Comparative Analysis of Content-based Personalized Microblog Recommendations [Experiments and Analysis]

Efi Karra Taniskidou
University of California, Irvine
ekarrata@uci.edu

George Papadakis
University of Athens
gpapadis@di.uoa.gr

George Giannakopoulos
NCSR Demokritos
ggianna@iit.demokritos.gr

Manolis Koubarakis
University of Athens
koubarak@di.uoa.gr

ABSTRACT

Microblogging platforms constitute a popular means of real-time communication and information sharing. They involve such a large volume of user-generated content that their users suffer from an information deluge. To address it, numerous recommendation methods have been proposed to organize the posts a user receives according to her interests. The content-based methods typically build a text-based model for every individual user to capture her tastes and then rank the posts in her timeline according to their similarity with that model. Even though content-based methods have attracted lots of interest in the data management community, there is no comprehensive evaluation of the main factors that affect their performance. These are: *(i)* the representation model that converts an unstructured text into a structured representation that elucidates its characteristics, *(ii)* the source of the microblog posts that compose the user models, and *(iii)* the type of user's posting activity. To cover this gap, we systematically examine the performance of 9 state-of-the-art representation models in combination with 13 representation sources and 3 user types over a large, real dataset from Twitter comprising 60 users. We also consider a wide range of 223 plausible configurations for the representation models in order to assess their robustness with respect to their internal parameters. To facilitate the interpretation of our experimental results, we introduce a novel taxonomy of representation models. Our analysis provides novel insights into the main factors determining the performance of content-based recommendation in microblogs.

1 INTRODUCTION

Microblogging platforms enable the instant communication and interaction between people all over the world. They allow their users to post messages in real-time, often carelessly and ungrammatically, through any electronic device, be it a mobile phone or a personal computer. They also allow for explicit connections between users so as to facilitate the dissemination and consumption of information. These characteristics led to the explosive growth of platforms like Twitter (www.twitter.com), Plurk (www.plurk.com), Sina Weibo (www.weibo.com) and Tencent Weibo (http://t.qq.com).

Their popularity has led to an information deluge: the number of messages that are transmitted on a daily basis on Twitter alone has jumped from 35 million tweets in 2010 to over 500 million in 2017 [29]. Inevitably, their users are constantly overwhelmed with information. As we also show in our experiments, this situation cannot be ameliorated by presenting the new messages in chronological order; the relatedness with users' interests is typically more important than the recency of a post. Equally ineffective is the list of trending topics, where the same messages are presented to all users, irrespective of their personal interests.

A more principled solution to information deluge is offered by *Personalized Microblog Recommendation* (PMR). Its goal is to capture users' preferences so as to direct their attention to the messages that better match their personal interests. A plethora of works actually focuses on *Content-based PMR* [6, 13, 15, 31, 40, 41, 45], which typically operates as follows: first, it builds a document model for every individual post in the training set by extracting features from its textual content. Then, it constructs a user model by assembling the document models that capture the user's preferences. Subsequently, it compares the user model to the models of recommendation candidates (documents) with a similarity measure. The resulting similarity scores are used to rank all candidates in descending order, from the highest to the lowest score, thus placing the most relevant ones at the top positions. Finally, the ranked list is presented to the user.

Content-based PMR is a popular problem that has attracted a lot of attention in the data management community [1, 15, 29–31, 55]. However, the experimental results presented in the plethora of relevant works are not directly comparable, due to the different configurations that are used for several important, yet overlooked parameters.

The core parameter is the *representation model* that is used for converting a set of unstructured texts into a structured representation that reveals their characteristics. The available options range from traditional vector space models [41, 65] to topic models [39, 50]. Also crucial is the *representation source*, i.e., the source of the microblog posts that compose user models. Common choices include the user's tweets [36] together with their retweets [17, 23, 41, 56] as well as the posts of followers [31, 50, 65] and followees [15, 31, 39]. Another decisive factor is the posting activity of a user, i.e., whether she is an information producer or seeker [5, 35]. Other parameters include the novel challenges posed by the short, noisy, multilingual content of microblogs as well as the external information that enriches their textual content, e.g., concepts extracted from Wikipedia [41] or the content of a Web page, whose URL is mentioned in a post [1].

Despite their significance, little effort has been allocated on assessing the impact of these parameters on Content-based PMR. To cover this gap, we perform a thorough experimental analysis that investigates the following questions: *Which representation model is the most effective for recommending short, noisy, multilingual microblog posts? Which is the most efficient one? How robust is the performance of each model with respect to its configuration?*

© 2019 Copyright held by the owner/author(s). Published in Proceedings of the 22nd International Conference on Extending Database Technology (EDBT), March 26-29, 2019, ISBN 978-3-89318-081-3 on OpenProceedings.org.

Which representation source yields the best performance? How does the behavior of individual users affect the performance of Content-based MPR? We leave the investigation of external information as a future work, due to the high diversity of proposed approaches, which range from language-specific word embeddings like Glove [49] to self-reported profile information [21].

To investigate the above questions, we focus on Twitter, the most popular microblogging service worldwide, with over 335 million active users per month.[1] We begin with a categorization of the representation sources and the users it involves, based on its special social graph: every user u_1 is allowed to unilaterally follow another user u_2, with u_1 being a *follower* of u_2, and u_2 a *followee* for u_1; if u_2 follows back u_1, the two users are *reciprocally connected.* Then, we list the novel challenges posed by the short, noisy, user-generated tweets in comparison with the long and curated content of traditional documents. We also introduce a taxonomy of representation models that provides insights into their endogenous characteristics. Based on it, we briefly present nine state-of-the-art representation models and apply them to a dataset of 60 real Twitter users (partitioned into three different categories) in combination with 223 parameter configurations, three user types and 13 representation sources. Finally, we discuss the experimental outcomes in detail, interpreting the impact of every parameter on the performance of Content-based PMR.

In short, we make the following contributions:

- We perform the first systematic study for content-based recommendation in microblogging platforms, covering nine representation models, 13 representation sources and three user types. We have publicly released our code along with guidelines for our datasets[2].
- We organize the main representation models according to their functionality in a novel taxonomy with three main categories and two subcategories. In this way, we facilitate the understanding of our experimental results, given that every (sub-)category exhibits different behavior.
- We examine numerous configurations for every representation model, assessing their relative effectiveness, robustness and time efficiency. Our conclusions facilitate their fine-tuning and use in real recommender systems.

The rest of the paper is structured as follows: Section 2 provides background knowledge on Twitter and formally defines the recommendation task we are tackling in this work. In Section 3, we present our taxonomy of representation models and describe the state-of-the-art models we consider. We present the setup of our experiments in Section 4 and their results in Section 5. Section 6 discusses relevant works, while Section 7 concludes the paper along with directions for future work.

2 PRELIMINARIES

Representation Sources. We consider five sources of tweets for modeling the preferences of a Twitter user, u:

(i) The past *retweets* of u, $R(u)$, which are the tweets she has received from her followees and has reposted herself. Apparently, their subjects have captured u's attention so intensely that she decided to share them with her followers.

(ii) All past *tweets* of u except her retweets, $T(u)$. They enclose themes u is interested in chatting or in informing her followers.

(iii) All (re)tweets of u's followees, $E(u) = \bigcup_{u_i \in e(u)}(R(u_i) \cup T(u_i))$, where $e(u) = \{u_1, \ldots, u_k\}$ is the set of her followees. $E(u)$ models a user as an *information seeker*, who actively and explicitly follows accounts providing interesting information [15, 31, 39].

(iv) All (re)tweets of u's followers, $F(u) = \bigcup_{u_i \in f(u)}(R(u_i) \cup T(u_i))$, where $f(u) = \{u_1, \ldots, u_m\}$ stands for the set of her followers. Given that they have actively decided to follow u, due to the interest they find in her posts, $F(u)$ models u as an *information producer* [31, 50, 65].

(v) All (re)tweets of u's reciprocal connections, $C(u)$=$E(u)$∩$F(u)$. Unlike the unilateral following relationship in Twitter, reciprocal connections may indicate users with very high affinity, thus providing valuable information for user modeling.

Note that all these *atomic* representation sources are complementary, as they cover different aspects of the activity of a particular user and her network. For this reason, $T(u)$ is typically combined with $R(u)$ [17, 23, 41, 56], with only rare exceptions like [36], which considers $T(u)$ in isolation. In this work, we consider not only $T(u) \cup R(u)$ (TR for short), but also the seven remaining pairwise combinations, which give rise to the following *composite* representation sources: $T(u) \cup E(u)$, $R(u) \cup E(u)$, $E(u) \cup F(u)$, $T(u) \cup F(u)$, $R(u) \cup F(u)$, $T(u) \cup C(u)$, and $R(u) \cup C(u)$. For simplicity, we denote them by TE, RE, EF, TF, RF, TC and RC, respectively.

Twitter Challenges. Tweets have some special characteristics that distinguish them from other conventional domains and pose major challenges to representation models [17, 44, 48].

(C1) Sparsity. Tweets are short, comprising up to 280 characters for most languages, except for Chinese, Korean and Japanese, where the length limit is 140 characters. As a result, they lack sufficient content for user and document modeling.

(C2) Noise. The real-time nature of Twitter forces users to tweet quickly, without taking into account the frequent grammar errors and misspellings; these are corrected in subsequent tweets.

(C3) Multilingualism. The global popularity of Twitter has led to a high diversity in tweet languages. This renders inapplicable most language-specific pre-processing techniques, such as stemming and lemmatization. Even tokenization becomes difficult: unlike the European ones, many Asian languages, such as Chinese, Japanese, and Korean, do not use spaces or other punctuation in order to distinguish consecutive words.

(C4) Non-standard language. Tweets offer an everyday informal communication, which is unstructured, ungrammatical or simply written in slang; words are abbreviated (e.g., "gn" instead of "goodnight"), or contain emoticons, such as :), hashtags like #edbt and emphatic lengthening (e.g., "yeeees" instead of "yes").

We consider the above challenges when discussing the outcomes of our experimental analysis in Section 5.

