$ e $pn(u, n)$, respectivamente. O sistema proposto em (MORALES; GIONIS; LUCCHESE, 2012) recomenda, em um determinado instante, uma lista de artigos de notícias de acordo com a equação abaixo:

$$RN(u, n) = \alpha * sbc(u, n) + \beta * sbs(u, n) + \gamma * pn(u, n) \tag{24}$$

Nesta equação, α, β e γ são coeficientes que especificam o peso relativo de cada componente. Em (MORALES; GIONIS; LUCCHESE, 2012) é utilizado classificação SVM para otimizar esses parâmetros em um conjunto de treinamento.

A proposta apresentada em (MORALES; GIONIS; LUCCHESE, 2012) é de interesse para a presente pesquisa por demonstrar o uso de um extrator de entidades baseado em artigos da Wikipedia e por demonstrar formas de abordar aspectos sociais e temporais durante o processo e recomendação.

3.7 Recomendação de URLs na plataforma Twitter utilizando-se aspectos sociais da plataforma

Em (CHEN et al., 2010), é apresentado um estudo comparando diferentes opções de projeto de um sistema de recomendação de URLs baseado na plataforma Twitter. O sistema, que recomenda URLs compartilhadas no Twitter aos seus usuários, pode ser configurado modificando-se três características fundamentais: seleção de URLs candidatas, uso de informação de conteúdo e uso de informação de rede social.

Em (CHEN et al., 2010) é avaliado o uso de dois conjuntos de URLs candidatas: um conjunto constituído pelas URLs presentes em *tweets* publicados pela vizinhança de até segundo nível do usuário (i.e., *tweets* publicados pelos usuários seguidos e também pelos usuários que estes seguem), e um conjunto composto pelas URLs mais populares.

Em (CHEN et al., 2010) é proposto que um usuário do Twitter apresenta dois perfis de interesses distintos na plataforma: um perfil de interesses como produtor de informação e outro como consumidor de informação. Com base nisso, é avaliado o uso de um perfil de interesses construído a partir de *tweets* publicados pelo usuário - o perfil como produtor; e um perfil construído a partir de *tweets* de sua vizinhança de até segundo nível - o perfil como consumidor. Em ambos perfis, é utilizado um modelo *bag-of-words* TFxIDF (MANNING; RAGHAVAN; SCHUTZE, 2008) para representar o interesse de um usuário em um determinado tópico, representado por uma palavra.

Para modelar um processo de votação social, foram exploradas informações de conexão entre usuários do Twitter. Nesse processo, cada usuário dentro da vizinhança de segundo nível vota a favor de uma URL no caso desse usuário ter mencionado essa URL em algum *tweet*. Os usuários que possuem mais seguidores dentro dessa vizinhança de segundo nível possuem maior poder de voto. De acordo com (CHEN et al., 2010), essa

abordagem tem o objetivo de capturar o conceito de confiança entre um usuário e os usuários que este segue. A frequência com que um usuário publica *tweets* também interfere no poder do voto, e é proposto que usuários que publicam menos devem obter um poder de voto maior.

Para avaliar a contribuição de cada opção descrita acima, são usados 12 modelos de recomendação representando todas as variações possíveis das escolhas de projeto. De acordo com (CHEN et al., 2010), o modelo que apresentou os melhores resultados utiliza o conjunto de URLs candidatas publicadas pelo usuário e sua vizinhança de segundo nível, utiliza o perfil do usuário como produtor de informação e também faz uso do processo de votação social.

O estudo em (CHEN et al., 2010) é relevante para o presente trabalho por ilustrar o uso de informações da rede social presente no Twitter no processo de recomendação de URLs.

3.8 Recomendação de *Tweets* com co-classificação de grafo

Em (YAN; LAPATA; LI, 2012) é apresentado um modelo teórico baseado em grafo que recomenda *tweets* a usuários do Twitter. O modelo apresentado classifica simultaneamente *tweets* e autores usando três grafos: um grafo $G_U = (V_U, E_U)$ conectando os autores, um grafo $G_M = (V_M, E_M)$ conectando os *tweets* entre si, e um terceiro grafo $G_{MU} = (V_{MU}, E_{MU})$ conectando *tweets* ao seus autores. V_U representa o conjunto de usuários, E_U representa o conjunto conexões entre usuários, V_M representa o conjunto de *tweets*, E_M representa o conjunto de conexões entre *tweets* e, $V_{MU} = V_M \cup V_U$ e E_{MU} representam as conexões entre cada *tweet* e seus autores. Tanto os *tweets* quanto os usuários são classificados utilizando-se um algoritmo de co-classificação inspirado pelo algoritmo PageRank (BRIN; PAGE, 1998).

O algoritmo de co-classificação proposto em (YAN; LAPATA; LI, 2012) combina a classificação de dois grafos distintos: o grafo conectando *tweets* e o grafo conectando usuários. A classificação de cada grafo é descrita a seguir.

A classificação do grafo de *tweets* é uma variação do algoritmo PageRank que permite a personalização do ranqueamento. O algoritmo PageRank padrão classificaria um conjunto de *tweets* sem levar em conta os interesses de informação de um determinado usuário. Seja M a matriz de transição do grafo de *tweets*, onde cada entrada representa

um valor de similaridade cosseno entre dois *tweets* e σ é um fator de amortização. Neste caso m, o vetor representando o PageRank de cada *tweet*, é definido por:

$$m = (1 - \sigma)M^T m + \frac{\sigma}{|V_M|}11^T \tag{25}$$

Nesta equação, M^T representa a transposta da matriz M, V_M representa o conjunto de *tweets* e 1 é um vetor de $|V_M|$ entradas unitárias.

Para adicionar personalização à classificação, o trabalho em (YAN; LAPATA; LI, 2012) utiliza um modelo de tópicos representado em um vetor $t = [t_1, t_2, \ldots, t_n]_{1xn}$ onde n representa o número de tópicos e t_i representa o grau de interesse de um usuário no tópico i. É utilizada a técnica de *Latent Dirichlet Allocation* (BLEI; NG; JORDAN, 2003) para extrair uma matriz de distribuição D, onde uma entrada D_{ij} representa a probabilidade do *tweet* m_i pertencer ao tópico t_j. Dessa forma, é possível calcular o vetor r, onde cada entrada r_i representa a probabilidade do usuário responder, ou *retuitar*, o *tweet* m_i:

$$r = tD^t \tag{26}$$

A forma personalizada do algoritmo PageRank pode ser obtida através da equação abaixo (YAN; LAPATA; LI, 2012):

$$m = (1 - \sigma)[Diag(r)M]^t m + \sigma r \tag{27}$$

Apesar da classificação obtida pela equação 27 ser personalizada, ela não garante diversidade, uma vez que será dado mais peso a conjuntos de *tweets* conectados e próximos. Assim, em (YAN; LAPATA; LI, 2012) é proposta uma alteração inspirada no algoritmo DivRank (MEI; GUO; RADEV, 2010). Nessa abordagem, assume-se que a probabilidade de transição entre os estados muda ao longo do tempo e em cada passo do algoritmo é criada uma matriz de transição dinâmica $M^{(\cdot)}$ que, após z iterações, torna-se:

$$M^{(z)} = (1 - \sigma)M^{z-1}m^{(z-1)} + \sigma r \tag{28}$$

Assim, m pode ser recalculado da seguinte forma (YAN; LAPATA; LI, 2012):

$$m^{(z)} = (1 - \sigma)[Diag(r)M^{(z)}]^T m + \sigma r \tag{29}$$

A classificação do grafo de autores ocorre de forma similar à classificação personalizada do grafo de *tweets*. De acordo com (YAN; LAPATA; LI, 2012), o grafo de autores

não é classificado com a abordagem DivRank pois não é necessário obter-se diversidade nessa classificação, uma vez que os usuários são, por definição, únicos. Na classificação dos autores, a personalização é obtida através de um vetor $p = [p_1, p_2, \ldots, p_n]$ onde n corresponde ao número de autores e os termos p_i são obtidos conforme a equação abaixo:

$$p_i^u = \frac{\#tweets\ vindos\ de\ u_i}{\#tweets\ de\ u} \tag{30}$$

Nesta equação, p_i^u representa a proporção de *tweets* herdados do usuário u_i, e um valor de p_i^u alto significa que o usuário u provavelmente responderá aos *tweets* de u_i.

Com isso, sendo u a matriz de transição do grafo de usuários, a classificação dos autores é obtida conforme a equação a seguir (YAN; LAPATA; LI, 2012):

$$u = (1 - \sigma)[Diag(p)U]^T u + \sigma p \tag{31}$$

Na equação acima, cada entrada (i, j) da matriz U é maior que zero caso o usuário i siga o usuário j, e zero caso contrário. Não se diz que o valor da entrada é unitário porque a matriz é normalizada.

Para descrever o algoritmo de co-classificação é necessário definir duas matrizes $MU_{|V_M|x|V_U|}$ e $UM_{|V_U|x|V_M|}$ que representam o grafo G_{MU}. Cada entrada $\bar{W}_{ij}$ da matriz MU representa a probabilidade condicional de transição do *tweet* m_i para o usuário u_j. Já na matriz UM ocorre o contrário: cada entrada $\hat{W}_{ji}$ representa a probabilidade condicional de transição do usuário u_j para o *tweet* m_i.

O algoritmo proposto em (YAN; LAPATA; LI, 2012) é executado em dois passos. Em um calcula-se os *tweets* mais importantes. No outro, calcula-se os usuários mais importantes (YAN; LAPATA; LI, 2012):

Passo 1:

$$m^{(z+1)} = (1 - \sigma)([Diag(r)M^{(Z)}]^T)m^{(z)} + \sigma UM^T u^{(z)} \tag{32}$$

Passo 2:

$$u^{(z+1)} = (1 - \sigma)([Diag(p)U]^T)u^{(z)} + \sigma MU^T m(z) \tag{33}$$

O trabalho apresentado em (YAN; LAPATA; LI, 2012) é relevante para a presente pesquisa por demonstrar a abordagem do problema de recomendação de *tweet* usando métodos inspirados no algoritmo PageRank.

