A

Integrating Social and Auxiliary Semantics for Multi-Faceted Topic Modeling in Twitter

JAN VOSECKY, DI JIANG, KENNETH WAI-TING LEUNG, KAI XING and WILFRED NG,
Hong Kong University of Science and Technology

Microblogging platforms, such as Twitter, have already played an important role in recent cultural, social and political events. Discovering latent topics from social streams is therefore important for many downstream applications, such as clustering, classification or recommendation. However, traditional topic models that rely on the bag-of-words assumption are insufficient to uncover the rich semantics and temporal aspects of topics in Twitter. In particular, microblog content is often influenced by external information sources, such as web documents linked from Twitter posts, and often focuses on specific entities, such as people or organizations. These external sources provide useful semantics to understand microblogs and we generally refer to these semantics as *auxiliary semantics*. In this paper, we address the mentioned issues and propose a unified framework for Multi-faceted Topic Modeling from Twitter streams. We first extract social semantics from Twitter by modeling the social chatter associated with hashtags. We further extract terms and named entities from linked web documents to serve as auxiliary semantics during topic modeling. The Multi-faceted Topic Model (MfTM) is then proposed to jointly model latent semantics among the social terms from Twitter, auxiliary terms from the linked web documents and named entities. Moreover, we capture the temporal characteristics of each topic. An efficient online inference method for MfTM is developed, which enables our model to be applied to large-scale and streaming data. Our experimental evaluation shows the effectiveness and efficiency of our model compared with state-of-the-art baselines. We evaluate each aspect of our framework and show its utility in the context of tweet clustering.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: social media; topic model; unsupervised learning; semantic enrichment

ACM Reference Format:
Jan Vosecky, Di Jiang, Kenneth W.-T. Leung, Kai Xing and Wilfred Ng, 2010. Integrating Social and Auxiliary Semantics for Multi-Faceted Topic Modeling in Twitter. *ACM Trans. Internet Technol.* V, N, Article A (January YYYY), 23 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

In recent years, social media and in particular microblogs have seen a steep rise in popularity, with users from a wide range of backgrounds contributing content in the form of short text messages. Twitter[1], a popular microblogging platform with over 500 million users is at the epicenter of the social media explosion. In Twitter, users are able

[1] http://twitter.com/

Author's addresses: Jan Vosecky, Di Jiang, Kenneth Wai-Ting Leung, Kai Xing and Wilfred Ng, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong.

DOI: http://dx.doi.org/10.1145/0000000.0000000

to create and publish short posts, referred to as *tweets*, in real time. Discovering latent topics from social streams is therefore important for many downstream applications, such as classification, clustering and user modeling [Jin et al. 2011; Rosa et al. 2010].

However, there is still a lack of accurate and efficient models for unsupervised topic modeling in microblogs. Current topic models employ a simplistic bag-of-words view of a "topic" [Blei et al. 2003; Zhao et al. 2011]. These methods fail to distinguish the rich semantics of microblog topics, such as which entities (e.g., people, organizations or locations) are being discussed. Moreover, temporal aspects of topics have not been fully addressed, although microblog topics develop in a dynamical manner. Figure 1 illustrates the mentioned characteristics of a topic in Twitter.

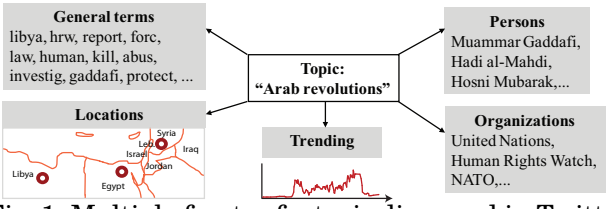


Fig. 1: Multiple facets of a topic discussed in Twitter

Topic modeling in microblogs faces many new challenges compared with traditional domains such as web documents. We summarize the main challenges as follows.

— Very brief content: microblog posts are short in length, with only 140 characters available to convey the author's message. This lack of context motivates the use of additional sources of semantics.

— Entity-oriented: microblog posts often discuss specific entities, such as famous people, organizations, or geographic locations [Celik et al. 2011]. Traditional models for textual data fail to utilize the various entity types available in microblogs.