User Categories. Twitter users are typically classified into three categories [5, 35]: *(i) Information Producers* (IP) are those users who tweet and retweet more frequently than they receive updates from their followees, (ii) *Information Seekers* (IS) are those users who are less active compared to their followees, and (iii) *Balanced Users* (BU) are those exhibiting a symmetry between the received and the posted messages.

To quantify these categories, we use the ratio of *outgoing* to *incoming tweets*. For a particular user u, the former involve the tweets and retweets she posts, $R(u) \cup T(u)$, while the latter comprise the tweets and retweets of her followees, $E(u)$. Dividing the outgoing with the incoming tweets, we get the *posting ratio*. Apparently, BU is the set of users with a posting ratio close to 1, i.e., $|R(u) \cup T(u)| \simeq |E(u)|$. To ensure significantly different behavior for the other two categories, we define IP as the set of users with a posting ratio higher than 2, thus indicating that they

[1] https://en.wikipedia.org/wiki/Twitter, last accessed on 14 January 2019.
[2] See https://github.com/efikarra/text-models-twitter for more details.

post at least twice as many tweets as they receive. Symmetrically, we define IS as the set of users with a posting ratio lower than 0.5, receiving at least twice as many tweets as those they publish.

Problem Definition. The task of Content-based PMR aims to distinguish a user's incoming messages, R∪T, into *irrelevant* and *relevant* ones. A common assumption in the literature [17, 33, 41, 56], which allows for large-scale evaluations, is that the relevant messages are those that are retweeted by the user that receives them, an action that implicitly indicates her interests – intuitively, a user forwards a tweet to her followers after carefully reading it and appreciating its content.

In this context, Content-based PMR is usually addressed via a *ranking-based recommendation algorithm* [11, 17, 20], which aims to rank relevant posts higher than the irrelevant ones. More formally, let D denote a set of documents, U a set of users and M the representation space, which is common for both users and microblog posts (e.g., M could be the real vector space). Given a user model $UM : U \longrightarrow M$ and a document model $DM : D \longrightarrow M$, we define this type of recommendation algorithms as follows:

Definition 2.1. **Content-based Personalized Microblog Recommendation** learns a ranking model $RM : M \times M \rightarrow \mathbb{R}$, which, given a user u and a set of testing posts $D_{test}(u) = \{d_1, \ldots, d_k\} \subseteq E(u)$, calculates the ranking scores $RM(UM(u), DM(d_i)), \forall i \in \{1 \ldots k\}$, and returns a list with $D_{test}(u)$ ranked in decreasing score.

We consider a user u as the set of documents that stem from a particular representation source s, i.e., $s(u)$, and we build a different $UM_s(u)$ for each s. Given a set of users U along with a representation source $s = \{s(u) : u \in U\}$ and a set of labeled data $D^s_{tr} = \{D^s_{tr}(u) : u \in U\}$, where $D^s_{tr}(u) = \{(d_i, l_i), d_i \in s(u), l_i \in L\}$ and $D^s_{tr}(u) \cap D_{test}(u) = \emptyset$, the recommendation algorithm is trained as follows for each individual source s: (i) for each $u \in U$, we learn $UM_s(u)$ and $DM(d_i)$ for each $d_i \in D^s_{tr}(u)$, and (ii) we train RM on $\bigcup_{u \in U}\{(UM_s(u), DM(d_i)) : i \in \{1, \ldots |D^s_{tr}(u)|\}$.

As a recommendation algorithm, we employ the one commonly used in the literature for ranking-based PMR [15, 31, 40]: in essence, RM is a similarity function and $RM(UM(u), DM(d_i))$ measures the similarity of document model d_i and user model u.

3 REPRESENTATION MODELS

3.1 Taxonomy

We first present a taxonomy of the state-of-the-art representation models based on their internal functionality, i.e., the way they handle the order of n-grams. Figure 1 presents an overview of their relations, with every edge **A** ⟶ **B** indicating that model B shares the same n-grams with A, but uses them in a different way when representing users and documents. Based on these relations, we identify three main categories of models:

(i) **Context-agnostic models** disregard the order of n-grams that appear in a document, when building its representation. E.g., the token-based models of this category yield the same representation for the phrases "Bob sues Jim" and "Jim sues Bob".

(ii) **Local context-aware models** take into account the relative order of characters or tokens in the n-grams that lie at the core of their document representations. Yet, they lose contextual information, as they ignore the ordering of n-grams themselves. Continuing our example, the token-based models of this category are able to distinguish the bigram "Bob sues" from "sues Bob", but cannot capture the bigrams that precede or follow it.

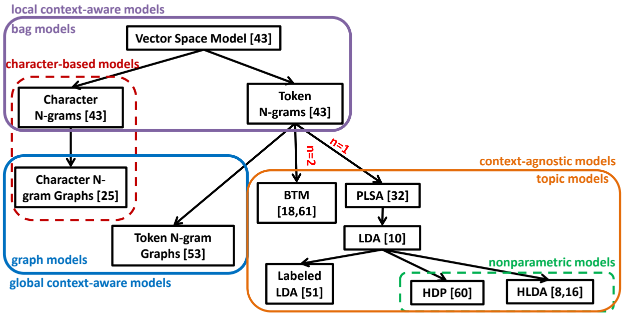

Figure 1: Taxonomy of representation models.

(iii) **Global context-aware models** incorporate into their representations both the relative ordering of tokens or characters in an n-gram and the overall ordering between n-grams in a document. Continuing our example, models of this category distinguish "Bob sues" from "sues Bob" and know that the former is followed by "sues Jim", while the latter is preceded by "Jim sues".

The first category comprises the *topic models* [9]. These are representation models that uncover the latent semantic structure of texts by determining the topics they talk about. The topics and their proportions in a document are considered as the hidden structure of a topic model, which can be discovered by exclusively analyzing the observed data, i.e., the individual tokens (words) in the original text. In general, they regard each document as a mixture of multiple topics, where each topic constitutes a set of co-occurring words. As a result, topic models are typically described in terms of probabilistic modeling, i.e., as generative processes that produce documents, words and users from distributions over the inferred topics [7].

A subcategory of context-agnostic (topic) models pertains to *nonparametric models* [16, 60], which adapt their representations to the structure of the training data. They allow for an unbounded number of parameters that grows with the size of the training data, whereas the parametric models are restricted to a parameter space of fixed size. For example, the nonparametric topic models assume that the number of topics is a-priori unknown, but can be inferred from the documents themselves, while the parametric ones typically receive a fixed number of topics as an input parameter before training.

The category of local context-aware models encompasses the bag models [43], which impose a strict order within n-grams: every n-gram is formed by a specific sequence of characters or tokens and, thus, two n-grams with different sequences are different, even if they involve the same characters or tokens; for example, the bigrams "ab" and "ba" are treated as different. The only exception is the token-based vector model with n=1, which essentially considers individual words; its context-agnostic functionality actually lies at the core of most topic models.

Finally, the category of global context-aware models includes the n-gram graph models, which represent every document as a graph in a language-agnostic way [25]: every node corresponds to an n-gram and edges connect pairs of n-grams co-located within a window of size n, with their weights indicating the co-occurrence frequency. These weighted edges allow graph models to capture global context, going beyond the local context of the bag models that use the same n-grams. Recent works suggest that the graph models outperform their bag counterparts in various tasks [42], which range from Information Retrieval [53] to Document Classification [48].

It should be stressed at this point that the bag and graph models share the second subcategory of our taxonomy: the *character-based models*. These operate at a finer granularity than their token-based counterparts, thus being more robust to noise [48]. For example, consider the words "tweet" and "twete", where the second one is misspelled; they are considered different in all types of token-based models, but for character bi-gram models, they match in three out of four bigrams. In this way, the character-based models capture more precisely the actual similarity between noisy documents.

In the following, we collectively refer to local and global context-aware models as *context-based models*.

3.2 State-of-the-art Models

We now elaborate on the main representation models that are widely used in the literature. To describe topic models, we use the common notation that is outlined in Table 1, while their generative processes are illustrated in Figures 2(i)-(vi) using *plate diagrams*: shaded nodes correspond to the observed variables, the unshaded ones to the hidden variables, the arrows connect conditionally dependent variables, and finally, the plates denote repeated sampling for the enclosed variables for as many times as the number in the right bottom corner.

Bag Models [43]. There are two types of n-grams, the character and the token ones. These give rise to two types of bag models: the *character n-grams model* (**CN**) and the *token n-grams model* (**TN**). Collectively, they are called *bag* or *vector space models*, because they model a document d_i as a vector with one dimension for every distinct n-gram in a corpus of documents D: $DM(d_i) = (w_{i1}, \ldots, w_{im})$, where m stands for the *dimensionality* of D (i.e., the number of distinct n-grams in it), while w_{ij} is the weight of the j^{th} dimension that quantifies the importance of the corresponding n-gram for d_i.

The most common weighting schemes are:

(i) Boolean Frequency (**BF**) assigns binary weights that indicate the absence or presence of the corresponding n-gram in d_i. More formally, $BF(t_j, d_i)$=1 if the n-gram t_j of the j^{th} dimension appears in document d_i, and 0 otherwise.

(ii) Term Frequency (**TF**) sets weights in proportion to the number of times the corresponding n-grams appear in document d_i. More formally, $TF(t_j, d_i)$=f_j/N_{d_i}, where f_j stands for the occurrence frequency of t_j in d_i, while N_{d_i} is the number of n-grams in d_i, normalizing TF so as to mitigate the effect of different document lengths on the weights.

(iii)Term Frequency-Inverse Document Frequency (**TF-IDF**) discounts the TF weight for the most common tokens in the entire corpus D, as they typically correspond to noise (i.e., stop words). Formally, $TF\text{-}IDF(t_j, d_i) = TF(t_j, d_i) \cdot IDF(t_j)$, where $IDF(t_j)$ is the inverse document frequency of the n-gram t_j, i.e., $IDF(t_j) = \log |D|/(|\{d_k \in D : t_j \in d_k\}| + 1)$. In this way, high weights are given to n-grams with high frequency in d_i, but low frequency in D.