3.9 Considerações

Neste capítulo foi apresentada uma visão geral da pesquisa em recomendação de documentos em redes sociais. Dentre estes estudos, alguns terão especial interesse para a presentes pesquisa, como o trabalho apresentado em (ABEL et al., 2011a) utilizado como base para o estudo preliminar descrito no próximo capítulo e o trabalho de (YAN; LAPATA; LI, 2012) por ilustrar o uso de uma ferramenta de extração de tópicos em conjunto de *tweets* que é uma abordagem similar à que será adotada em nosso experimento principal, descrito no capítulo 6.

4 Estudo exploratório

Como um primeiro passo a fim de obter maior conhecimento sobre o problema de recomendação em redes sociais, a presente pesquisa foi iniciada com a implementação de um sistema simples de recomendação de *tweets*. O sistema foi treinado e avaliado usando-se um conjunto de dados extraídos da plataforma Twitter, contendo aproximadamente 90 milhões de *tweets* de 48 mil usuários. O objetivo do sistema implementado foi o de recomendar a um usuário seguidor o conjunto de *tweets* mais relevantes produzidos por usuários seguidos. Além disso, a implementação proposta explora certos aspectos temporais da tarefa de recomendação que serão refinados no experimento principal desta dissertação a ser descrito no capítulo 6

Este capítulo está organizado da seguinte forma. A seção 4.1 descreve o funcionamento do sistema implementado. A seção 4.2 apresenta os detalhes da representação e da construção do perfil de interesses do usuário. A seção 4.3 descreve a implementação do algoritmo de coleta de dados e as características do conjunto. A metodologia de avaliação do sistema é explicada na seção 4.4. Os resultados obtidos pelo sistemas são apresentados na seção 4.5 e discutidos na seção 4.6.

4.1 Visão geral

O sistema recomendador implementado, que é uma adaptação do sistema proposto em (ABEL et al., 2011a), utiliza um vetor ponderado de elementos para a representação do perfil de interesses do usuário. Cada perfil de usuário é construído utilizando-se *hashtags*, palavras iniciadas com o símbolo sustenido (#), publicadas em *tweets*. A opção pelo uso de *hashtags* foi motivada pelo interesse em implementar um modelo simples, de caráter ilustrativo do problema computacional da recomendação de conteúdo em microblogs.

A diferença fundamental entre o sistema implementado e aquele proposto por (ABEL et al., 2011a) consiste nos itens de recomendação. Enquanto que no sistema proposto em (ABEL et al., 2011a) os itens são links para artigos de notícia, no sistema implementado os itens são os próprios *tweets*.

O sistema recebe como entrada a *timeline* do usuário e a utiliza para construir o perfil de interesses do usuário. As recomendações são geradas com base em dois instantes de tempo t e $t\prime$, sendo $t\prime$ anterior a t. As publicações da *timeline* do usuário anteriores a t

e posteriores a $t\prime$ compõem o conjunto de itens candidatos. Para cada publicação coletada, é calculada sua similaridade com o perfil de interesses do usuário e então o sistema retorna uma lista contendo as publicações ordenadas de acordo com o valor de similaridade.

Em um estudo preliminar, foi observado que a janela de tempo utilizada para extrair candidatos para recomendação tem um impacto significativo na qualidade das recomendações. Essa janela de tempo foi definida acima como o intervalo entre os instantes $t\prime$ e t. Por este motivo, o sistema foi implementado de modo que fosse possível especificar a janela de recomendações a ser utilizada durante a computação das recomendações.

Neste estudo exploratório, nossa hipótese inicial foi a de que diferentes usuários possuem diferentes janelas de tempo ideais de candidatos a recomendação. Ou seja, um usuário que segue um número grande de usuários, e por conta disso recebe diariamente uma grande quantidade de publicações, terá uma janela menor de candidatos à recomendação. Isto ocorre porque, quando este acessa sua *timeline*, é pouco provável que sua lista de *tweets* seja percorrida até uma publicação muito antiga. Por outro lado, um usuário que siga um número menor de pessoas, e portanto exposto a um volume menor informação, poderá percorrer sua lista de *tweets* até publicações mais antigas. Além disso, usuários distintos podem possuir padrões de uso distinto da plataforma, consumindo informação em ritmos diferentes.

Com base nisso, experimentamos o sistema de recomendação com duas janelas distintas para coleta de candidatos: uma janela de 5 dias constante para todos usuários, e uma janela personalizada para cada usuário, calculada com base no intervalo médio entre os *retweets* dos usuários e as publicações originais.

Tanto o sistema de recomendação quanto o experimento descrito foram implementados[1] em C++, tendo como dependências somente a biblioteca de expressões regulares do Boost[2], a biblioteca C++ de acesso ao PostgreSQL pqxx[3] e a biblioteca de testes automatizados googletest[4]. .

[1] O sistema implementado está disponível em github.com/casimiro/naive-tweet-recommender
[2] www.boost.org
[3] www.pqxx.org
[4] code.google.com/p/googletest

4.2 Perfil de usuário

No sistema implementado, o perfil de interesses do usuário é representado por um conjunto de *hashtags* presentes em suas publicações. Para cada *hashtag* deste conjunto, existe um valor real que representa o nível de interesse do usuário dono do perfil em assuntos relacionados àquela *hashtag*. É importante observar que as *hashtags* exercem o papel de palavras-chave representando tópicos. Por exemplo, a *hashtag* #economia claramente representa um tópico tradicional de notícia. No entanto, outras *hashtags* podem representar o mesmo tópico, como por exemplo a *hashtag* #inflação.

O perfil pode ser representado formalmente pela equação abaixo (RICCI et al., 2010):

$$P(u) = \{(h, w(u,h)) | h \in H, u \in U\} \tag{34}$$

Nesta equação, U representa o conjunto de usuários do sistema, H representa o conjunto de *hashtags*, e $w(u,h)$ é uma função peso que calcula o nível de interesse do usuário u pela *hashtag* h. No sistema implementado, foi utilizada uma função de peso simples que retorna apenas a quantidade de publicações das *hashtags*, de modo que posteriormente esses valores sejam normalizados para que sua soma seja igual a 1.

O perfil dos itens recomendados possui uma estrutura similar ao perfil do usuário, diferindo principalmente pela função peso. Ou seja, o perfil de um *tweet* é também um conjunto ponderado de *hashtags*.

O perfil dos itens recomendados está representado na equação abaixo (RICCI et al., 2010):

$$P(t) = \{(h, w_t(t,h)) | h \in H, t \in T\} \tag{35}$$

Nesta equação, T representa o conjunto de *tweets*, H representa o conjunto de *hashtags* e $w_t(t,h)$ é uma função peso que simplemente retorna a quantidade de vezes que a *hashtag* h aparece no *tweet* t.

O problema da recomendação propriamente dito é abordado como um problema de recuperação de informação onde o perfil do usuário representa uma cadeia de busca e os perfis dos *tweets* representam documentos que precisam ser ordenados de acordo com sua proximidade à *string* de busca. Isso é feito calculando-se a similaridade entre cada *tweet*

candidato e o perfil do usuário, e retornando-se a lista dos candidatos ordenada de acordo com suas respectivas similaridades.

Seja $p(u)$ e $p(t)$ a representação vetorial do perfil de usuário e do perfil de *tweet*, respectivamente, onde a i-ésima dimensão de $p(u)$ e $p(t)$ representa $w(u, h_i)$ e $w(t, h_i)$, respectivamente. Assim, a similaridade entre um perfil de usuário e um perfil de *tweet* é calculada pela função coseno similaridade descrita na equação 12 do capítulo 3.

4.3 Conjunto de dados

O conjunto de dados utilizado para treinar e avaliar o sistema de recomendação foi extraído da plataforma Twitter. O Twitter possui algumas vantagens em relação a outras redes sociais: a plataforma permite a marcação de usuários em publicações, a republicação de *tweets*, a publicação de *links* e uma estrutura social relativamente simples. Por outro lado, o limite de 140 caracteres das publicações no Twitter representa um desafio considerável para a modelagem do perfil do usuário com base apenas nesta informação

Inicialmente, tentou-se utilizar o conjunto de dados disponibilizado em (ABEL et al., 2011a) para treinar e testar o sistema implementado. Entretanto, foram encontrados problemas técnicos na base de dados. O conjunto de dados disponibilizado em (ABEL et al., 2011a) é composto por *tweets* e por *links* de artigos de notícia publicados em *RSS Feeds* de grandes mídias como New York Times[5]. Os *links* para notícias eram os itens a serem recomendados para usuários do Twitter, e o consumo do item de recomendação era verificado pela existência de um *tweet* contendo um link apontando para a notícia que o item recomendado apontara. Entretanto, muitos *links* presentes na base não eram mais válidos, e com isso não era possível verificar o consumo de boa parte dos artigos de notícia na base.

Por este motivo, foi desenvolvido um programa para coletar automaticamente os dados necessários para treinamento e avaliação do sistema de recomendação. O programa foi projetado visando obter um conjunto de dados contendo apenas dados do Twitter, evitando-se assim dependência de dados externos durante o processo de treinamento e avaliação. Além disso, procurou-se obter uma base de dados contendo dados de relacionamento entre usuários da rede de modo que esses dados pudessem ser reaproveitados posteriormente, com o intuito de construir perfis de usuários mais sofisticados.