— Facilitating social interaction: social tags in microblog posts (called *hashtags* in Twitter) present a new kind of social metadata generated by the public, with dynamic trending characteristics. In Section 3.2, we discuss how the "voice of the crowd" associated with hashtags can aid topic modeling. Also, we measure the importance of *recency* when modeling the social chatter associated with hashtags.

— Dynamic: topics in microblogs are constantly evolving, implying the need to model their temporal characteristics. Moreover, the evolving nature of social content calls for scalable and updatable models.

Previously proposed topic models are not sufficient to cover the above-mentioned issues in a unified manner. Rather, they only focus on isolated aspects of the topic discovery problem. First, some topic models consider the temporal dimension (e.g., Wang and McCallum [2006]). Second, entity-topic models (e.g., Newman et al. [2006]) consider named entities, however do not distinguish different types of entities. Third, some models utilize auxiliary data to aid topic discovery in short documents (e.g., Jin et al. [2011]). To tackle these issues in a unified and comprehensive manner, we establish a topic modeling framework illustrated in Figure 2. The framework can be divided into the following major components: (1) web document-based semantic enrichment, (2) time-sensitive hashtag-based semantic enrichment, and (3) multi-faceted topic model.

When observing Twitter posts, we notice that users often comment about issues happening in the world. Such comments may range from opinions about news events to business announcements. This observation motivates us to integrate two sources of

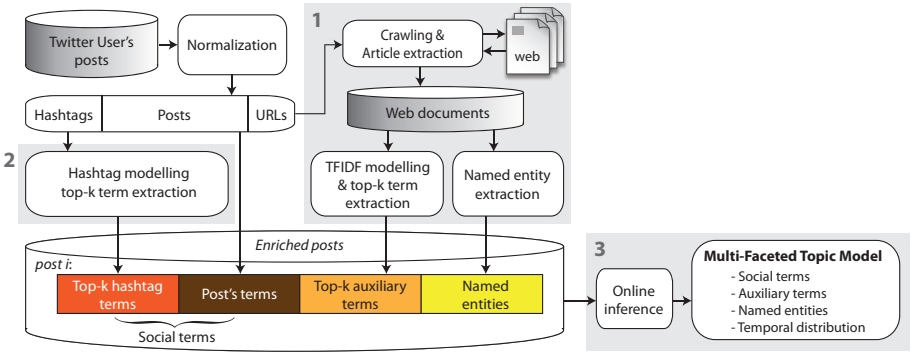


Fig. 2: Multi-faceted Twitter Topic Modeling Framework

semantics within the proposed framework. On the one hand, we consider *social semantics*, which are produced by users in Twitter. On the other hand, we utilize *auxiliary semantics*, which originate from external information sources. In a Twitter post, the linkage to auxiliary information may be shown explicitly by including a URL link. For this reason, our framework utilizes the linked web documents as they are an easily accessible source of auxiliary information. In general, other sources of auxiliary semantics may also be integrated in our framework.

Our framework takes Twitter posts as input. As a pre-processing step, the short posts are then enriched by extracting *auxiliary terms* and named entities from the linked web documents. Posts are further enriched by extracting *social terms* that co-occur with hashtags, thus obtaining additional semantics from the "voice of the crowd". To extract latent topics from the pre-processed data, we adopt a topic modeling approach and propose the Multi-faceted Topic Model (MfTM). In essence, each latent topic obtained by MfTM has multiple orthogonal 'facets'. For example, the latent topic 'Arab revolutions' may consist of the following six facets: social terms from Twitter (e.g. 'protest', 'omg', 'gather'), auxiliary terms from linked web documents (e.g. 'libya', 'war', 'protest'), person names (e.g. 'Muammar Gaddafi'), organizations (e.g. 'United Nations'), location names (e.g. 'Libya', 'Egypt') and a temporal distribution (cf. Fig. 1).

Parameter inference is a known bottleneck of topic models, in particular in face of the scale of microblog data. We therefore build upon the recent advances in variational inference methods and develop an "online" inference algorithm for MfTM. In contrast to Gibbs sampling or batch variational inference, our algorithm processes data in a streaming fashion. As we show in our evaluation, our inference method easily scales to large datasets and has the advantage of continuous updatability.