To construct the bag model for a specific user u, we aggregate the vectors corresponding to the documents that capture u's interests. The end result is a weighted vector $(a(w_1), ..., a(w_m))$, where $a(w_j)$ is the *aggregation function* that calculates the importance of the j^{th} dimension for u.

The main aggregation functions are: *(i)* the sum of weights, i.e., $a(w_j) = \sum_{d_i \in D} w_{ij}$. *(ii)* the centroid of unit (normalized) document vectors, i.e., $a(w_j) = \frac{1}{|D|} \cdot \sum_{d_i \in D} \frac{w_{ij}}{||DM(d_i)||}$, where $||DM(d_i)||$ is the magnitude of $DM(d_i)$. *(iii)* the Rocchio algorithm, i.e., $a(w_j) = \alpha/|D^p| \cdot \sum_{d_i \in D^p} w_{ij}/||DM(d_i)|| - \beta/|D^n| \cdot \sum_{d_i \in D^n} w_{ij}/||DM(d_i)||$, where D^p and D^n are the sets of positive (relevant) and negative (irrelevant) documents in D, respectively, while $\alpha, \beta \in [0, 1]$ control the relative importance of positive and negative examples, respectively, such that $\alpha + \beta = 1.0$ [43].

Table 1: The notation describing topic models.

Symbol	Meaning
D	the corpus of the input documents
$\lvert D\rvert$	the number of input documents
d	an individual document in D
N_d	the number of words in d, i.e., the *document length*
U	the set of users
u	an individual user in U
$N_{d,u}$	the number of words in document d of user u
D_u	the documents posted by a user u
$\lvert D_u\rvert$	the number of documents posted by user u
V	the *vocabulary* of D (i.e., the set of words it includes)
$\lvert V\rvert$	the number of distinct words in D
w	an individual word in V
$w_{d,n}$	the word at position n in d
Z	the set of latent topics
$\lvert Z\rvert$	the number of latent topics
z	an individual topic in Z
$z_{d,n}$	the topic assigned to the word at position n in document d
θ_d	the multinomial distribution of document d over Z, $\{P(z \mid d)\}_{z \in Z}$
$\theta_{d,z}$	the probability that topic z appears in document d, $P(z \mid d)$
ϕ_z	the multinomial distribution of topic z over V, $\{P(w \mid z)\}_{w \in V}$
$\phi_{z,w}$	the probability that word w appears in topic z, $P(w \mid z)$
$Dir(\alpha)$	a symmetric Dirichlet distribution parameterized by α

To compare two bag models, $DM(d_i)$ and $DM(d_j)$, one of the following similarity measures is typically used:

(i) Cosine Similarity (**CS**) measures the cosine of the angle of the weighted vectors. Formally, it is equal to their dot product similarity, normalized by the product of their magnitudes: $CS(DM(d_i), DM(d_j)) = \sum_{k=1}^{m} w_{ik} w_{jk}/||DM(d_i)||/||DM(d_j)||$.

(ii) Jaccard Similarity (**JS**) treats the document vectors as sets, with weights higher than (equal to) 0 indicating the presence (absence) of the corresponding n-gram. On this basis, it defines as similarity the ratio between the sizes of set intersection and union: $JS(DM(d_i), DM(d_j))$=$|DM(d_i) \cap DM(d_j)|/|DM(d_i) \cup DM(d_j)|$.

(iii) Generalized Jaccard Similarity (**GJS**) extends JS so that it takes into account the weights associated with every n-gram: $GJS(DM(d_i), DM(d_j))$=$\sum_{k=1}^{m} min(w_{ik}, w_{jk})/\sum_{k=1}^{m} max(w_{ik}, w_{jk})$. Note that for BF weights, GJS is identical with JS.

Graph Models [25, 53]. There are two graph models, one for each type of n-grams, i.e., *token n-gram graphs* (**TNG**) [53] and *character n-gram graphs* (**CNG**) [25]. Both models represent each document d as an undirected graph G_d that contains one vertex for each n-gram in d. An edge connects every pair of vertices/n-grams that co-occur within a window of size n in d. Every edge is weighted according to the co-occurrence frequency of the corresponding n-grams. Thus, the graphs incorporate *contextual information* in the form of n-grams' closeness.

To construct the model for a user u, we merge the graphs of the documents representing u's interests using the update operator, which is described in [26]. To compare graph models, we can use the following graph similarity measures [25]: *(i) Containment Similarity* (**CoS**) estimates the number of edges shared by two graph models, G_i and G_j, regardless of the corresponding weights (i.e., it merely estimates the portion of common n-grams in the original texts). Formally: $CoS(G_i, G_j) = \sum_{e \in G_i} \mu(e, G_j)/min(|G_i|, |G_j|)$, where $|G|$ is the size of graph G, and $\mu(e, G) = 1$ if $e \in G$, or 0 otherwise. *(ii) Value Similarity* (**VS**) extends CoS by considering the weights of common edges. Formally, using w_e^k for the weight of edge e in G_k:

$VS(G_i, G_j) = \sum_{e\in(G_i\cap G_j)} \frac{min(w_e^i, w_e^j)}{max(w_e^i, w_e^j)\cdot max(|G_i|,|G_j|)}$. *(iii) Normalized Value Similarity* (**NS**) extends VS by mitigating the impact of imbalanced graphs, i.e., the cases where the comparison between a large graph with a much smaller one yields similarities close to 0: $NS(G_i, G_j) = \sum_{e\in(G_i\cap G_j)} min(w_e^i, w_e^j)/max(w_e^i, w_e^j)/min(|G_i|, |G_j|)$.

Probabilistic Latent Semantic Analysis (PLSA) [32]. This model assigns a topic z to every observed word w in a document d. Thus, every document is modeled as a distribution over multiple topics, assuming that the observed variables w and d are conditionally independent given the unobserved topic z: $P(w|d, z) = P(w|z)$. For an observed pair (w, d), the joint probability distribution is: $P(w, d) = P(d)\cdot\sum_z P(w, z|d) = P(d)\cdot\sum_z P(z|d)\cdot P(w|z) = P(d)\cdot\sum_z \theta_{d,z}\cdot\phi_{z,w}$, where $\theta_{d,z}$ stands for the probability that a topic z appears in document d, while $\phi_{z,w}$ denotes the probability that a word w appears in topic z (see Table 1).

Figure 2(i) depicts the generative process of PLSA: *(1)* Select a document d with probability $P(d)$. *(2)* For each word position $n \in \{1, \ldots, N_d\}$: *(a)* Select a topic $z_{d,n}$ from distribution θ_d. *(b)* Select the word $w_{d,n}$ from distribution $\phi_{z_{d,n}}$. Note that $P(d)$ is the frequency of d in the corpus, thus being uniform in practice.

In total, PLSA should estimate $|D| \cdot |Z| + |Z| \cdot |V|$ parameters: $\theta_d = \{P(z|d)\}_{z\in Z}$ for each $d \in D$ and $\phi_z = \{P(w|z)\}_{w\in V}$ for each $z \in Z$. Consequently, the number of parameters grows linearly with the number of documents, leading to overfitting [10].

Latent Dirichlet Allocation (LDA) [10]. Unlike PLSA, which regards each document d as a list of probabilities θ_d, LDA assigns a random variable of $|Z|$ parameters with a Dirichlet prior to distribution θ_d. In a latter variant, a $|V|$-parameter variable with a Dirichlet prior was also assigned to ϕ_z [28]. The number of topics $|Z|$ is given as a parameter to the model and raises the following issue: the smaller the number of topics is, the broader and more inaccurate is their content, failing to capture the diverse themes discussed in the corpus; in contrast, for large values of $|Z|$, the model is likely to overfit, learning spurious word co-occurrence patterns [58].

Figure 2(ii) shows the generative process of LDA: *(1)* For each topic $z \in Z$, draw a distribution ϕ_z from $Dir(\beta)$. *(2)* For each document $d \in D$: *(a)* Select a distribution θ_d from $Dir(\alpha)$. *(b)* For each word position $n \in \{1, \ldots, N_d\}$: *(i)* Draw a topic $z_{d,n}$ from distribution θ_d. *(ii)* Draw the word $w_{d,n}$ from distribution $\phi_{z_{d,n}}$.

Note that the hyperparameters α and β of the Dirichlet priors on θ and ϕ, respectively, distinguish LDA from PLSA. The former denotes the frequency with which a topic is sampled for a document, while the latter shows the frequency of a word in a topic, before actually observing any words in D.

Labeled LDA (LLDA) [51]. This is a supervised variant of LDA that characterizes a corpus D with a set of observed labels Λ. Each document d is modeled as a multinomial distribution of labels from $\Lambda_d \subseteq \Lambda$. Subsequently, each word $w \in d$ is picked from a distribution ϕ_z of some label z contained in Λ_d. Besides the observed labels, LLDA can also use $|Z|$ latent topics for all documents, assigning the labels "Topic 1",..., "Topic $|Z|$" to each document $d \in D$ [50].

Figure 2(iii) presents the generative process of LLDA: *(1)* For each topic $z \in Z$, draw a distribution ϕ_z from $Dir(\beta)$. *(2)* For each document $d \in D$: *(a)* Construct distribution Λ_d by selecting each topic $z \in Z$ as a label based on a Bernoulli distribution with parameter Φ_z. *(b)* Select a multinomial distribution θ_d over Λ_d from $Dir(\alpha)$. *(c)* For each word position $n \in \{1, \ldots, N_d\}$: *(i)* Draw a label $z_{d,n}$ from distribution θ_d. *(ii)* Draw the word $w_{d,n}$ from distribution $\phi_{z_{d,n}}$. Note that the prior probability of adding a topic z to $\Lambda_d(\Phi_z)$ is practically superfluous, as Λ_d is observed for each document d.