[5] www.nytimes.com

O algoritmo projetado[6] foi implementado em Python utilizando-se uma abordagem *multi thread* para agilizar o processo de coleta. O processo de coleta constituiu em fazer chamadas à API do Twitter[7], contornando as limitações de uso da API (e.g., quantidade máxima de chamadas por hora) ativando e desativando VPNs por meio de chamadas ao programa vpnc[8]. Os dados foram armazenados em um banco de dados PostgreSQL[9]

O algoritmo de coleta recuperou publicações, menções a usuários, e conexões de 412 usuários. Esses usuários foram sorteados de uma amostra de usuários que satisfaziam os seguintes requisitos: ter o perfil público, seguir dois grandes perfis de divulgação de notícia (Estado de São Paulo[10] e Folha de São Paulo[11]), ter publicações anteriores a 01/01/2013 e ter publicações posteriores a 01/05/2013. Esses critérios foram escolhidos com o objetivo de selecionar usuários com publicações durante o período utilizado para treinamento e validação e que também utilizassem o Twitter com a finalidade de obter artigos de notícia. Apesar destes critérios definirem um público majoritariamente brasileiro, e portanto coletar publicações em língua portuguesa, é possível que ocorra no conjunto *tweets* de línguas estrangeiras. Como o modelo explorado neste estudo preliminar baseia-se em *hashtags*, e não nas palavras das publicações, isto não é, em princípio, um problema significativo.

Para cada um dos 412 usuários, o algoritmo recuperou, além das publicações, todas conexões com usuários seguidos. Posteriormente, foram recuperadas também as publicações de todos usuários seguidos. Como resultado, foi obtida uma sub rede social completa desses 412 usuários, contendo todos os *tweets* que estes usuários teriam acesso no período coletado (de 01/01/2013 a 01/05/2013). Essa é uma característica importante do conjunto de dados, pois assegura que, durante o período de avaliação do sistema, há disponível o conjunto quase completo de candidatos para um dado usuário. O conjunto só não é realmente completo porque o usuário pode retuitar *tweets* de publicidade ou *tweets* aleatórios encontrados, e.g., em buscas realizadas na plataforma.

No Twitter, os usuários podem mencionar outros usuários em suas publicações. Essas informações foram também coletadas e podem ser utilizadas para modelar o relacionamento entre usuários. Ou seja, pode-se construir um perfil de interesses utilizando-se não apenas

[6] A implementação deste procedimento está disponível em github.com/casimiro/master-preproc
[7] dev.twitter.com
[8] www.unix-ag.uni-kl.de/ massar/vpnc
[9] www.postgresql.org
[10] www.estadao.com.br
[11] www.folha.com.br

as publicações do usuário, mas também explorando-se aspectos sociais como utilizar publicações daqueles usuários que mais se relacionam com este.

A coleta resultou em um conjunto de dados com aproximadamente 48 mil usuários, 97 milhões de *tweets* e 60 milhões de menções de usuários.

Finalmente, o conjunto de dados foi anonimizado substituindo-se o código de identificação dos usuários por valores únicos e aleatórios.

4.4 Avaliação

O sistema implementado foi avaliado em um experimento projetado para reproduzir o que seria seu uso provável em produção. Um sistema de recomendação de conteúdo para o Twitter tipicamente terá de reordenar a *timeline* do usuário sempre que o usuário visitar sua *timeline*. Dessa forma, deveríamos idealmente gerar recomendações para cada vez que um determinado usuário acessou o Twiter no período de teste, e verificar qual posição da lista de recomendação os itens retuitados ocuparam. No entanto, o Twitter não disponibiliza a informação de quando um usuário acessa a rede. A alternativa mais comum, e também adotada em estudos como (ABEL et al., 2011a; YAN; LAPATA; LI, 2012), é o uso de *retweets* como indicativo de interesse do usuário. Assim, *retweets* são utilizados na avaliação de sistemas de recomendação no Twitter.

Para o treinamento do sistema, foram utilizados dados publicados até 31-03-2013. Para a avaliação, foi utilizado o período de 01-04-2013 a 01-05-2013. Na prática, os modelos dos usuários foram construídos utilizando-se *tweets* publicados até 01/04/2013, e para cada cada *retweet* publicado por um usuário no período de 01/04/2013 a 01/05/2013 foi gerada uma lista de recomendação.

O sistema implementado foi avaliado com base em três medidas: MRR (*Mean Reciprocal Rank*), S@5 e S@10 (ABEL et al., 2011a) descritas na seção 2.3.4

Como a avaliação foi conduzida em função dos *retweets* dos usuários, para cada *retweet* dentro do período de avaliação é gerada uma lista de recomendações, e cada uma das medidas citadas é calculada. Dessa forma, esses valores podem ser calculados para o sistema como uma média simples conforme descrito abaixo (ABEL et al., 2011a):

$$E = \frac{1}{|U|} \sum_{u}^{|U|} \frac{1}{|R_u|} \sum_{j}^{|R_u|} e_j \tag{36}$$

Nesta equação, e_j representa as medidas de avaliação calculadas para cada *retweet* de um usuário u. $|R_u|$ representa a quantidade de *retweets* do usuário u, $|U|$ representa a quantidade de usuários e E representa as medidas de avaliação calculadas para o sistema.

O sistema foi comparado com um sistema *baseline* que gera recomendações aleatoriamente. Tanto para a abordagem aleatória quanto para abordagem baseada no perfil de *hashtags*, foram experimentadas a janela de candidatos fixa e personalizada.

4.5 Resultados

Apesar de ter sido coletado um conjunto de dados de 412 usuários, apenas 256 usuários possuíam *retweets* no período de avaliação, e portanto apenas esses 256 usuários foram considerados nesta avaliação. Os resultados do experimento estão sumarizados na tabela 1. Os melhores resultados são destacados em negrito.

Tabela 1 – Resultados para recomendações baseadas no perfil de hashtags e recomendações aleatórias

Tipo de Janela	Hashtags			Aleatório		
	MRR	S@5	S@10	MRR	S@5	S@10
Fixa	0,00310	0,00160	0,00562	0,00459	0,00301	0,00304
Personalizada	**0,03018**	**0,04524**	**0,07345**	**0,03114**	**0,03996**	**0,05700**

Com base nos resultados, observa-se em primeiro lugar a baixa qualidade das recomendações baseadas no perfil de *hashtags*. De acordo com os resultados, não é possível afirmar que tais recomendações são melhores que recomendações geradas aleatoriamente. A abordagem baseada em *hashtags* também padece de outro problema: a escassez desses elementos em *tweets*. Dos 256 usuários utilizados no experimento, 90 usários não usaram *hashtags* em seus seus *retweets*, e portanto não poderiam ser recomendados.

Por outro lado, o uso de janelas personalizadas para coletar *tweets* candidatos teve um impacto positivo tanto para recomendações baseadas no perfil de *hashtags* quanto nas recomendações aleatórias. Isso sugere que há uma diferença nos padrões de utilização do Twitter, e que é importante escolher com cuidado a janela de coleta de candidatos para cada usuário. Os resultados deste estudo foram publicados em (CASIMIRO; PARABONI, 2014).

4.6 Discussão

O sistema baseado em (ABEL et al., 2011a) apresenta baixo desempenho na tarefa de recomendação. As recomendações geradas não são melhores que recomendações aleatórias. Além disso, as recomendações baseadas em *hashtags* estão limitadas a *tweets* que contenham esses elementos.

Também de acordo com o experimento realizado, observamos que há uma quantidade expressiva de usuários que não continham *retweets* com *hashtags* no período de avaliação. Dessa forma o sistema não pôde fazer recomendações para aproximadamente 30% dos usuários. Outra dificuldade observada é a de que o sistema depende dessas *hashtags* para aprender os interesses de leitura dos usuários, e ignora quaisquer outras palavras. Esta seria uma possível explicação para a baixa qualidade das recomendações geradas: se é uma tarefa difícil extrair tópicos de interesse de textos breves, parece ainda mais difícil tomar por base estruturas ainda mais esparsas. Essa escassez de informação pode ser observada na proporção de *tweets* da base de dados contendo *hashtags*: 12%, ou aproximadamente 11 milhões dos 97 milhões de *tweets* armazenados.

Com base nestes resultados, nos parece uma estratégia promissora utilizar ferramentas de extração de tópicos a partir de textos breves como *tweets*, como por exemplo, o uso de *Latent Dirichlet Allocation* (BLEI; NG; JORDAN, 2003). Nesta técnica, já empregada em (YAN; LAPATA; LI, 2012), tanto o perfil de interesses do usuário quanto o perfil dos *tweets* seriam representados por vetores ponderados de tópicos, e não mais vetores ponderados de *tweets*. Isso potencialmente resolveria o problema de dependência de *hashtags*.

5 Recomendação de conteúdo com base em aspectos temporais

Este capítulo apresenta o tema principal de que trata esta dissertação - relacionado à recomendação de conteúdo com base em informação temporal -, e os sistemas de recomendação desenvolvidos para teste destes conceitos. A seção 5.1 apresenta uma visão geral de como o fator temporal foi explorado nesta pesquisa no contexto de recomendação de *tweets*. Esta seção motiva o uso da informação de tempo de vida útil de tópico no processo de recomendação de conteúdo no Twitter e discute como a personalização desta informação pode também ser aplicada ao problema. A seção 5.2 apresenta os sistemas que foram implementados, descrevendo suas variações. Estes sistemas serão utilizados para avaliar as hipóteses de pesquisa a serem discutidas no capítulo 6.