Our framework is able to serve two purposes. First, MfTM discovers a set of latent topics from Twitter data. Second, MfTM can be used to transform unstructured text in tweets into a rich multi-faceted topic representation. The representation can be used in further applications, such as clustering, user profiling or recommendation.

In this paper, the performance of MfTM is evaluated on the task of tweet clustering. By comparing against multiple baselines, we demonstrate that MfTM is more suitable to microblog streams than standard topic models.

Our contributions are summarized as follows.

— We propose a Multi-faceted Topic Model (MfTM), which extracts latent topics from semantically-enriched posts and models multiple facets of information associated with each topic: social terms facet, auxiliary terms facet, named entity facets and a temporal distribution. We develop an efficient inference algorithm, which makes our model scalable and updatable.

— We propose a pre-processing method for microblog posts, referred to as Time-sensitive Hashtag-based Semantic Enrichment. The method is used to enrich the semantics of posts containing hashtags, by producing a list of the most significant terms associated with a hashtag and a timestamp. We further propose a detailed parametrization, which considers the recency sensitivity of each hashtag.
— We conduct a detailed evaluation of our framework and study the effectiveness of each component. We evaluate the proposed hashtag-based and web-document-based semantic enrichment methods and show that both methods are important for improving the quality of the discovered topics. On the task of tweet clustering, we demonstrate that our framework is able to outperform multiple baseline topic models.

The rest of the paper is structured as follows: We first discuss related work in Section 2. In Section 3, we present individual components of our framework for multi-faceted topic modeling in Twitter. Section 4 presents our experimental evaluation. Finally, we conclude our findings and discuss relevant issues for future work in Section 5.

2. RELATED WORK

Two categories of work that are most related to this paper are semantic enrichment and topic modeling.

Semantic Enrichment. To tackle the short length of microblog posts, methods for incorporating additional semantics have been proposed. Abel et al. [2011] identify online news articles related to the posts, in order to extract named entities and include them in the user profile. In another approach, posts are mapped to Wikipedia articles and the existing Wikipedia ontology is utilized for categorization [Genc et al. 2011]. In contrast to these approaches, we perform semantic enrichment based on explicit URL links included in posts. We obtain both named entities and also the top-k general terms from the web document. We choose this approach over these methods, since our approach does not require a matching algorithm, thus reducing computational cost.

Hashtags, however, have not been previously used as a basis for semantic enrichment. Some early work has been done to study the tagging behavior in Twitter [Huang et al. 2010], retrieving relevant hashtags given a query [Efron 2010], measuring hashtags' interestingness [Weng et al. 2010] and sentiments associated with hashtags [Wang et al. 2011a]. Groh et a. [2013] doubled each occurrence of a hashtag in a post to strengthen its role as a topical indicator. To our knowledge, our work is the first to model the "social chatter" associated with hashtags and to include such additional semantics into individual posts.

Topic Modeling. Our topic modeling approach is related to previous works on probabilistic topic models. Blei et al. [2003] proposed Latent Dirichlet Allocation (LDA) to analyze electronic archives. The Topics-Over-Time (TOT) model by Wang and McCallum [2006] incorporates temporal information into LDA, by assuming that the timestamps of a topic follow a beta distribution. The topic-aspect model by Paul and Girju [2010] is proposed to model "multi-faceted" topics. In their definition, a "multi-faceted" topic is a topic that is expressed differently across different aspects, such as scientific disciplines. Our focus and definition of multi-faceted topics is therefore fundamentally different. The entity-topic model by Newman et al. [2006] considers two facets of a topic: general terms and entities. In contrast, our model considers general terms and each entity type in a separate facet, as well as modeling timestamps.