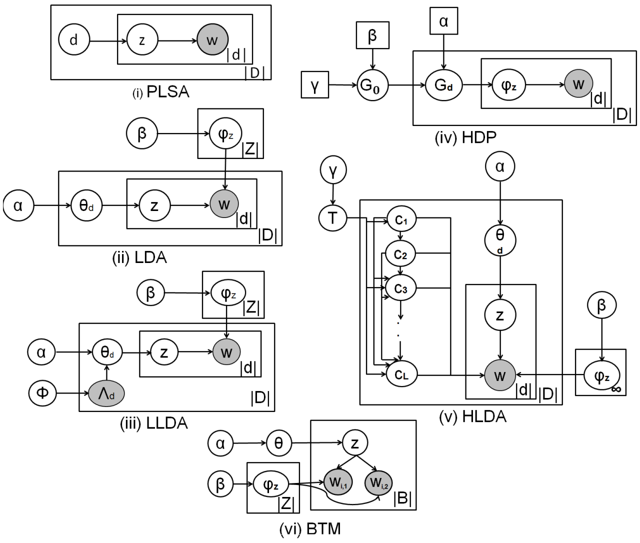

Figure 2: Plate diagrams for: (i) Probabilistic Latent Semantic Analysis (PLSA), (ii) Latent Dirichlet Allocation (LDA), (iii) Labeled LDA (LLDA), (iv) Hierarchical Dirichlet Process (HDP), (v) Hierarchical LDA (HLDA), and (vi) Biterm Topic Model (BTM).

Hierarchical Dirichlet Process (HDP) [60]. This Bayesian nonparametric model is crafted for clustering the observations of a group into mixing components. In PMR, each document corresponds to a group, the words of the document constitute the observations within the group, and the topics comprise the mixing components in the form of distributions over words.

Two are the main properties of HDP: *(i)* The number of mixing components is countably infinite and unknown beforehand. This is achieved by assigning a random variable G_j to the j^{th} group distributed according to $DP(\alpha, G_0)$, where DP stands for a Dirichlet Process, which is a probability distribution over distributions (i.e., samples from a DP are probability distributions themselves). G_0 is the base probability distribution, playing the role of the mean around which distributions are sampled by DP, while α is called concentration parameter and can be thought as an inverse variance. G_0 also follows a $DP(\gamma, H)$. *(ii)* The groups share the same components. This is achieved by linking the DPs of all groups, under the same G_0. More formally, HDP is defined as: $G_0|\gamma, H \sim DP(\gamma, H)$ and $G_j|\alpha, G_0 \sim DP(\alpha, G_0)\ \forall j$.

Figure 2(iv) shows the generative process for 1 hierarchical level: *(1)* Draw G_0 from $DP(\gamma, H)$, where H is a $Dir(\beta)$. G_0 provides an unbounded number of ϕ_z distributions, i.e., topics that can be assigned to any document $d \in D$ [60]. *(2)* For each document $d \in D$: *(a)* Associate a subset of distributions ϕ_z with d, by drawing G_d from $DP(\alpha, G_0)$. *(b)* For each word position $n \in \{1, \ldots, N_d\}$: *(i)* Pick a distribution $\phi_{z_{d,n}}$ from G_d. *(ii)* Draw the word $w_{d,n}$ from $\phi_{z_{d,n}}$.

Note that it is straightforward to add more hierarchical levels to HDP, by exploiting its recursive nature. For example, in multiple corpora, where documents are grouped into broader categories, topics shared between categories are revealed and can be compared with topics shared between individual documents [60].

Hierarchical LDA (HLDA) [8, 16]. This model extends LDA by organizing the topics in Z into a hierarchical tree such that every tree node represents a single topic. The broader a topic is, the higher is its level in the tree, with the most specific topics

Table 2: Statistics for each user group in our dataset.

	IS	BU	IP	All Users
Users	20	20	9	60
Outgoing tweets (TR)	47,659	48,836	42,566	192,328
Minimum per user	1,100	766	1,602	766
Mean per user	2,383	2,442	4,730	3,205
Maximum per user	6,406	8,025	17,761	17,761
Retweets (R)	27,344	32,951	38,013	140,649
Minimum per user	840	445	1,198	445
Mean per user	1,367	1,648	4,224	2,344
Maximum per user	2,486	6,814	17,761	17,761
Incoming tweets (E)	390,638	49,566	10,285	484,698
Minimum per user	8,936	696	525	525
Mean per user	19,532	2,478	1,143	8,078
Maximum per user	53,003	7,726	1,985	53,003
Follower's tweets (F)	665,778	166,233	50,330	1,391,579
Minimum per user	1,074	110	348	110
Mean per user	33,289	8,312	5,592	23,193
Maximum per user	144,398	52,318	33,639	447,639

assigned to the leaves. Although the tree levels are fixed, the branching factor is inferred from the data, leading to a nonparametric functionality. Each document is modeled as a path from the root to a leaf and its words are generated by the topics across this path. Hence, every document representation is derived from the topics of a single path, rather than all topics in Z (as in LDA).

Specifically, HLDA is based on the *Chinese Restaurant Process* (CRP), which is a distribution over partitions of integers. CRP assumes the existence of a Chinese restaurant with an infinite number of tables. The first customer selects the first table, while the n^{th} customer selects a table based on the following probabilities [8]: $\mathbf{P_1}$=P(occupied $table_i$|previous customers)=$n_i/(n$-1+$\gamma)$, $\mathbf{P_2}$=P(next unoccupied $table_i$|previous customers)=$\gamma/(n$-1+$\gamma)$, where γ is a parameter controlling the possibility for a new customer to sit to an occupied or an empty table, and n_i are the customers already seated on table i. The placement of M customers produces a partition of M integers.

In fact, HLDA relies on the *nested Chinese Restaurant Process* (nCRP), which extends CRP by building a infinite hierarchy of Chinese restaurants. It assumes that there exists an infinite number of Chinese restaurants, each with an infinite number of tables. One of the restaurants is the root and every restaurant's table has a label, pointing to another restaurant – a restaurant cannot be referred by more than one label in the hierarchy. To illustrate the functionality of this process, assume a customer that visits the restaurants for L days. Starting from the root, she forms a path of L restaurants, one per day, following the labels of the tables she has chosen to sit based on P_1 and P_2. The repetition of this process for M customers yields an L-level hierarchy of restaurants.

Figure 2(v) presents the generative process of HLDA, with T denoting an infinite tree drawn from nCRP(γ) (i.e., the infinite set of possible L-level paths): *(1)* For each restaurant z in T, draw a distribution ϕ_z from $Dir(\beta)$. *(2)* For each document $d \in D$: *(a)* Draw an L-level path from T as follows: Let $c_{d,1}$ be the root restaurant. For each level $l \in \{2, \ldots, L\}$ pick a table from restaurant $c_{d,l-1}$ using P_1 and P_2 and set $c_{d,l-1}$ to refer to the restaurant $c_{d,l}$, which is indicated by that table. *(b)* Select a distribution θ_d over $\{1, \ldots, L\}$ from $Dir(\alpha)$. *(c)* For each word position $n \in \{1, \ldots, N_d\}$: *(i)* Draw a level $l_{d,n} \in \{1, \ldots, L\}$ from θ_d. *(ii)* Draw the word $w_{d,n}$ from distribution $\phi_{z_{d,n}}$, where $z_{d,n}$ is the topic corresponding to the restaurant $c_{d,l_{d,n}}$.

Biterm Topic Model (BTM) [18, 61]. At the core of this model lies the notion of *biterm*, which is an unordered pair of words that are located in close distance within a given text. In short documents, close distance means that both words belong to the same document, whereas in longer texts, it means that they co-occur within a window of tokens that is given as a parameter to BTM. Based on this notion, BTM addresses the sparsity in short texts like tweets (Challenge C1) in two ways: *(i)* It models the word co-occurrence for topic learning *explicitly* by considering biterms (rather than *implicitly* as in LDA, where word co-occurrence is captured by drawing a document's words from topics of the same topic distribution θ_d). *(ii)* It considers the entire corpus D as a set of biterms B, extracting word patterns from the entire training set (rather than an individual document). Overall, BTM assumes that the corpus consists of a mixture of topics and directly models the biterm generation from these topics.

Figure 2(vi) shows the generative process of BTM: *(1)* For each topic $z \in Z$, draw a distribution ϕ_z from $Dir(\beta)$. *(2)* For the entire corpus D, select a multinomial distribution θ over Z from $Dir(\alpha)$. *(3)* For each biterm position n in the entire corpus $\{1, \ldots, |B|\}$: *(a)* Draw a topic z_n from θ. *(b)* Draw two words $w_{n,1}$, $w_{n,2}$ from ϕ_{z_n}.

Note that BTM does not contain a generation process for documents. The distribution θ_d for an individual document $d \in D$ is inferred from the formula $P(z/d) = \sum_{b \in d} P(z/b) \cdot P(b/d)$, which presumes that the document-level topic proportions can be derived from the document-level generated biterms [18, 61].

Other models. There is a plethora of topic models in the literature. Most of them lie out of the scope of our experimental analysis, because they encapsulate external or non-textual information. For example, [54] and [12] incorporate temporal information from users' activity, while [21] proposes a representation model called *badges*, which combines the posting activity of Twitter users with self-reported profile information. Other topic models are incompatible with the ranking-based recommendation algorithm for Content-based PMR. For example, Twitter-LDA [64] and Dirichlet Multinomial Mixture Model [47] assign a single topic to every tweet, thus yielding too many ties when ranking document models in decreasing similarity score - all tweets with the same inferred topic are equally similar with the user model.

Using Topic Models. To build a topic model for a particular user u, we average the distributions corresponding to the documents that capture her preferences. To compare user models with document models, we use the cosine similarity. To address the four Twitter challenges, which hamper the functionality of topic models due to the scarcity of word co-occurrence patterns, we apply two different pooling schemes to the training data: *(i)* the aggregation on users, called *User Pooling* (UP), where all tweets posted by the same user are considered as a single document, and *(ii)* the aggregation on hashtags, called *Hashtag Pooling* (HP), where all tweets annotated with the same hashtag form a single document (the tweets without any hashtag are treated as individual documents). We also consider *No Pooling* (NP), where each tweet is considered as an individual document. Finally, we estimate the parameters of all topic models using *Gibbs Sampling* [24], except for PLSA, which uses *Expectation Maximization* [19].

4 EXPERIMENTAL SETUP

All methods were implemented in Java 8 and performed in a server with Intel Xeon E5-4603@2.20 GHz (32 cores) and 120GB RAM, running Ubuntu 14.04.