5.1 Aspectos temporais da tarefa de recomendação

O presente estudo parte da observação prática de que diferentes tipos de conteúdo possuem tempos de vida útil distintos. Ou seja, é possível que certos *tweets* "envelheçam"mais rápido que outros. Um *tweet* de algum incidente de congestionamento de veículos, por exemplo, tende a ter um prazo de validade muito menor do que um *tweet* de um caso de corrupção na gestão pública. É provável que no dia seguinte após a publicação da *tweet* de trânsito, o mesmo *tweet* já não sirva para nada, o que talvez não ocorra para o caso da notícia política. Com os tópicos extraídos, podemos medir o intervalo médio entre o *tweet* e os *retweets* de cada tópico, e verificar se existem valores estáveis para os intervalos. Além disso, é possível que esta importância relativa do conteúdo seja diferente para cada usuário, ou seja: o tempo de vida útil de um conteúdo pode ser maior ou menor para pessoas diferentes, fenômeno que pode ser observado em outras atividades de interpretação e produção de língua natural (FERREIRA; PARABONI, 2014).

Para efeito deste estudo, definimos os conceitos de vida útil de um tópico da seguinte forma. O tempo de vida útil de tópico, ou VUT, foi estimado utilizando-se o intervalo de tempo entre *retweet* e *tweet*. Nesta pesquisa, como o *retweet* é utilizado como evidência de interesse na informação republicada, o intervalo de tempo entre a publicação de um *tweet* e seu último (ou mais recente) *retweet* indicará por quanto tempo este *tweet* permaneceu "interessante"na rede. Definimos este intervalo como VUp, ou vida útil da publicação.

Existem diversas maneiras possíveis de se computar VUT. Na fase inicial desta pesquisa, este valor era estimado utilizando-se simplesmente o maior VUp de um *tweet* pertencente ao tópico. No entanto, observou-se que alguns *tweets* apresentavam um VUp muito maior que outros do mesmo tópico. Estes episódios aconteciam, tipicamente, em regiões da rede social com baixa atividade. Por exemplo, um usuário pouco ativo na rede - seguindo apenas poucos usuários que publicam com baixa frequência - tende a retuitar publicações antigas, pois não há muita atividade em sua rede, e, por consequência, não há publicações novas para serem retuitadas com frequência.

Este problema era comum a vários tópicos. Com isto, os tópicos apresentavam VUT inadequadamente grande, sem muita distinção entre si. Com base nestas observações, identificou-se a necessidade de elaborar uma estratégia que considerasse o valor de VUp de todos *tweets* pertencentes ao tópico, e não apenas o maior. A estratégia seguida foi estimar a distribuição normal de VUT a partir do valor VUp de seus *tweets*, e então escolher o maior valor dentro de uma porcentagem da área da distribuição. Com esta abordagem, foi possível testar uma gama de porcentagens de área de normal, a fim de encontrar a melhor forma de estimar tempos de vida útil.

Pode-se notar que este processo é similar ao uso da média de VUp mais alguns graus de desvio padrão da distribuição. De fato, escolher uma porcentagem de aproximadamente 70% da área da distribuição é equivalente à média mais uma vez o desvio padrão. O objetivo deste processo foi o de descartar valores VUp muito altos, que podem ser considerados *outliers*.

Foram avaliados tempos de vida útil calculados utilizando-se nove porcentagens distintas de área da distribuição normal: 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80% e 90%. Nota-se que ao utilizar porcentagens de área maiores o tempo de vida útil estimado tende a ser maior. Os tempos de vida útil foram calculados utilizando-se apenas os *tweets* publicados no período de treinamento. Além disso, é importante observar que, quanto menor a porcentagem escolhida, mais próximo será VUT da média simples dos valores de VUp.

5.2 Sistemas desenvolvidos

O objetivo principal do presente estudo é demonstrar o uso de aspectos temporais na tarefa de recomendação de conteúdo no Twitter. De forma mais específica, pretende-se

demonstrar que a recomendação de conteúdo levando em conta informação de vida útil de tópicos melhora a tarefa, e que o conceito de vida útil pode ser customizado para cada usuário.

Além disso, serão analisados também dois aspectos secundários da tarefa de recomendação, os quais podem ter impacto substancial sobres nossos resultados: o uso de diferentes representações de tópicos, e de alternativas para a aprendizagem, ou representação, do perfil do usuário.

No estudo realizado, foram experimentados três sistemas de recomendação básicos. Estes sistemas diferem entre si na forma de representação do tópico, no processo de aprendizagem do perfil de interesses do usuário e na apresentação final dos itens recomendados ao usuário. Cada uma destas dimensões serão descritas adiante.

5.2.1 Representação de tópicos

Foram utilizados dois tipos de tópicos para a representação dos *tweets*: as próprias palavras das publicações e tópicos latentes extraídos por meio do algoritmo LDA (BLEI; NG; JORDAN, 2003), daqui em diante denotados tópicos LDA. A primeira abordagem é a mais simples e consiste apenas em utilizar o conjunto de palavras do *tweet* para representá-lo, sendo que cada palavra denota um tópico de informação. Nesta técnica, que é usualmente chamada de *bag of words*, atribui-se um valor (e.g. por meio de uma função) a cada palavra do texto, que representa o nível de relacionamento da palavra com o texto. A segunda abordagem consiste em aplicar a técnica LDA em um conjunto de *tweets* para extrair um conjunto de tópicos latentes e posteriormente atribuí-los aos *tweets*. O relacionamento entre um tópico e um *tweet* é expresso por um valor que representa o grau de pertinência do *tweet* ao tópico. Em ambas técnicas o *tweet* pode ser representado pela equação seguinte:

$$P(p) = \{(t, w(p,t)) | t \in T, p \in P\} \tag{37}$$

Nesta equação, P representa o conjunto de tweets (ou publicações), T representa o conjunto de tópicos (palavras ou tópicos latentes) e $w(p,t)$ é uma função peso que denota o nível de pertinência do *tweet* p ao tópico t. Na representação por *Bag of Words* (primeira abordagem), a função $w(p,t)$ apenas retorna a frequência da palavra t na publicação p. Na representação por tópicos latentes este valor é atribuído pelo algoritmo LDA.

5.2.2 Aprendizagem do perfil de usuário

O perfil de usuário foi estimado de duas formas distintas descritas a seguir. Na primeira abordagem, aqui denotada NAIVE, o perfil de usuário é representado por um vetor de tópicos ponderado descrito pela seguinte equação:

$$P(u) = \{(t, w(u,t)) | t \in T, u \in U\} \tag{38}$$

Nesta equação T é o conjunto de tópicos (palavras ou tópicos latentes extraídos do córpus), U é o conjunto de usuários do sistema e $w(u,t)$ é uma função peso que denota o nível de interesse pelo tópico t que o usuário u expressa em suas publicações. A função $w(u,t)$ é calculada somando-se os valores de $w(p,t)$ para todo *tweet* p publicado por u:

$$w(u,t) = \sum_{i}^{|P_u|} w(p_i, t) \tag{39}$$

Na equação 39 P_u representa o conjunto de publicações do usuário u. Finalmente, o vetor de tópicos é normalizado de modo que a soma de seus pesos seja igual a 1. Como pode ser observado nas equações 38 e 39, o perfil de interesses do usuário é estimado apenas somando-se o valor dos relacionamentos entre seus *tweets* e os tópicos. Por exemplo, suponha que um dado usuário tenha publicado apenas dois *tweets* p_1 e p_2. p_1 está relacionado aos tópicos t_1 e t_2 pelos valores 0.7 e 0.5, respectivamente. De forma similar, p_2 está relacionado aos tópicos t_1 e t_2 pelos valores 0.3 e 0.5, respectivamente. Com isto, o perfil do usuário será composto pelos tópicos t_1 e t_2 ambos com peso igual a 0.5.

Na segunda abordagem para estimar o perfil de interesses de usuário foi aplicado a ferramenta de aprendizado de máquina SVM (*Support Vector Machines*). O conjunto de *tweets* (publicados entre 01/01/2013 e 01/07/2013) reservados para estimar os perfis de interesse foram novamente divididos entre treinamento (01/01/2013 a 01/06/2013) e teste (01/06/2013 a 01/07/2013). O objetivo do modelo é aprender a prever quando um *tweet* será retuitado. O conjunto de treinamento é composto por instâncias positivas e negativas. As instâncias positivas, rotuladas com o valor 1, são *tweets* publicados pelo usuário, o que inclui *retweets*. As instâncias negativas são *tweets* publicados por usuários seguidos pelo usuário em questão mas não retuitados por este. Como no problema de recomendação é necessário produzir uma lista de recomendações ordenadas por relevância, o SVM foi configurado para resolver um problema de regressão, ao invés de um problema

de classificação. Deste modo, o modelo treinado passa a representar o perfil de interesses do usuário.

5.2.3 Ordenação dos itens recomendados

A última característica variável dos sistemas de recomendação utilizados refere-se ao modo pelo qual os itens de recomendação são ordenados. Ao fornecer uma lista de recomendação o sistema recomendador precisa coletar um conjunto de candidatos e ordená-los seguindo algum critério. Este critério depende de como o perfil do usuário é estimado. Caso o perfil de interesses do usuário seja estimado usando a ferramenta SVM, o critério utilizado para ordenar os itens é a estimativa do rótulo do item realizada pelo SVM. Se o rótulo estimado do item é próximo de 1, é provável que este item seja retuitado. Caso contrário, é provável que este não seja. Com isto, basta ordenar os candidatos de modo que os itens com maior valor de rótulo ocupem o topo da lista.

Caso o perfil do usuário seja o vetor ponderado de tópicos descrito na equação 38 então o critério para ordenar os candidatos é diferente. Neste caso, o critério para a ordenação é a similaridade cosseno calculada entre as representações vetoriais do perfil do usuário e de um candidato. Seja $p(u)$ e $p(p)$ a representação vetorial do perfil do usuário e do perfil de *tweet*, respectivamente, em que a i-ésima dimensão de $p(u)$ e $p(p)$ representa $w(u, t_i)$ e $w(p, t_i)$, respectivamente. A similaridade entre um perfil de usuário e um perfil de *tweet* é então calculada pela função similaridade cosseno descrita na equação 12 do capítulo 3.