In the microblogging environment, Hong and Davison [2010] study approaches to apply LDA on Twitter data. To improve topic modeling in Twitter, the Twitter-LDA model by Zhao et al. [2011] aims to demote frequent non-topical words (e.g., "cool", "today", etc.) by incorporating the global word distribution. Twitter-LDA is then used to compare topics in Twitter and in news articles [Zhao et al. 2011]. The DLDA model by

Jin et al. [2011] aims to improve topic modeling on short documents by incorporating auxiliary semantics. DLDA estimates a set of "target topics" from short documents and a set of "auxiliary topics" from auxiliary documents. In our model, auxiliary semantics are directly integrated into each topic. In addition, our model estimates the importance of auxiliary versus social terms within each topic, thus allowing more fine-grained control over the influence of auxiliary semantics.

Our work also builds upon recent advances in topic model inference, in particular stochastic variational inference [Hoffman et al. 2013]. It enables topic models to be trained on massive and streaming data, since it operates in a sequential rather than batch fashion (such as Gibbs sampling [Walsh 2004]). We adopt this technique to develop an online learning method for MfTM. There have been other approaches to online inference for topic models. In AlSumait et al. [2008] and Lau et al. [2012], online extensions of Gibbs sampling are proposed, which allow updating the LDA model as new data arrives. However, these approaches suffer from scalability issues in the face of real-world dataset scale, since Gibbs sampling is not a streaming algorithm in nature. Our scalability evaluation demonstrates the strength of our online inference method and compares with the online Gibbs sampling method by Lau et al. [2012].

Our preliminary work in Vosecky et al. [2013] studied multi-faceted topic modeling of Twitter content. However, it has not addressed semantic enrichment using social and auxiliary sources and the integration of such semantics into the multi-faceted topic model.

3. PROPOSED FRAMEWORK

In this section, we present our Twitter topic modeling framework as illustrated in Figure 2. We start by discussing data pre-processing methods, which comprise extracting auxiliary semantics from linked web-documents and social terms associated with hashtags. These *semantic enrichment* methods are closely tied to the design of the Multi-faceted Topic Model to improve the quality of the obtained latent topics.

Before discussing the pre-processing methods, we introduce the representation of microblog posts throughout the paper.

DEFINITION 1. *A microblog post is a tuple $p = \langle a_p, T_p, H_p, U_p, \tau_p \rangle$, where $a \in A$ is the author of the post, $T_p \subset T$ is the set of terms, $H_p \subset H$ is the set of hashtags, U_p is the set of URLs and τ_p is the timestamp of the post.*

DEFINITION 2. *An enriched post is a tuple $\hat{p} = \langle a_p, T_{p,soc}, T_{p,aux}, E_p, \tau_p \rangle$, where $T_{p,soc}$ denotes social terms, consisting of T_p and terms inserted during Hashtag-based Semantic Enrichment. $T_{p,aux}$ denotes auxiliary terms inserted during Web-document-based Semantic Enrichment. $E_p = \{E_{p,1}, \ldots, E_{p,X}\}$ is a super-set containing X sets of named entities, each set $E_{p,x}$ containing entities of type x.*

3.1. Web-document-based Semantic Enrichment

URL links contained in posts provide an opportunity to obtain additional semantics from the referred web documents. In our framework, such auxiliary semantics are utilized for topic discovery within the Multi-faceted Topic Model. We utilize the referred web documents in two different ways: (1) to extract the top-k terms from the web document; and (2) to extract named entities. This information is then included in the *enriched post* $\hat{p}$ (cf. Figure 2).

As the first step, we obtain all web documents referred within posts. Second, we represent the web document corpus in the vector-space model using standard TF-IDF term weighting [Manning et al. 2008]. From each web document, we select top-k terms with the highest TF-IDF value. These terms are then included as auxiliary terms $T_{p,aux}$ of the *enriched post* $\hat{p}$ containing the respective URL link u_p. We note that the reason

for including top-k terms from u, rather than the full document, is due to the short length of p. Thus, we avoid the problem of "overwhelming" the short posts with the long auxiliary documents. The value of k can be determined experimentally, as discussed in Section 4.4.3.

Apart from the top-k general terms, we also obtain X types of named entities from the web documents. Named entities of each type are stored separately within the enriched post $\hat{p}$ for further processing, i.e. $E_p = \{E_{p,1}, \ldots, E_{p,X}\}$.