Table 3: The 10 most frequent languages in our dataset, which collectively cover 1,879,470 tweets (91% of all tweets).

	English	Japanese	Chinese	Portuguese	Thai	French	Korean	German	Indonesian	Spanish
Total Tweets	1,710,919	71,242	35,356	14,416	13,964	12,895	10,220	5,038	4,339	1,081
Relative Frequency	82.71%	3.44%	1.71%	0.70%	0.68%	0.62%	0.49%	0.24%	0.21%	0.05%

Our dataset was derived from a large Twitter corpus that captures almost 30% of all public messages published on Twitter worldwide between June 1, 2009 and December 31, 2009 [62]. Although slightly dated, recent studies have verified that core aspects of the users activity in Twitter remain unchanged over the years (e.g., retweet patterns for individual messages and users) [29]. Most importantly, this dataset can be combined with a publicly available snapshot of the entire social graph of Twitter as of August, 2009 (https://an.kaist.ac.kr/traces/WWW2010.html). Given that every record includes the raw tweet along with the corresponding usernames and timestamps, we can simulate the tweet feed of every user during the second half of 2009 with very high accuracy. To ensure that there is a critical mass of tweets for all representation sources and a representative set of testing documents for every user, we removed from our dataset those users that had less than three followers and less than three followees. We also discarded all users with less than 400 retweets.

From the remaining users, we populated each of the three user types we defined in Section 2 with 20 users. For IS, we selected the 20 users with the lowest posting ratios, for BU the 20 users with the closest to 1 ratios, and for IP, the 20 users with the highest ratios. The difference between the maximum ratio for IS (0.13) and the minimum one for BU (0.76) is large enough to guarantee significantly different behavior. However, the maximum ratio among BU users is 1.16, whereas the minimum one among IP users is 1.20, due to the scarcity of information providers in our dataset. This means that the two categories are too close, a situation that could introduce noise to the experimental results. To ensure distinctive behavior, we placed in the IP group the nine users that have a posting ratio higher than 2. The remaining 11 users with the highest ratios are included in the group All Users, which additionally unites IS, BU and IP to build a large dataset with 2.07 million tweets and 60 users, in total. The technical characteristics of the four resulting user groups appear in Table 2.

Each user has a different train set, which is determined by the representation source; for example, the train set for T contains all tweets of a user, except for her retweets. In contrast, the test set of every user is independent of the representation source and contains her incoming tweets; the retweeted ones constitute the *positive examples* and the rest are the *negative examples* [17, 33, 41, 56]. However, the former are sparse and imbalanced across time in our dataset. Following [17], we retain a reasonable proportion between the two classes for each user by placing the 20% most recent of her retweets in the test set. The earliest tweet in this sample splits each user's timeline in two phases: the *training* and the *testing phase*. Based on this division, we sampled the negative data as follows: for each positive tweet in the test set, we randomly added four negative ones from the testing phase [17]. Accordingly, the train set of every representation source is restricted to all the tweets that fall in the training phase.

Based on this setup, we build the user models as follows: for the bag and graph models, we learn a separate model $UM_s(u)$ for every combination of a user u and a representation source s, using the corresponding train set. For the topic models, we first learn a single model $M(s)$ for each representation source, using the train set of all 60 users. Then, we use $M(s)$ to infer the distributions over topics for the training tweets of u that stem from s. Finally, the user model $UM_s(u)$ is constructed by computing the centroid of u's training vectors/tweets [50].

Note that for all models, we converted the raw text of all training and testing tweets into lowercase. For all token-based models, we tokenized all tweets on white spaces and punctuation, we squeezed repeated letters and we kept together URLs, hashtags, mentions and emoticons. We also removed the 100 most frequent tokens across all training tweets, as they practically correspond to stop words. We did not apply any language-specific pre-processing technique, as our dataset is multilingual.

In more detail, Table 3 presents the 10 most frequent languages in our dataset along with the number of tweets that correspond to them. To identify them, we first cleaned all tweets from hashtags, mentions, URLs and emoticons in order to reduce the noise of non-English tweets. Then, we aggregated the tweets per user (UP) to facilitate language detection. Finally, we automatically detected the prevalent language in every pseudo-document (i.e., user) [46] and assigned all relevant tweets to it. As expected, English is the dominant language, but our corpus is highly multilingual, with 3 Asian languages ranked within the top 5 positions. This multilingual content (Challenge C3) prevents us from boosting the performance of representation models with language-specific pre-processing like stemming, lemmatization and part-of-speech tagging, as is typically done in the literature [4, 17, 65]. Instead, our experimental methodology is language-agnostic.

Performance Measures. We assess the effectiveness of representation models using the *Average Precision* (**AP**) of a user model $UM_s(u)$, which is the average of the Precision-at-n ($P@n$) values for all retweets. Formally [17, 41]: $AP(UM_s(u)) = 1/|R(u)| \cdot \sum_{n=1}^{N} P@n \cdot RT(n)$, where $P@n$ is the proportion of the top-n ranked tweets that have been retweeted, $RT(n)$=1 if n is a retweet and 0 otherwise, N is the size of the test set, and $|R(u)|$ is the total number of retweets in the test set. Thus, AP expresses the performance of a representation model over an individual user. To calculate the performance of a user group U, we define *Mean Average Precision* (**MAP**) as the average AP over all users in U.

To assess the robustness of a representation model with respect to its internal configuration, we consider its *MAP deviation*, i.e., the difference between the maximum and the minimum MAP of the considered parameter configurations over a specific group of users. The lower the MAP deviation, the higher the robustness.

To estimate the time efficiency of representation models, we employ two measures: *(i)* The *training time* (**TTime**) captures the aggregated modeling time that is required for modeling all 60 users. For topic models, this also includes the time that is required for training once the model $M(s)$ from the entire train set. *(ii)* The *testing time* (**ETime**) expresses the total time that is required for processing the test set of all 60 users, i.e., to compare all user models with their testing tweets and to rank the latter in descending order of similarity. For a fair comparison, we do not consider works that parallelize representation models (e.g., [16, 57, 63]), as most models are not adapted to distributed processing.

Parameter Tuning. For each representation model, we tried a wide range of meaningful parameter configurations. They are reported in Tables 4 and 5. In total, we employed 223 different

Table 4: Configurations of the 5 context-agnostic (topic) models. CS, NP, UP and HP stand for cosine similarity, no pooling, user pooling and hashtag pooling, respectively.

	LDA	LLDA	BTM	HDP	HLDA
#Topics	{50,100,150,200}			-	-
#Iterations	{1,000, 2,000}		1,000		
Pooling	{NP, UP, HP}				UP
α	50/#Topics			1.0	{10, 20}
β	0.01			{0.1, 0.5}	
γ	-	-	-	1.0	{0.5, 1.0}
Aggregation function	{centroid, Rocchio}				
Similarity measure	CS				
Fixed parameters	-	-	r=30	-	levels=3
#Configurations	48	48	24	12	16

configurations – excluding those violating the *memory constraint* (the memory consumption should be less than 32GB RAM), or the *time constraint* (*TTime* should be less than 5 days). As a result, we excluded PLSA from our analysis, since it violated the memory constraint for all configurations we considered.

In total, 9 representation models were tested in our experiments. For LDA, LLDA and BTM, α and β were tuned according to [58]. For LLDA, the number of topics refers to the latent topics assigned to every tweet in addition to the tweet-specific labels. For the latter, we followed [50], using: *(i)* one label for every hashtag that occurs more than 30 times in the training tweets, *(ii)* the question mark, *(iii)* 9 categories of emoticons (i.e., "smile", "frown", "wink", "big grin", "heart", "surprise", "awkward" and "confused"), and *(iv)* the @user label for the training tweets that mention a user as the first word. Most of these labels were quite frequent in our corpus and, thus, we considered 10 variations for them, based on [50]. For example, "frown" produced the labels: :(-0 to :(-9. The only labels with no variations are the hashtag ones and the emoticons "big grin", "heart", "surprise" and "confused".

For HLDA, we did not employ the pooling strategies NP and HP and more than three hierarchical levels, as these configurations violated the time constraint. Hence, we only varied α and γ.

For BTM, we selected 1,000 iterations following [61]. For individual tweets, we set the context window (r), i.e., the maximum distance between two words in a biterm, equal to the size of the tweet itself. For the large pseudo-documents in user and hashtag pooling, we set r=30 based on [61]; for greater values, BTM conveys no significant improvement over LDA, since the larger the distance between the words in a biterm is, the more irrelevant are their topics. Larger window sizes yield higher *TTime*, too.

For bag models, some configuration combinations are invalid: JS is applied only with BF weights, GJS only with TF and TF-IDF, and the character-based n-grams, CN, are not combined with TF-IDF. Also, BF is exclusively coupled with the sum aggregation function, which in this case is equivalent to applying the boolean operator OR among the individual document models. For the Rocchio algorithm, we set α=0.8 and β=0.2, and we used only the CS similarity measure in combination with the TF and TF-IDF weights for those representation sources that contain both positive and negative examples, namely C, E, TE, RE, TC, RC and EF.

5 EXPERIMENTAL ANALYSIS

Effectiveness & Robustness. To assess the performance of the 9 representation models, we measure their *Mean MAP*, *Min MAP* and *Max MAP*, i.e., their average, minimum and maximum MAP, respectively, over all relevant configurations for a particular combination of a user type and a representation source. The outcomes for All Users are presented in Figure 3, which considers the 5 individual representation sources along with the 3 combinations that achieve the best average performance. Due to space limitations, the remaining 5 combinations along with the figures for IP, BU and IS are only presented in the extended version of the paper [59]. Nevertheless, we consider them in the discussion of our experimental results.

Table 5: Configurations of the 4 context-based models. Remember that BF and TF stand for boolean and term frequency weights, respectively, while (G)JS, CoS, VS, and NS denote the (generalized) Jaccard, the containment, the value and the normalized value graph similarities, resp.