5.3 Implementação

Apesar das diferenças descritas anteriormente permitirem a criação de mais de três sistemas de recomendação, o processo de avaliação é computacionalmente oneroso e, portanto, apenas três sistemas foram utilizados.

O primeiro sistema, mais simples, utiliza as próprias palavras dos *tweets* como tópicos e é o sistema *baseline* do experimento. O segundo sistema é baseado em tópicos extraídos por LDA para descrever os *tweets*. Espera-se que os resultados deste sistema sejam superiores aos do primeiro. Finalmente, foi implementado um sistema que utiliza SVM para, a partir de tópicos extraídos por LDA, estimar o perfil do usuário e fazer recomendações.

É esperado que este último sistema seja superior aos dois anteriores. Daqui em diante, estes três sistemas apresentados serão referenciados por RecBow, RecLDA e RecSVM, respectivamente. A tabela 2 sumariza a configuração dos três sistemas considerados[1].

Tabela 2 – Sistemas implementados

Sistema	Tópico	Aprendizagem	Ordenação
RecBOW	Palavras	NAIVE	Similaridade cosseno
RecLDA	Tópicos LDA	NAIVE	Similaridade cosseno
RecSVM	Tópicos LDA	SVM	Rótulo SVM

[1] O sistemas implementados encontram-se disponíveis em github.com/casimiro/master-experiments

6 Avaliação

Este capítulo descreve um experimento para avaliar o impacto do uso do fator temporal VUT (vida útil de tópico) na tarefa de recomendação de conteúdo no Twitter. Aqui são retomadas e aprofundadas as hipóteses de pesquisa a serem investigadas (6.1), os dados de treinamento e teste considerados (6.2), o procedimento de avaliação adotado (6.3) e os resultados obtidos (6.4).

6.1 Hipóteses

O experimento realizado envolveu uma série de comparações entre os resultados obtidos pelos sistemas de recomendação RecBOW (sistema que usa as próprias palavras dos *tweets* como tópicos e a abordagem NAIVE para estimar o perfil do usuário), RecLDA (sistema que usa tópicos LDA e a abordagem NAIVE para estimar o perfil do usuário) e RecSVM (sistema que usa tópicos LDA e a abordagem SVM para estimar o perfil do usuário) descritos na seção 5.2. De forma específica, foram investigadas duas hipóteses principais:

h1 Utilizar a informação de vida útil de tópico em um sistema de recomendação de conteúdo melhora a qualidade da recomendação em relação a uma abordagem padrão que não utiliza este tipo de informação.

h2 Utilizar a informação de vida útil de tópico personalizada por usuário em um sistema de recomendação de conteúdo melhora ainda mais a qualidade da recomendação em relação a uma abordagem que utiliza apenas informações de vida útil de tópico sem personalização.

Além das hipóteses descritas acima, duas outras hipóteses de caráter secundário serão também investigadas:

h3 Utilizar tópicos LDA (seção 5.2) para representar os *tweets* em um sistema de recomendação de conteúdo melhora a qualidade da recomendação em relação a uma abordagem que utilize as próprias palavras dos *tweets* para descrevê-los.

h4 Utilizar máquinas de vetor de suporte para estimar o perfil de interesses de usuário em sistema de recomendação de conteúdo melhora a qualidade da recomendação em relação a uma abordagem que utilize a técnica NAIVE (seção 5.2).

Todas as avaliações são baseadas nas métricas MRR, S@5 e S@10 conforme descrito na seção 2.3.4.

A hipótese h1 será avaliada comparando-se os resultados obtidos pelo sistema RecLDA sem o emprego da informação de VUT, com os resultados do mesmo sistema RecLDA empregando-se informação de VUT. Em todos os casos, espera-se que os resultados do sistema RecLDA com o uso de VUT sejam superiores aos do sistema RecLDA sem o uso de VUT.

A hipótese h2 será avaliada comparando-se os resultados obtidos pelo sistema RecLDA sem o emprego da informação de VUT personalizada por usuário com os resultados do mesmo sistema RecLDA empregando-se informação de VUT personalizada. Em todos os casos, espera-se que os resultados do sistema RecLDA com o uso de VUT personalizada sejam superiores aos do sistema RecLDA sem o uso de VUT personalizada.

A hipótese h3 será avaliada comparando-se os resultados obtidos pelo sistema RecBOW com os resultados do sistema RecLDA. Em todos os casos, espera-se que os resultados do sistema RecLDA sejam superiores aos do sistema RecBOW.

Finalmente, a hipótese h4 será avaliada comparando-se os resultados obtidos pelo sistema RecLDA com os resultados do sistema RecSVM. Em todos os casos, espera-se que os resultados do sistema RecSVM sejam superiores aos do sistema RecLDA.

6.2 Dados

Esta seção descreve o processo utilizado para coletar o conjunto de dados utilizado no experimento. A seção também descreve as características dos dados, o processo de tratamento para remover ruídos - processo fundamental ao se trabalhar com dados do Twitter -, e, finalmente, como os dados foram divididos para as tarefas de treinamento e testes dos modelos.

6.2.1 Processo de coleta

O conjunto de dados utilizado durante os experimentos foi extraído do Twitter. O programa utilizado para realizar a coleta de dados foi o mesmo descrito no capítulo 4. No entanto, é importante observar que o conjunto de dados utilizado na quele estudo preliminar é diferente do conjunto descrito a seguir. A diferença consiste nos requisitos utilizados para selecionar usuários e na quantidade de usuários coletados.

O programa coletou publicações e informações de 5.020 usuários, aqui denotados usuários alvo. Estes usuários foram sorteados seguindo alguns requisitos: serem seguidores dos jornais Folha de São Paulo, Estado de São Paulo ou o Globo; ter ao menos 10 *tweets* publicados entre 01/01/2013 e 01/05/2013 e ao menos um *retweet* publicado entre 01/05/2013 e 01/08/2013. O primeiro critério tem o objetivo de selecionar usuários que utilizam o Twitter como meio de consumo de notícias. O segundo e terceiro critério tem a finalidade selecionar usuários com dados nos período de treinamento e teste, respectivamente.

Para cada um dos usuários alvo o algoritmo recuperou, além de suas publicações, todas conexões com usuários seguidos. Finalmente, também foram coletadas as publicações desses usuários seguidos.

O processo de coleta gerou um conjunto de dados com 414 milhões de *tweets*, publicados no período de 01/01/2013 a 01/08/2013, e 321 mil usuários, dos quais 5020 são usuários alvo. Os usuários alvo receberão recomendações, enquanto os outros usuários serão apenas utilizados como fontes de informação.

6.2.2 Tratamento dos dados

O algoritmo LDA é muito sensível à qualidade do conteúdo dos documentos que este utilizará para extrair e atribuir tópicos. Aplicar o LDA em um conjunto de documentos contendo ruídos (e.g., como números ou caracteres que não sejam letras) pode fazer com que os tópicos extraídos tenham pouco ou nenhum significado. De fato, como será explicado mais adiante, pode ser necessário remover mais do que apenas símbolos indesejados. Como no caso do Twitter as publicações possuem no máximo 140 caracteres e muitos usuários optam por uma escrita informal, ou mesmo incorreta, muitos *tweets* acabam não tendo conteúdo algum, o que evidentemente se torna um problema para o algoritmo de extração

de tópicos latentes. É muito comum ver publicações contendo apenas risadas (e.g., "hehehe", "hahaha", "kkk"etc), URLs ou *emoticons* (e.g., "=)", "=("etc).

Devido à grande importância do tratamento do conteúdo de *tweets* para a execução bem sucedida do algoritmo LDA e à complexidade do processo de tratamento, foi escrito um programa[1] em C++ para tratar de forma simplificada todo o conjunto de publicações. O programa faz uso extensivo de expressões regulares para remover padrões de ruídos como risadas, descritas anteriormente. De forma específica, para cada *tweet* do conjunto de dados são executados os passos descritos a seguir para remoção de conteúdo indesejado.

1. Remove símbolos do *tweet*.
2. Remove acentos do *tweet*.
3. Substitui letras maiúsculas por minúsculas.
4. Remove *emoticons* do *tweet*.
5. Remove risadas do *tweet*.
6. Remove URLs do *tweet*.
7. Remove menções a usuários do *tweet*.
8. Extrai nomes e palavras desconhecidas do *tweet*.
9. Descarta *tweet* se considerado estrangeiro.
10. Descarta *tweet* se considerado pequeno.

O procedimento recebe como entrada o conteúdo do *tweet*. O primeiro passo do algoritmo (passo 1) consiste em remover caracteres que não são letras (e.g., algarismos, '(', ')', '=' etc). No segundo passo (passo 2) remove-se a acentuação do texto. Todos caracteres acentuados são substituídos por suas respectivas versões sem acento. Por exemplo, 'ç' é substituído por 'c', 'à' por 'a' e assim por diante. Após remover os acentos, letras maiúsculas são substituídas por suas respectivas minúsculas (passo 3). A seguir *emoticons* são removidos do texto (passo 4), assim como expressões de risadas (passo 5), URLs (passo 6) e menções a outros usuários (e.g., @estadao). O próximo passo (passo 8) consiste em extrair nomes e palavras desconhecidas do do conteúdo restante. Neste passo, foram removidos verbos, adjetivos, advérbios etc, preservando apenas substantivos e palavras desconhecidas segundo o dicionário DELAF (MUNIZ, 2004). Palavras desconhecidas foram preservadas pois podem representar nomes próprios, além disso, é importante ressaltar que os substantivos foram extraídos em suas formas canônicas.