	TN	CN	TNG	CNG
n	{1,2,3}	{2,3,4}	{1,2,3}	{2,3,4}
Weighting scheme	{BF,TF,TF-IDF}	{BF,TF}	-	-
Aggregation function	{sum, centroid, Rocchio}		-	-
Similarity measure	{CS, JS, GJS}		{CoS, VS, NS}	
#Configurations	36*	21*	9	9

* excluding invalid configuration combinations

Every diagram also reports the MAP for two baseline methods: (i) the *Chronological Ordering* (CHR), which ranks the test set from the latest tweet (first position) to the earliest one (last position), and (ii) the *Random Ordering* (RAN), which sorts the test set in an arbitrary order. For RAN, we performed 1,000 iterations per user and considered the overall average per user type.

Starting with All Users in Figure 3, we observe that TNG consistently outperforms all other models across all representation sources. Its Mean MAP fluctuates between 0.625 (EF) and 0.784 (T). The second most effective model is TN, whose Mean MAP ranges from 0.541 (EF) to 0.673 (T). The dominance of TNG over TN is statistically significant (p<0.05) and should be attributed to its ability to capture the relations between neighboring token n-grams through the weighted edges that connect them. In this way, TNG incorporates global contextual information into its model and inherently alleviates sparsity (Challenge C1). Instead, TN exclusively captures local contextual information in the form of token sequences. The same holds for Challenges C2 and C4: both models fail to identify misspelled or non-standard token n-grams, but TNG is better suited to capture their patterns by encapsulating their neighborhood.

Regarding *robustness*, TN is more sensitive to its configuration. Its MAP deviation ranges from 0.359 (TF, RF) to 0.444 (EF), while for TNG, it fluctuates between 0.096 (T) and 0.281 (EF). In all cases, the difference between the two models is statistically significant (p<0.05). TNG is superior, because its performance is fine-tuned by just two parameters: the size of n-grams (n) and the similarity measure. TN additionally involves the aggregation function and the weighting scheme, increasing drastically its possible configurations. In general, *every configuration parameter acts as a degree of freedom for a representation model; the higher the overall number of parameters is, the more flexible the model gets and the less robust it is expected to be with respect to its configuration.*

Comparing the character-based instantiations of bag and graph models, we notice that their difference in Mean Map is statistically insignificant. For CNG, it fluctuates between 0.368 (TF) and 0.477 (R), while for CN, it ranges from 0.334 (TF) to 0.436 (RC). This implies that *there is no benefit in considering global contextual*

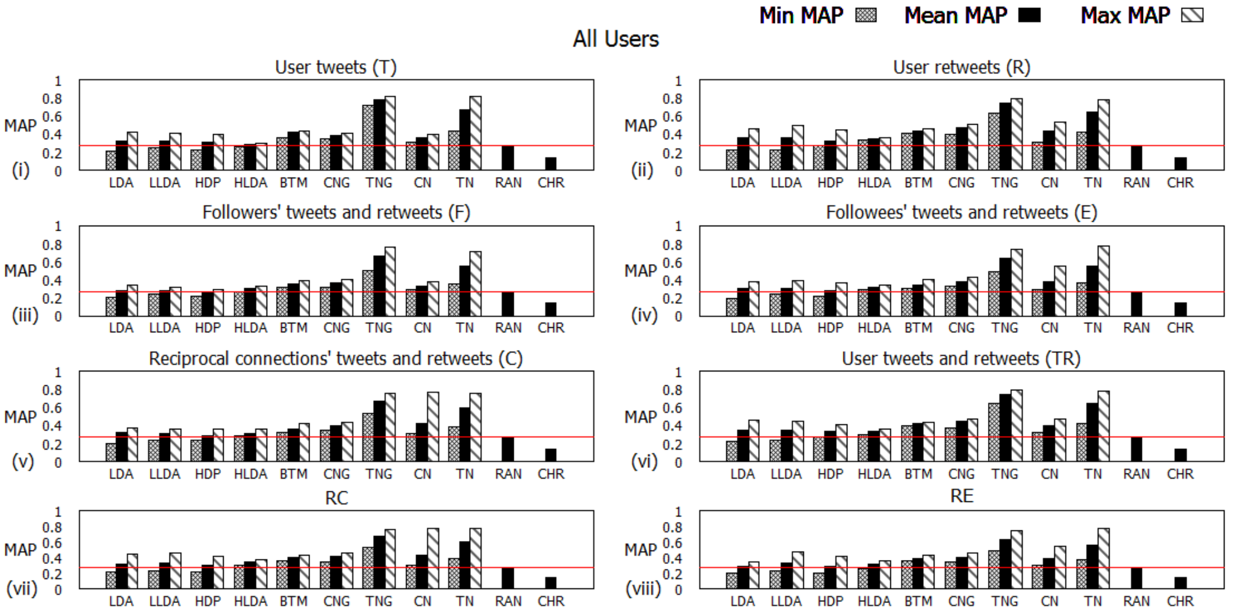

Figure 3: Effectiveness of the nine representation models over All Users in combination with the five individual representation sources and their three best performing pairwise combinations with respect to MAP. Higher bars indicate better performance. The red line corresponds to the performance of the best baseline, RAN.

information for representation models that are based on character n-grams. The strength of these models lies in the local contextual information that is captured in the sequence of characters.

Regarding robustness, the relative sensitivity of character-based models exhibits the same patterns as their token-based counterparts: the bag models are significantly less robust than the graph ones, due to their larger number of parameters and configurations. In more detail, the MAP deviation ranges from 0.061 (T) to 0.114 (RF) for CNG and from 0.077 (T) to 0.476 (RC) for CN.

Concerning the topic models, we observe that BTM consistently achieves the highest effectiveness across all representation sources. Its Mean MAP fluctuates between 0.340 (EF) and 0.434 (R). All other topic models exhibit practically equivalent performance: their highest Mean MAP ranges from 0.337 (for HDP with TR) to 0.360 (for LLDA with R), whereas their lowest Mean MAP fluctuates between 0.265 (for HDP over F) and 0.270 (for LLDA over TF). As a result, HDP, LDA, LLDA and HLDA outperform only CHR to a large extent. This indicates that *recency alone constitutes an inadequate criterion for recommending content in microblogging platforms. Any model that considers the personal preferences of a user offers more accurate suggestions.*

Compared to the second baseline (RAN), HDP, LDA, LLDA and HLDA are more effective, but to a minor extent. The Mean MAP of RAN amounts to 0.270, thus, some configurations of these topic models perform consistently worse than RAN, across all representation sources. Most of these configurations correspond to the absence of pooling (NP), where every tweet is considered as an individual document. In these settings, these four topic models fail to extract distinctive patterns from any representation source, producing noisy user and document models. This suggests that *sparsity is the main obstacle to most topic models.*

Regarding robustness, we can distinguish the 5 topic models into two categories: the first one involves 3 models that are highly sensitive to their configuration, namely LDA, LLDA and HDP. Their MAP deviation starts from 0.119, 0.075, and 0.077, respectively, and raises up to 0.250, 0.264 and 0.211, respectively. These values are extremely high, when compared to their absolute Mean MAP. This means that *extensive fine-tuning is required for successfully applying most topic models to text-based PMR.* In contrast, MAP deviation fluctuates between 0.034 and 0.109 for both HLDA and BTM. For the former, this is probably caused by the limited number of configurations that satisfied our time constraint, while for the latter, it should be attributed to its Twitter-specific functionality.

Discussion. We now discuss the novel insights that can be deduced from our experimental analysis. We start by comparing token- with character-based models. We observe that the former are significantly more effective than the latter for both bag and graph models. At first, this seems counter-intuitive, as CNG and CN are in a better position to address Challenge C1: they extract more features from sparse documents than TNG and TN, respectively [48]. They are also better equipped to address Challenges C2 and C4: by operating at a finer granularity, they can identify similarities even between noisy and non-standard tokens. Yet, the character-based models seem to accumulate noise when aggregating individual tweet models into a user model. Their similarity measures fail to capture distinctive information about the real interests of a user, yielding high scores for many irrelevant, unseen documents. The lower n is, the more intensive is this phenomenon. In fact, most bigrams are shared by both relevant and irrelevant examples, which explains why the poorest performance corresponds to n=2 for both CNG and CN. The only advantage of character-based models over their token-based counterparts is their higher robustness. However, their lower values for MAP deviation are probably caused by their lower absolute values for MAP.

Among the topic models, BTM consistently performs best with respect to effectiveness and robustness. Its superiority stems from the two inherent characteristics that optimize its functionality for the short and noisy documents in Twitter: *(i)* it considers pairs of words (i.e., biterms), instead of individual tokens, and *(ii)* it bypasses sparsity (Challenge C1) by capturing topic patterns at the level of entire corpora, instead of extracting them from individual documents. Compared to context-based models, though, BTM is significantly less effective than the token-based bag and graph models. Its performance is very close to the character-based models, especially CNG, with their difference in terms of effectiveness and robustness being statistically insignificant.

Table 6: Performance of all 13 representation sources over the 4 user types with respect to Min(imum), Mean and Max(imum) MAP across all configurations of the 9 representation models. The rightmost column presents the average performance per user type.