[1] Implementação disponível em github.com/casimiro/master-preproc

Conteúdos estrangeiros também afetam negativamente a execução do LDA. Assim, *tweets* considerados estrangeiros são descartados (passo 9). Neste trabalho, assim como em (YAN; LAPATA; LI, 2012), foi utilizado a taxa de palavras desconhecidas para determinar se um *tweet* era considerado de língua portuguesa ou não. As palavras eram tidas como conhecidas ou desconhecidas de acordo com o dicionário DELAF (MUNIZ, 2004). Mediante inspeção manual, verificou-se que *tweets* escritos em língua portuguesa apresentavam uma taxa de palavras desconhecidas muito inferior a 0,5, enquanto que a taxa de publicações estrangeiras eram muito superiores a 0,5. Deste modo, optou-se por descartar publicações com taxa de palavras desconhecidas superior a 0,5.

Finalmente, *tweets* considerados pequenos são descartados (passo 10). Após testar algumas quantidades mínimas de palavras, optou-se por escolher uma quantidade mínima de palavras igual a 4. Deste modo, todo *tweet* com menos de 4 palavras foi descartado.

Depois do processamento o córpus possui aproximadamente 198 milhões de *tweets*. Mais da metade dos *tweets* coletados foi descartada, considerada ruído de acordo com as regras descritas acima. É importante também notar que a etapa de extração de tópicos do LDA foi executada com um córpus agrupado por usuário. Ou seja, neste córpus, ao invés de cada documento ser um *tweet*, cada documento é composto pelo conjunto de *tweets* de um determinado usuário. Em testes preliminares, observamos que esta técnica contribuiu para uma melhora significativa da qualidade dos tópicos extraídos.

6.3 Procedimento

Foi implementado um programa[2] em C++ responsável por treinar e avaliar os sistemas de recomendação descritos anteriormente. Ao ser executado, o programa recebe um conjunto de parâmetros que dizem respeito às configurações do experimento a ser realizado. Estas configurações especificam o sistema de recomendação, o conjunto de vida útil de tópicos (10%, 20%, 30%, 40%, 50%, 60%, 70%, 80% ou 90%), o período de treinamento e o período de avaliação a serem utilizados durante a avaliação.

Com relação à informação de vida útil de tópico, esta foi utilizada após a coleta de candidatos. Ou seja, após coletar os candidatos e antes de ordená-los, foram eliminados todos candidatos considerados ultrapassados. Em um dado instante t, um candidato é considerado ultrapassado caso o período de tempo entre o momento de sua publicação

[2] Implementação disponível em github.com/casimiro/master-experiments

e o instante t seja superior ao tempo de vida útil de qualquer um de seus tópicos. Por exemplo, suponha que em um determinado instante um *tweet* p_i tenha sido publicado há 20 horas e que ele pertença a dois tópicos tp_1 e tp_2. Além disso, suponha que o tempo de vida útil dos tópicos seja 30 e 15 horas, respectivamente. Desta forma, o candidato p_i é considerado ultrapassado, porque o tempo de vida útil do tópico tp_2 é de 15 horas e o candidato foi publicado há 20 horas.

Durante sua execução, o programa estima o perfil de interesses de cada usuário utilizando dados publicados no período de treinamento, e realiza recomendações utilizando dados do período de avaliação. Concretamente, para cada *retweet* publicado pelo usuário no período de avaliação, é gerada uma lista de recomendação. Cada lista de recomendação é então avaliada de acordo com a posição do respectivo *retweet* na lista, calculando-se o valor das métricas de avaliação descritas anteriormente. A avaliação final do sistema corresponde à média das métricas calculadas individualmente para cada lista de recomendação.

O programa foi implementado seguindo a metodologia de desenvolvimento orientado a testes e possui 29 testes unitários e de aceitação que verificam o funcionamento esperado do sistema. Os 29 testes cobrem 100% do código fonte do experimento.

O modelo de tópicos aqui discutido é evidentemente simples, com possível impacto negativo no desempenho do sistema de recomendação. Dados os objetivos da pesquisa (conforme seção 6.1), entretanto, acreditamos que este modelo seja suficiente para ilustrar o uso de aspectos temporais nesta tarefa.

6.4 Resultados

As tabelas 3 e 4 sumarizam os resultados dos experimentos. Em ambas tabelas, as colunas superiores dizem respeito ao sistema utilizado – RecBOW, RecLDA ou RecSVM –, as colunas internas descrevem a métrica utilizada e as linhas referem-se ao cálculo de vida útil (VU) utilizado. Nas duas tabelas, a primeira linha representa o resultado dos sistemas sem a utilização de informação de vida útil (S/VU). Na tabela 3, as demais linhas representam o cálculo de vida útil não personalizado, enquanto que na tabela 4, as demais colunas representam o cálculo de vida útil personalizado (VUP). Os dados destacados em negrito indicam os melhores resultados de acordo com cada métrica – quanto maior, melhor.

Tabela 3 – Resultados dos sistemas experimentados considerando uma vida útil do tópico única para todos usuários.

VU	RecBOW			RecLDA			RecSVM		
	MRR	S@5	S@10	MRR	S@5	S@10	MRR	S@5	S@10
S/VU	0,00357	0,00179	0,00493	0,00541	0,00438	0,00826	0,00319	0,00186	0,00379
VU10	**0,00362**	**0,00224**	**0,00538**	**0,01027**	**0,00900**	**0,01690**	**0,00677**	**0,00459**	**0,00926**
VU20	**0,00362**	**0,00224**	**0,00538**	0,01026	0,00898	0,01687	0,00676	0,00458	0,00923
VU30	**0,00362**	**0,00224**	**0,00538**	0,01024	0,00896	0,01684	0,00675	0,00457	0,00920
VU40	**0,00362**	**0,00224**	**0,00538**	0,01023	0,00893	0,01683	0,00674	0,00456	0,00918
VU50	**0,00362**	**0,00224**	**0,00538**	0,01022	0,00892	0,01681	0,00672	0,00455	0,00915
VU60	0,00361	**0,00224**	**0,00538**	0,01020	0,00889	0,01679	0,00670	0,00455	0,00913
VU70	0,00359	0,00179	**0,00538**	0,01018	0,00887	0,01677	0,00668	0,00452	0,00907
VU80	0,00359	0,00179	**0,00538**	0,01016	0,00884	0,01674	0,00666	0,00451	0,00904
VU90	0,00359	0,00179	**0,00538**	0,01013	0,00883	0,01670	0,00663	0,00451	0,00899

Tabela 4 – Resultados dos sistemas experimentados considerando uma vida útil do tópico customizada para cada usuário.

VU	RecBOW			RecLDA			RecSVM		
	MRR	S@5	S@10	MRR	S@5	S@10	MRR	S@5	S@10
S/VU	**0,00357**	**0,00179**	**0,00493**	0,00541	0,00438	0,00826	0,00319	0,00186	0,00379
VUP10	0,00260	0,00134	0,00404	**0,03473**	**0,04467**	**0,07581**	**0,01736**	**0,01773**	**0,03576**
VUP20	0,00260	0,00134	0,00404	0,03398	0,04347	0,07398	0,01701	0,01717	0,03482
VUP30	0,00261	0,00134	0,00404	0,03334	0,04234	0,07204	0,01668	0,01669	0,03405
VUP40	0,00261	0,00134	0,00404	0,03255	0,04107	0,07020	0,01637	0,01624	0,03322
VUP50	0,00263	0,00134	0,00404	0,03193	0,04005	0,06828	0,01602	0,01578	0,03236
VUP60	0,00264	0,00134	0,00404	0,03125	0,03895	0,06657	0,01572	0,01534	0,03154
VUP70	0,00265	0,00134	0,00404	0,03051	0,03763	0,06463	0,01542	0,01496	0,03069
VUP80	0,00267	0,00134	0,00404	0,02964	0,03649	0,06278	0,01507	0,01453	0,02966
VUP90	0,00287	0,00134	0,00449	0,02983	0,03648	0,06290	0,01543	0,01468	0,02981

Com relação à hipótese h1 (sobre o uso de informação de vida útil de tópico), os resultados na tabela 3 demonstram que, a despeito da porcentagem de área utilizada no cálculo, todos sistemas apresentaram resultados melhores para todas métricas avaliadas ao utilizarem a referida informação temporal. Ou seja, utilizar informação de vida útil de tópico melhora a qualidade da recomendação em relação a uma abordagem padrão que não utiliza este tipo de informação. Estes resultados confirmam, para todos os casos avaliados, a hipótese h1.

Os resultados dos testes específicos para a hipótese h2 (sobre o uso de informação personalizada de vida útil de tópico) são sumarizados na tabela 5.

De acordo com os resultados da tabela 5, observamos que o sistema RecLDA com uso de informação personalizada de vida útil de tópico apresentou resultados superiores para todas métricas avaliadas, independente da porcentagem de área normal utilizada no cálculo da informação temporal. Ou seja, o uso de informação personalizada de vida útil

Tabela 5 – Resultados referentes à hipótese h2, comparando o sistema RecLDA com vida útil de tópico não personalizada e com vida útil de tópico personalizada.

VU	RecLDA			RecLDAPersonalizado		
	MRR	S@5	S@10	MRR	S@5	S@10
S/VU	0,00541	0,00438	0,00826	0,00541	0,00438	0,00826
VU10	**0,01027**	**0,00900**	**0,01690**	**0,03473**	**0,04467**	**0,07581**
VU20	0,01026	0,00898	0,01687	0,03398	0,04347	0,07398
VU30	0,01024	0,00896	0,01684	0,03334	0,04234	0,07204
VU40	0,01023	0,00893	0,01683	0,03255	0,04107	0,07020
VU50	0,01022	0,00892	0,01681	0,03193	0,04005	0,06828
VU60	0,01020	0,00889	0,01679	0,03125	0,03895	0,06657
VU70	0,01018	0,00887	0,01677	0,03051	0,03763	0,06463
VU80	0,01016	0,00884	0,01674	0,02964	0,03649	0,06278
VU90	0,01013	0,00883	0,01670	0,02983	0,03648	0,06290

de tópico melhora a qualidade da recomendação em relação a uma abordagem que utiliza apenas informação de vida útil de tópico sem personalização. Isso confirma a hipótese h2.

Os resultados para a hipótese h3 (sobre o uso de tópicos LDA para representar *tweets*), são sumarizados na tabela 6.

Tabela 6 – Resultados referentes à hipótese h3, comparando o sistema RecLDA com o sistema RecBOW.

VU	RecBOW			RecLDA		
	MRR	S@5	S@10	MRR	S@5	S@10
S/VU	0,00357	0,00179	0,00493	0,00541	0,00438	0,00826
VU10	**0,00362**	**0,00224**	**0,00538**	**0,01027**	**0,00900**	**0,01690**
VU20	**0,00362**	**0,00224**	**0,00538**	0,01026	0,00898	0,01687
VU30	**0,00362**	**0,00224**	**0,00538**	0,01024	0,00896	0,01684
VU40	**0,00362**	**0,00224**	**0,00538**	0,01023	0,00893	0,01683
VU50	**0,00362**	**0,00224**	**0,00538**	0,01022	0,00892	0,01681
VU60	0,00361	**0,00224**	**0,00538**	0,01020	0,00889	0,01679
VU70	0,00359	0,00179	**0,00538**	0,01018	0,00887	0,01677
VU80	0,00359	0,00179	**0,00538**	0,01016	0,00884	0,01674
VU90	0,00359	0,00179	**0,00538**	0,01013	0,00883	0,01670

De acordo com os resultados da tabela 6, observamos que o sistema RecLDA apresenta melhores resultados que o sistema RecBOW para todas as métricas. Ou seja, utilizar tópicos LDA para representar os *tweets* em um sistema de recomendação de conteúdo melhora a qualidade da recomendação em relação a uma abordagem que utilize as próprias palavras para descrevê-los. Isso confirma a hipótese h3.

Finalmente, os resultados para a hipótese h4 (sobre o uso de máquinas de vetor de suporte) são sumarizados na tabela 7.

Tabela 7 – Resultados referentes à hipótese h4, comparando o sistema RecSVM com o sistema RecLDA.

VU	RecLDA			RecSVM		
	MRR	S@5	S@10	MRR	S@5	S@10
S/VU	0,00541	0,00438	0,00826	0,00319	0,00186	0,00379
VU10	**0,01027**	**0,00900**	**0,01690**	**0,00677**	**0,00459**	**0,00926**
VU20	0,01026	0,00898	0,01687	0,00676	0,00458	0,00923
VU30	0,01024	0,00896	0,01684	0,00675	0,00457	0,00920
VU40	0,01023	0,00893	0,01683	0,00674	0,00456	0,00918
VU50	0,01022	0,00892	0,01681	0,00672	0,00455	0,00915
VU60	0,01020	0,00889	0,01679	0,00670	0,00455	0,00913
VU70	0,01018	0,00887	0,01677	0,00668	0,00452	0,00907
VU80	0,01016	0,00884	0,01674	0,00666	0,00451	0,00904
VU90	0,01013	0,00883	0,01670	0,00663	0,00451	0,00899

De acordo com os resultados da tabela 7, observamos um efeito contrário ao esperado: os resultados do sistema RecLDA superam os resultados do sistema RecSVM. Isto não confirma a hipótese h4.

6.4.1 Discussão

A execução do experimento confirmou nossas duas hipóteses centrais (h1 e h2, sobre o uso de informação de vida útil de tópico e sobre o uso da versão personalizada desta informação), e também uma segunda hipótese secundária (h3, sobre o uso de tópicos LDA para representar os *tweets*). A segunda hipótese secundária (h4, sobre o uso de máquinas de vetor de suporte para estimar o perfil do usuário) não foi confirmada. Destes resultados, as seguintes lições podem ser obtidas:

- O conceito de vida útil de tópicos pode ser explorado para melhoria da qualidade de recomendação de conteúdo (conforme h1).
- A informação da vida útil de tópico pode ser personalizada por usuário, o que possui impacto positivo adicional sobre a recomendação (conforme h2).
- O modelo baseado em tópicos latentes, apesar da forma simplificada como foi computado neste experimento, demonstrou-se superior ao modelo BOW.

7 Conclusão

Este trabalho apresentou um estudo que avalia a aplicação da informação de vida útil de tópico, e sua forma personalizada, na tarefa de recomendação de conteúdo no Twitter. O estudo realizado também explorou dois aspectos secundários inerentes à tarefa de recomendação: o modelo de tópicos utilizado para representar os *tweets* e a abordagem utilizada para estimar o perfil de interesses do usuário. Os resultados apresentados confirmam a hipótese de que considerar a informação de vida útil de tópico implica em melhora na qualidade da recomendação. Os resultados também confirmam a hipótese de que utilizar a informação de vida útil de tópico customizada para cada usuário melhora ainda mais a qualidade da recomendação. Quanto aos aspectos secundários desta pesquisa, verificou-se por meio dos resultados que o uso de tópicos LDA para representar os *tweets* melhora a qualidade da recomendação. Por outro lado, os resultados apontaram que o uso de máquinas de vetor de suporte não necessariamente melhora a qualidade da recomendação.

A principal contribuição da presente pesquisa diz respeito à aplicação da informação de vida útil de tópico, e sua forma personalizada, ao problema de recomendação de conteúdo no Twitter. Quanto à divulgação deste trabalho, foi publicado um artigo sobre o estudo preliminar descrito no capítulo 4 (CASIMIRO; PARABONI, 2014).

Como trabalho futuro, identificamos a necessidade de implementação de um mecanismo mais robusto de representação de documentos, possivelmente com uso de uma ferramenta de reconhecimento de entidades nomeadas (RICCI et al., 2010). Além disso, pode-se explorar o conceito de grupos de usuários computando-se a informação de vida útil de tópico não apenas de forma personalizada mas também por grupos de usuários. Também identificamos a necessidade de um método mais sofisticado para estimar o perfil de interesses do usuário.

Referências[1]

ABEL, F. et al. Analyzing user modeling on twitter for personalized news recommendations. In: *Proceedings of the 19th international conference on User modeling, adaption, and personalization*. Berlin, Heidelberg: Springer-Verlag, 2011. (UMAP '11), p. 1–12. ISBN 978-3-642-22361-7. Disponível em: ⟨http://dl.acm.org/citation.cfm?id=2021855.2021857⟩. Citado 11 vezes nas páginas 27, 28, 29, 30, 31, 33, 42, 43, 46, 48 e 50.

ABEL, F. et al. Semantic enrichment of twitter posts for user profile construction on the social web. In: *Proceedings of the 8th extended semantic web conference on The semanic web: research and applications - Volume Part II*. Berlin, Heidelberg: Springer-Verlag, 2011. (ESWC '11), p. 375–389. ISBN 978-3-642-21063-1. Disponível em: ⟨http://dl.acm.org/citation.cfm?id=2017936.2017967⟩. Citado na página 29.

ARMENTANO, M. G.; GODOY, D.; AMANDI, A. A. Followee recommendation based on text analysis of micro-blogging activity. *Inf. Syst.*, Elsevier Science Ltd., Oxford, UK, UK, v. 38, n. 8, p. 1116–1127, nov. 2013. ISSN 0306-4379. Disponível em: ⟨http://dx.doi.org/10.1016/j.is.2013.05.009⟩. Citado na página 13.

BLEI, D. M. Probabilistic topic models. *Commun. ACM*, ACM, New York, NY, USA, v. 55, n. 4, p. 77–84, abr. 2012. ISSN 0001-0782. Disponível em: ⟨http://doi.acm.org/10.1145/2133806.2133826⟩. Citado na página 21.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *J. Mach. Learn. Res.*, JMLR.org, v. 3, p. 993–1022, mar. 2003. ISSN 1532-4435. Disponível em: ⟨http://dl.acm.org/citation.cfm?id=944919.944937⟩. Citado 5 vezes nas páginas 21, 22, 40, 50 e 53.

BOLSTAD, W. *Introduction to Information Retrieval*. [S.l.]: Wiley-Interscience, 2004. Citado na página 19.

BRIN, S.; PAGE, L. The anatomy of a large-scale hypertextual web search engine. In: *Proceedings of the seventh international conference on World Wide Web 7*. Amsterdam, The Netherlands, The Netherlands: Elsevier Science Publishers B. V., 1998. (WWW7), p. 107–117. Disponível em: ⟨http://dl.acm.org/citation.cfm?id=297805.297827⟩. Citado na página 39.

CASIMIRO, C. R.; PARABONI, I. Temporal aspects of content recommendation on a microblog corpus. *Lecture Notes in Artificial Intelligence*, Springer, v. 8775, p. 189–194, 2014. Citado 2 vezes nas páginas 49 e 66.

CELEBI, H. B.; USKUDARLI, S. Content based microblogger recommendation. In: *Proceedings of the 2012 ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust*. Washington, DC, USA: IEEE Computer Society, 2012. (SOCIALCOM-PASSAT '12), p. 605–610. ISBN 978-0-7695-4848-7. Disponível em: ⟨http://dx.doi.org/10.1109/SocialCom-PASSAT.2012.124⟩. Citado na página 13.

CHEN, J. et al. Short and tweet: experiments on recommending content from information streams. In: *Proceedings of the 28th international conference on Human factors in*

[1] De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

computing systems. New York, NY, USA: ACM, 2010. (CHI '10), p. 1185–1194. ISBN 978-1-60558-929-9. Disponível em: ⟨http://doi.acm.org/10.1145/1753326.1753503⟩. Citado 3 vezes nas páginas 13, 38 e 39.