		R	**T**	**E**	**F**	**C**	**TR**	**RE**	**RF**	**RC**	**TE**	**TF**	**TC**	**EF**	Average
All Users	Min MAP	0.225	0.217	0.196	0.208	0.199	0.222	0.205	0.199	0.212	0.201	0.205	0.201	0.198	0.207
	Mean MAP	0.456	0.429	0.392	0.378	0.410	0.448	0.402	0.386	0.427	0.387	0.377	0.406	0.377	0.406
	Max MAP	0.796	0.816	0.771	0.764	0.768	0.797	0.775	0.758	0.783	0.775	0.764	0.776	0.780	0.779
IS	Min MAP	0.193	0.199	0.192	0.191	0.196	0.212	0.199	0.195	0.188	0.200	0.192	0.195	0.195	0.196
	Mean MAP	0.415	0.383	0.351	0.336	0.364	0.407	0.357	0.340	0.374	0.347	0.342	0.357	0.336	0.362
	Max MAP	0.776	0.818	0.733	0.699	0.732	0.790	0.732	0.699	0.733	0.732	0.699	0.733	0.736	0.739
BU	Min MAP	0.212	0.174	0.168	0.192	0.168	0.198	0.168	0.182	0.178	0.167	0.179	0.171	0.175	0.179
	Mean MAP	0.474	0.430	0.400	0.376	0.420	0.453	0.419	0.389	0.443	0.391	0.381	0.406	0.376	0.412
	Max MAP	0.733	0.775	0.747	0.746	0.745	0.753	0.746	0.741	0.753	0.746	0.741	0.743	0.741	0.747
IP	Min MAP	0.243	0.220	0.200	0.198	0.186	0.242	0.200	0.202	0.182	0.189	0.212	0.198	0.208	0.206
	Mean MAP	0.524	0.446	0.460	0.458	0.488	0.493	0.470	0.467	0.497	0.428	0.443	0.444	0.440	0.466
	Max MAP	0.878	0.857	0.854	0.858	0.839	0.854	0.854	0.856	0.854	0.854	0.858	0.857	0.854	0.856

Generally, we can conclude that all topic models fail to improve the token-based bag and graph models in the context of Content-based PMR. Both TN and TNG achieve twice as high Mean MAP, on average, across all representation sources. The poor performance of topic models is caused by two factors: *(i)* they disregard the contextual information that is encapsulated in word ordering, and *(ii)* the Challenges C1 to C4 of Twitter. Indeed, most topic models were originally developed to extract topics from word co-occurrences in individual documents, but the sparse and noisy co-occurrence patterns in the short text of tweets reduce drastically their effectiveness.

User Types. We notice that the *relative* performance of representation models for IP, BU and IS remains practically the same as in Figure 3. Yet, there are significant differences in their *absolute* performance. On average, across all models and representation sources, IP users have higher Mean MAP than IS and BU users by 30% and 13% respectively, while BU surpasses IS by 15%. Compared to All Users, IP increases Mean MAP by 12%, BU by just 1%, while IS decreases it by 11%, on average, across all models and representation sources. These patterns are also demonstrated in the rightmost column of Table 6, which essentially expresses the average values for Min(imum), Mean and Max(imum) MAP per user type across all combinations of representation models, configurations and representation sources.

Therefore, we can conclude that *the more information an individual user produces, the more reliable are the models that represent her interests*, and vice versa: *taciturn users are the most difficult to model.* This should be expected for R and T, since the posting activity of a user increases the content that is available for building their models. However, this pattern applies to the other representation sources, too, because IP users are typically in closer contact with their followees, followers and reciprocal users, who thus can represent their interests effectively. The opposite holds for IS users. In the middle of these two extremes lie BU users, which exhibit a balanced activity in all respects.

Overall, these patterns suggest that the user categories we defined in Section 2 have a real impact on the performance of Content-based PMR. Therefore, they should be taken into account when designing a Content-based PMR approach, as different user types call for different recommendation methods.

Representation Sources. We now examine the relative effectiveness of the five representation sources and their eight pairwise combinations. Table 6 reports the performance for every combination of a user type and a representation source with respect to Min(imum), Mean and Max(imum) MAP over all configurations of the nine representation models.

Starting with individual representation sources, we observe that R consistently achieves the highest Mean MAP across all user types. This suggests that *retweets constitute the most effective means of capturing user preferences under all settings*, as users repost tweets that have attracted their attention and approval.

The second best individual source is T. This applies to all user types, except for IP, where T actually exhibits the worst performance across all individual sources. These patterns suggest that *the tweets of a user offer valuable information for her interests as long as her posting ratio is lower than 2*; such users post their messages thoughtfully, when they have something important to say. Instead, the IP users are hyperactive, posting quite frequently careless and noisy messages that do not reflect their preferences, e.g., by engaging into irrelevant discussions with other users.

Among the remaining individual sources, C achieves the best performance, followed by E and F. This pattern is consistent across all user types, with the differences being statistically significant in most cases. We can deduce, therefore, that *the reciprocal connections in Twitter reflect friendships between users that share common interests to a large extent.* Instead, the one-way connections offer weaker indications of common tastes among users, especially when they are not initiated by the ego user: the followers' posts (F) invariably result in noisy user models.

For the eight pairs of sources, we observe the following patterns: *(i)* All combinations of R with another source X result in higher performance for X, with the difference being statistically significant. Hence, R is able to enrich any representation source with valuable information about user preferences. *(ii)* All combinations of T with another source result in a (usually insignificant) lower performance in all cases but two, namely TF over IS and BU. This means that T typically conveys noisy, irrelevant information. *(iii)* For both R and T, all pairwise combinations degrade their own performance. The only exception is TR, which improves the effectiveness of T.

On the whole, we can conclude that *a user's retweets (R) should be used as the sole information source for building her model.* There is no need to combine it with T.

Time Efficiency. Figures 4(i) and (ii) depict the minimum, average and maximum values for $TTime$ and $ETime$, respectively, for every representation model across all configurations, representation sources and users. The vertical axes are logarithmic, with lower values indicating better performance.

Among the global context-aware models, we observe that on average, TNG is faster than CNG with respect to both $ETime$ and $TTime$ by 1 and 2 orders of magnitude, respectively. Similarly, among the local context-aware models, TN is faster than CN

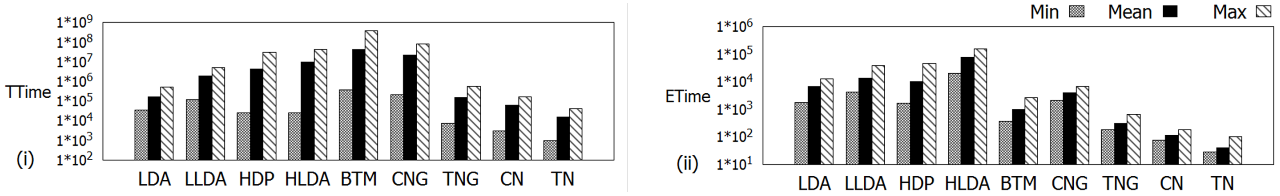

Figure 4: Time efficiency of the 9 representation models with respect to *(i)* Training Time ($TTime$), and *(ii)* Testing Time ($ETime$), on average across all configurations, representation sources and users. Lower bars indicate better performance.

by 4 and 3 times, respectively. The reason in both cases is the relative dimensionality of token- and character-based models. Typically, the latter yield a much larger feature space than the former, depending, of course, on the size of the n-grams – the larger n is, the more character n-grams are extracted from a corpus [48] and the more time-consuming is their processing.

Among the topic models, BTM exhibits the highest $TTime$. The reason is that it operates on the level of biterms, which result in a much higher dimensionality than the individual tokens that lie at the core of the other topic models. However, our analysis does not consider the most time-consuming configurations of the second worst model (HLDA), as they violated the time constraint. In practice, HLDA is expected to be slower than BTM, since its nonparametric nature lies in exploring the possible L-level hierarchical trees during topic inference. On the other extreme, LDA is the fastest topic model, while LLDA is almost 5 times faster than HDP, on average, due to its simpler models.

These patterns are slightly altered in the case of $ETime$. The worst performance by far corresponds to HLDA, due to its nonparametric functionality, while BTM turns out to be the fastest model. Unlike the other topic models, which perform Gibbs sampling for topic inference, BTM simply iterates over the biterms in a document d in order to calculate $P(z|d)$ for each topic $z \in Z$.

Comparing the model categories between them, we observe that the graph models are more time-consuming than their bag counterparts: on average, TNG and CNG are 1 and 2 orders of magnitude slower than TN and CN, respectively, for both $TTime$ and $ETime$. This should be attributed to the contextual information they incorporate in their edges, whose number is much larger in the case of CNG. Similarly, most topic models are at least 1 order of magnitude slower than their base model, TN, for both time measures, due to the time required for topic inference. Overall, *TN is consistently the most efficient representation model, due to the sparsity of tweets and the resulting low dimensionality.*

These patterns are consistent across all user types and representation sources. For the latter, we also observed that the size of the train set affects directly $TTime$ for all representation models: the more tweets are used to build a user model, the more patterns are extracted, degrading time efficiency.

6 RELATED WORK

There has been a bulk of work on recommender systems over the years [2, 3]. Most recent works focus on microblogging platforms and Social Media, employing the bag model in order to suggest new followees [13], URLs [15] and hashtags [40]. Others employ topic models for the same tasks, e.g., hashtag recommendations [27]. In the latter case, emphasis is placed on tackling sparsity through *pooling techniques*, which aggregate short texts that share similar content, or express similar ideas into lengthy pseudo-documents [4, 44]. E.g., Latent Dirichlet Allocation [10] and the Author Topic Model [52] are trained on individual messages and on messages aggregated by user and hashtag in [33].

Content-based PMR has attracted lots of interest in the data management community [14, 16, 22, 55, 63], where many works aim to improve the time efficiency of topic models. [16] parallelizes the training of HLDA through a novel concurrent dynamic matrix and a distributed tree. [14] scales up LDA through the WarpLDA algorithm, which achieves $O(1)$ time complexity per-token and fits the randomly accessed memory per-document in the L3 cache. Along with other state-of-the-art LDA samplers, this work is incorporated into LDA*, a high-end system that scales LDA training to voluminous datasets, using different samplers for various types of documents [63]. Another approach for the massive parallelization of LDA is presented in [57]. Other works facilitate real-time content recommendations in Twitter. This is done either by partitioning the social graph across a cluster in order to detect network motifs in parallel [30], or by holding the entire graph in the main memory of a single server in order to accelerate random walk-based computations on a bipartite interaction graph between users and tweets [55].

On another line of research, external resources are employed in order to augment text representation and improve their performance in various tasks. For short text clustering, Dual Latent Dirichlet Allocation learns topics from both short texts and auxiliary documents [37]. For personalized Twitter stream filtering, tweets can be transformed into RDF triples that describe their author, location and time in order to use ontologies for building user profiles [38]. User profiles can also be enriched with Wikipedia concepts [41] and with concepts from news articles that have been read by the user [34]. These approaches lie out of the scope of our analysis, which focuses on recommendations based on Twitter's internal content.

To the best of our knowledge, no prior work examines systematically Content-based PMR with respect to the aforementioned parameters, considering a wide range of options for each one.

7 CONCLUSIONS

We conclude with the following five observations about Content-based Personalized Microblog Recommendation: *(i)* The token-based vector space model achieves the best balance between effectiveness and time efficiency. In most cases, it offers the second most accurate recommendations, while involving the minimum time requirements both for training a user model and applying it to a test set. On the flip side, it involves four parameters (i.e., degrees of freedom), thus being sensitive to its configuration. *(ii)* The token n-gram graphs achieve the best balance between effectiveness and robustness. Due to the global contextual information they capture, they consistently outperform all other representation models to a significant extent, while exhibiting limited sensitivity to their configuration. Yet, they are slower than the vector space model by an order of magnitude, on average. *(iii)* The character-based models underperform their token-based counterparts, as their similarity measures cannot tackle the noise that piles up when assembling document models into user models. *(iv)* The topic models exhibit much lower effectiveness than the token-based bag models for three reasons: *1)* most of them are not crafted for the sparse, noisy and multilingual content of Twitter, *2)* they depend heavily on their configuration, and *3)* they are context-agnostic, ignoring the sequence of words in documents. Their processing is time-consuming, due to the inference of topics, requiring parallelization techniques to scale to voluminous data [16, 57, 63]. *(v)* All representation models perform best when they are built from the retweets of hyperactive users (information producers).

In the future, we plan to expand our comparative analysis to more recommendation tasks for microblogging platforms.

REFERENCES

[1] F. Abel, Q. Gao, G.-J. Houben, and K. Tao. Analyzing user modeling on twitter for personalized news recommendations. In *UMAP*, pages 1–12, 2011.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE TKDE*, 17(6):734–749, 2005.

[3] C. C. Aggarwal. *Recommender Systems - The Textbook*. Springer, 2016.

[4] D. Alvarez-Melis and M. Saveski. Topic modeling in twitter: Aggregating tweets by conversations. In *ICWSM*, pages 519–522, 2016.

[5] M. Armentano, D. Godoy, and A. Amandi. Topology-based recommendation of users in micro-blogging communities. *J. Comp. Sci. Tech.*, 27(3):624–634, 2012.

[6] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM (CACM)*, 40(3):66–72, 1997.

[7] D. M. Blei. Probabilistic topic models. *CACM*, 55(4):77–84, 2012.

[8] D. M. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *NIPS*, pages 17–24, 2003.

[9] D. M. Blei and J. D. Lafferty. Topic models. *Text mining: classification, clustering, and applications*, 10(71):34, 2009.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[11] C. Chen, D. Wu, C. Hou, and X. Yuan. Facet-based user modeling in social media for personalized ranking. In *ECIR*, pages 443–448, 2014.

[12] C. Chen, X. Zheng, C. Zhou, and D. Chen. Making recommendations on microblogs through topic modeling. In *WISE Workshops*, pages 252–265, 2013.

[13] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy. Make new friends, but keep the old: recommending people on social networking sites. In *SIGCHI*, pages 201–210, 2009.

[14] J. Chen, K. Li, J. Zhu, and W. Chen. Warplda: a cache efficient O(1) algorithm for latent dirichlet allocation. *PVLDB*, 9(10):744–755, 2016.

[15] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi. Short and tweet: experiments on recommending content from information streams. In *SIGCHI*, pages 1185–1194, 2010.

[16] J. Chen, J. Zhu, J. Lu, and S. Liu. Scalable training of hierarchical topic models. *PVLDB*, 11(7):826–839, 2018.

[17] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu. Collaborative personalized tweet recommendation. In *SIGIR*, pages 661–670, 2012.

[18] X. Cheng, X. Yan, Y. Lan, and J. Guo. Btm: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):2928–2941, 2014.

[19] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[20] Y. Duan, L. Jiang, T. Qin, M. Zhou, and H.-Y. Shum. An empirical study on learning to rank of tweets. In *COLING*, pages 295–303, 2010.

[21] K. El-Arini, U. Paquet, R. Herbrich, J. V. Gael, and B. A. y Arcas. Transparent user models for personalization. In *SIGKDD*, pages 678–686, 2012.

[22] A. El-Kishky, Y. Song, C. Wang, C. R. Voss, and J. Han. Scalable topical phrase mining from text corpora. *PVLDB*, 8(3):305–316, 2014.

[23] W. Feng and J. Wang. Retweet or not?: personalized tweet re-ranking. In *WSDM*, pages 577–586, 2013.

[24] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE TPAMI*, (6):721–741, 1984.

[25] G. Giannakopoulos, V. Karkaletsis, G. Vouros, and P. Stamatopoulos. Summarization system evaluation revisited: N-gram graphs. *ACM Transactions on Speech and Language Processing (TSLP)*, 5(3):5, 2008.

[26] G. Giannakopoulos and T. Palpanas. Content and type as orthogonal modeling features: a study on user interest awareness in entity subscription services. *International Journal of Advances on Networks and Services*, 3(2), 2010.

[27] F. Godin, V. Slavkovikj, W. De Neve, B. Schrauwen, and R. Van de Walle. Using topic models for twitter hashtag recommendation. In *WWW (Companion Volume)*, pages 593–596, 2013.

[28] T. L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 101(suppl 1):5228–5235, 2004.

[29] Q. Grossetti, C. Constantin, and et al. An homophily-based approach for fast post recommendation on twitter. In *EDBT*, pages 229–240, 2018.

[30] P. Gupta, V. Satuluri, A. Grewal, S. Gurumurthy, V. Zhabiuk, Q. Li, and J. J. Lin. Real-time twitter recommendation: Online motif detection in large dynamic graphs. *PVLDB*, 7(13):1379–1380, 2014.

[31] J. Hannon, M. Bennett, and B. Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *ACM RecSys*, pages 199–206, 2010.

[32] T. Hofmann. Probabilistic latent semantic analysis. In *UAI*, pages 289–296, 1999.

[33] L. Hong and B. D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*, pages 80–88, 2010.

[34] W. IJntema, F. Goossen, F. Frasincar, and F. Hogenboom. Ontology-based news recommendation. In *Proceedings of the EDBT/ICDT Workshops*, 2010.

[35] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *SNA-KDD*, pages 118–138, 2007.

[36] M. Jiang, P. Cui, F. Wang, W. Zhu, and S. Yang. Scalable recommendation with social contextual information. *IEEE TKDE*, 26(11):2789–2802, 2014.

[37] O. Jin, N. N. Liu, K. Zhao, Y. Yu, and Q. Yang. Transferring topical knowledge from auxiliary long texts for short text clustering. In *CIKM*, pages 775–784, 2011.

[38] P. Kapanipathi, F. Orlandi, A. P. Sheth, and A. Passant. Personalized filtering of the twitter stream. In *SPIM*, pages 6–13, 2011.

[39] Y. Kim and K. Shim. TWITOBI: A recommendation system for twitter using probabilistic modeling. In *IEEE ICDM*, pages 340–349, 2011.

[40] S. M. Kywe, T.-A. Hoang, E.-P. Lim, and F. Zhu. On recommending hashtags in twitter networks. In *SocInfo*, pages 337–350, 2012.

[41] C. Lu, W. Lam, and Y. Zhang. Twitter user modeling and tweets recommendation based on wikipedia concept graph. In *AAAI Workshops*, pages 33–38, 2012.

[42] F. D. Malliaros, P. Meladianos, and M. Vazirgiannis. Graph-based text representations: Boosting text mining, nlp and information retrieval with graphs. In *WWW Tutorials*, 2018.

[43] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[44] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *SIGIR*, pages 889–892, 2013.

[45] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *ACM Conference on Digital Libraries*, pages 195–204, 2000.

[46] F. R. Nakatani Shuyo, Fabian Kessler and R. Theis. Language detector. https://github.com/optimaize/language-detector, 2018.

[47] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2-3), 2000.

[48] G. Papadakis, G. Giannakopoulos, and G. Paliouras. Graph vs. bag representation models for the topic classification of web documents. *World Wide Web*, 19(5):887–920, 2016.

[49] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[50] D. Ramage, S. T. Dumais, and D. J. Liebling. Characterizing microblogs with topic models. In *ICWSM*, 2010.

[51] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*, pages 248–256, 2009.

[52] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *UAI*, pages 487–494, 2004.

[53] F. Rousseau and M. Vazirgiannis. Graph-of-word and TW-IDF: new approach to ad hoc IR. In *CIKM*, pages 59–68, 2013.

[54] J. Sang, D. Lu, and C. Xu. A probabilistic framework for temporal user modeling on microblogs. In *ACM CIKM*, pages 961–970, 2015.

[55] A. Sharma, J. Jiang, P. Bommannavar, B. Larson, and J. J. Lin. Graphjet: Real-time content recommendations at twitter. *PVLDB*, 9(13):1281–1292, 2016.

[56] K. Shen, J. Wu, Y. Zhang, Y. Han, X. Yang, L. Song, and X. Gu. Reorder user's tweets. *ACM TIST*, 4(1):6:1–6:17, 2013.

[57] A. J. Smola and S. M. Narayanamurthy. An architecture for parallel topic models. *PVLDB*, 3(1):703–710, 2010.

[58] M. Steyvers and T. Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

[59] E. K. Taniskidou, G. Papadakis, G. Giannakopoulos, and M. Koubarakis. Compadrative analysis of content-based personalized microblog recommendations - experiments and analysis. *CoRR*, abs/1901.0549, 2019.

[60] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *J. Amer. Stat. Assoc.*, 101(476):1566–1581, 2006.

[61] X. Yan, J. Guo, Y. Lan, and X. Cheng. A biterm topic model for short texts. In *WWW*, pages 1445–1456, 2013.

[62] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *WSDM*, pages 177–186, 2011.

[63] L. Yu, B. Cui, C. Zhang, and Y. Shao. Lda*: A robust and large-scale topic modeling system. *PVLDB*, 10(11):1406–1417, 2017.

[64] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *ECIR*, pages 338–349, 2011.

[65] X. Zhao and K. Tajima. Online retweet recommendation with item count limits. In *IEEE/WIC/ACM WI-IAT*, pages 282–289, 2014.