DUAN, Y. et al. An empirical study on learning to rank of tweets. In: *Proceedings of the 23rd International Conference on Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010. (COLING '10), p. 295–303. Disponível em: ⟨http://dl.acm.org/citation.cfm?id=1873781.1873815⟩. Citado na página 13.

FACEBOOK. *Key Facts - Facebook Newsroom*. 2012. Disponível em: ⟨http://newsroom.fb.com/Key-Facts⟩. Citado na página 13.

FERREIRA, T. C.; PARABONI, I. Referring expression generation: taking speakers' preferences into account. *Lecture Notes in Artificial Intelligence*, Springer, v. 8655, p. 539–546, 2014. Citado na página 51.

GAO, Q. et al. Interweaving trend and user modeling for personalized news recommendation. In: *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*. [S.l.: s.n.], 2011. v. 1, p. 100 –103. Citado 3 vezes nas páginas 31, 32 e 33.

HANNON, J.; BENNETT, M.; SMYTH, B. Recommending twitter users to follow using content and collaborative filtering approaches. In: *Proceedings of the Fourth ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2010. (RecSys '10), p. 199–206. ISBN 978-1-60558-906-0. Disponível em: ⟨http://doi.acm.org/10.1145/1864708.1864746⟩. Citado na página 13.

HOFMANN, T. Probabilistic latent semantic indexing. In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM, 1999. (SIGIR '99), p. 50–57. ISBN 1-58113-096-1. Disponível em: ⟨http://doi.acm.org/10.1145/312624.312649⟩. Citado na página 21.

HONG, L.; DOUMITH, A. S.; DAVISON, B. D. Co-factorization machines: Modeling user interests and predicting individual decisions in twitter. In: *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*. New York, NY, USA: ACM, 2013. (WSDM '13), p. 557–566. ISBN 978-1-4503-1869-3. Disponível em: ⟨http://doi.acm.org/10.1145/2433396.2433467⟩. Citado na página 13.

KIM, D. et al. Trendssummary: A platform for retrieving and summarizing trendy multimedia contents. *Multimedia Tools Appl.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 73, n. 2, p. 857–872, nov. 2014. ISSN 1380-7501. Disponível em: ⟨http://dx.doi.org/10.1007/s11042-013-1547-0⟩. Citado na página 13.

KOTZ, S.; BALAKRISHNAN, N.; JOHNSON, N. L. *Continuous multivariate distributions. Volume 1. , Models and applications*. New York, Chichester, Weinheim: J. Wiley & sons, 2000. (Wiley series in probability and statistics). ISBN 0-471-18387-3. Disponível em: ⟨http://opac.inria.fr/record=b1127772⟩. Citado na página 22.

KYWE, S. M. et al. On recommending hashtags in twitter networks. In: *Proceedings of the 4th International Conference on Social Informatics*. Berlin, Heidelberg: Springer-Verlag, 2012. (SocInfo'12), p. 337–350. ISBN 978-3-642-35385-7. Disponível em: ⟨http://dx.doi.org/10.1007/978-3-642-35386-4_25⟩. Citado na página 13.

LIANG, H. et al. Time-aware topic recommendation based on micro-blogs. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2012. (CIKM '12), p. 1657–1661. ISBN 978-1-4503-1156-4. Disponível em: ⟨http://doi.acm.org/10.1145/2396761.2398492⟩. Citado na página 13.

LIU, J.; DOLAN, P.; PEDERSEN, E. R. Personalized news recommendation based on click behavior. In: *Proceedings of the 15th international conference on Intelligent user interfaces*. New York, NY, USA: ACM, 2010. (IUI '10), p. 31–40. ISBN 978-1-60558-515-4. Disponível em: ⟨http://doi.acm.org/10.1145/1719970.1719976⟩. Citado na página 29.

MANNING, C.; RAGHAVAN, P.; SCHUTZE, H. *Introduction to Information Retrieval*. [S.l.]: Cambridge University Press, 2008. Citado 7 vezes nas páginas 17, 18, 19, 20, 21, 32 e 38.

MEI, Q.; GUO, J.; RADEV, D. Divrank: the interplay of prestige and diversity in information networks. In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2010. (KDD '10), p. 1009–1018. ISBN 978-1-4503-0055-1. Disponível em: ⟨http://doi.acm.org/10.1145/1835804.1835931⟩. Citado na página 40.

MESSENGER, A.; WHITTLE, J. Recommendations Based on User-Generated Comments in Social Media. In: *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*. [S.l.: s.n.], 2011. p. 505–508. Citado 2 vezes nas páginas 35 e 36.

MORALES, G. D. F.; GIONIS, A.; LUCCHESE, C. From chatter to headlines: harnessing the real-time web for personalized news recommendation. In: *Proceedings of the fifth ACM international conference on Web search and data mining*. New York, NY, USA: ACM, 2012. (WSDM '12), p. 153–162. ISBN 978-1-4503-0747-5. Disponível em: ⟨http://doi.acm.org/10.1145/2124295.2124315⟩. Citado 3 vezes nas páginas 36, 37 e 38.

MUNIZ, M. C. M. *A construção de recursos lingüístico-computacionais para o português do Brasil: o projeto de Unitex-PB*. Dissertação (Mestrado) — Instituto de Ciências Matemáticas de São Carlos, USP, 2004. Citado 2 vezes nas páginas 60 e 61.

O'BANION, S.; BIRNBAUM, L.; HAMMOND, K. Social media-driven news personalization. In: *Proceedings of the 4th ACM RecSys workshop on Recommender systems and the social web*. New York, NY, USA: ACM, 2012. (RSWeb '12), p. 45–52. ISBN 978-1-4503-1638-5. Disponível em: ⟨http://doi.acm.org/10.1145/2365934.2365943⟩. Citado 2 vezes nas páginas 33 e 34.

OTSUKA, E.; WALLACE, S. A.; CHIU, D. Design and evaluation of a twitter hashtag recommendation system. In: *Proceedings of the 18th International Database Engineering & Applications Symposium*. New York, NY, USA: ACM, 2014. (IDEAS '14), p. 330–333. ISBN 978-1-4503-2627-8. Disponível em: ⟨http://doi.acm.org/10.1145/2628194.2628238⟩. Citado na página 13.

PARANJPE, D. Learning document aboutness from implicit user feedback and document structure. In: *Proceedings of the 18th ACM conference on Information and knowledge management*. New York, NY, USA: ACM, 2009. (CIKM '09), p. 365–374. ISBN 978-1-60558-512-3. Disponível em: ⟨http://doi.acm.org/10.1145/1645953.1646002⟩. Citado na página 36.

PENNACCHIOTTI, M. et al. Making your interests follow you on twitter. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2012. (CIKM '12), p. 165–174. ISBN 978-1-4503-1156-4. Disponível em: ⟨http://doi.acm.org/10.1145/2396761.2396786⟩. Citado na página 13.

PHELAN, O. et al. On using the real-time web for news recommendation & discovery. In: *Proceedings of the 20th International Conference Companion on World Wide Web*. New York, NY, USA: ACM, 2011. (WWW '11), p. 103–104. ISBN 978-1-4503-0637-9. Disponível em: ⟨http://doi.acm.org/10.1145/1963192.1963245⟩. Citado na página 13.

PORTEOUS, I. et al. Fast collapsed gibbs sampling for latent dirichlet allocation. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2008. (KDD '08), p. 569–577. ISBN 978-1-60558-193-4. Disponível em: ⟨http://doi.acm.org/10.1145/1401890.1401960⟩. Citado na página 22.

PUNIYANI, K. et al. Social links from latent topics in microblogs. In: *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010. (WSA '10), p. 19–20. Disponível em: ⟨http://dl.acm.org/citation.cfm?id=1860667.1860677⟩. Citado na página 13.

RICCI, F. et al. *Recommender Systems Handbook*. [S.l.]: Springer, 2010. Citado 9 vezes nas páginas 13, 14, 23, 24, 25, 26, 36, 45 e 66.

TWITTER. *Twitter turns six*. 2012. Disponível em: ⟨https://blog.twitter.com/2012/twitter-turns-six⟩. Citado na página 13.

WU, J. et al. Trust-aware media recommendation in heterogeneous social networks. *World Wide Web*, Kluwer Academic Publishers, Hingham, MA, USA, v. 18, n. 1, p. 139–157, jan. 2015. ISSN 1386-145X. Disponível em: ⟨http://dx.doi.org/10.1007/s11280-013-0243-3⟩. Citado na página 13.

YAN, R.; LAPATA, M.; LI, X. Tweet recommendation with graph co-ranking. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012. (ACL '12), p. 516–525. Disponível em: ⟨http://dl.acm.org/citation.cfm?id=2390524.2390597⟩. Citado 7 vezes nas páginas 39, 40, 41, 42, 48, 50 e 61.

YI, X. et al. Beyond clicks: Dwell time for personalization. In: *Proceedings of the 8th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2014. (RecSys '14), p. 113–120. ISBN 978-1-4503-2668-1. Disponível em: ⟨http://doi.acm.org/10.1145/2645710.2645724⟩. Citado na página 13.

YU, S. J. The dynamic competitive recommendation algorithm in social network services. *Inf. Sci.*, Elsevier Science Inc., New York, NY, USA, v. 187, p. 1–14, mar. 2012. ISSN 0020-0255. Disponível em: ⟨http://dx.doi.org/10.1016/j.ins.2011.10.020⟩. Citado na página 13.

ZHANG JOSE IRIA, C. B. Z.; CIRAVEGNA, F. A comparative evaluation of term recognition algorithms. In: *Proceedings of the Sixth International Conference on*

Language Resources and Evaluation (LREC'08). Marrakech, Morocco: European Language Resources Association (ELRA), 2008. ISBN 2-9517408-4-0. Http://www.lrec-conf.org/proceedings/lrec2008/. Citado na página 35.