We note that apart from web documents, named entity extraction may also be performed directly on Twitter posts [Ritter et al. 2011]. Such entities may be included into the enriched post $\hat{p}$ in the same manner as above. A comparative study of the importance of named entities extracted from tweets and those extracted from web documents falls outside the scope of this paper and is thus left as an area for future work.

Finally, we note our framework can also accommodate other alternative sources of auxiliary semantics (e.g., Wikipedia). The general procedure to semantic enrichment from an auxiliary domain involves two steps: (1) semantic linkage of a tweet to one or more auxiliary documents, and (2) top-k term extraction from an auxiliary document and their inclusion in the original tweet. The specifics of these two steps vary depending on the domain. Interested readers may refer to relevant work that studies the use of the desired domain (e.g., Genc et al. [2011] for semantic linkage to Wikipedia).

3.2. Time-sensitive Hashtag-based Semantic Enrichment

Hashtags are user-defined tags included in microblog posts, which indicate the discussed topics and enable posts related to the same topics to be searched easily. This facility has recently played a vital role in sharing news and comments related to specific events or entities [Wang et al. 2011; Weng et al. 2010]. However, to our knowledge the use of hashtags as a source of additional semantics in order to improve topic modeling has not been previously explored.

In this paper, we utilize hashtags to insert additional social semantics into tweets, in order to aid the topic discovery task. In essence, we view a hashtag as an agglomeration of "social chatter" associated with a specific theme. For example, by summarizing all posts in the last day that contain the hashtag *#iphone*, we can obtain the "voice of the crowd" for that day, related to the respective mobile phone product. Our goal is to use this information to enrich posts that mention the particular hashtag.

Our hashtag-based semantic enrichment method (*hash-SE*) starts by extracting all hashtags from the corpus. We then construct a document for each hashtag, which includes all posts that contain the respective hashtag. If a post contains multiple hashtags, it will be included in all the respective hashtag documents. We can then represent all hashtag documents by using the vector space model. Following the intuitions behind TF-IDF weighting, let $TF_H(t,h)$ to be the term frequency of t in posts containing hashtag h. Let $IHF(t)$ be the inverse hashtag frequency, calculated using the number of hashtags that co-occur with term t. Formally, we define a simple measure of the significance of a term t in the hashtag document as follows:

DEFINITION 3. *Term significance (TS) of term t w.r.t. hashtag h is defined as:*

$$TS(t,h) = TF_H(t,h) \cdot IHF(t), \tag{1}$$

where $TF_H(t,h) = \big|\{p | t \in T_p \wedge h \in H_p\}\big|, \quad IHF(t) = \log\left(\frac{|H|}{\big|\{h|\exists p : t \in T_p \wedge h \in H_p\}\big|}\right),$

p is a post, T_p is the set of terms in p, H_p is the set of hashtags within p, and H is the set of all hashtags in the corpus.

However, this method does not take into account the time when a term was used. Since topics discussed in microblogs often evolve dynamically, fresh information re-

lated to a particular hashtag is often preferred. Thus, we introduce time-sensitive term frequency $TF_H(t, h, \tau_i)$, where τ_t is the time when term t was used. We use the following intuition when designing the time-sensitive function: a term should have a higher weight if it was used close to a queried timestamp. Conversely, a term should have a lower weight if it was used a long time away from a specific timestamp. Formally, we utilize a *temporal weight function* $\delta_h(\Delta) \in [0, 1]$ for each hashtag h, where $\Delta = |\tau_q - \tau_j|$ is the difference between a queried timestamp τ_q and a tweet's timestamp τ_j. The choice of the function δ_h directly affects the importance of *recency* when calculating term significance. We discuss further details of the function δ_h in Section 3.2.1.

We now formalize three metrics to capture the time-sensitive significance of a term:

DEFINITION 4. *Temporal term frequency (TTF) of term t w.r.t. hashtag h at time τ_i is defined as:*

$$TTF(t, h, \tau_i) = \sum_j TF_H(t, h, \tau_j) \cdot \delta_h(|\tau_i - \tau_j|). \tag{2}$$

DEFINITION 5. *Temporal hashtag frequency (THF) of term t at time τ_i is defined as:*

$$THF(t, \tau_i) = \sum_j HF(t, \tau_j) \cdot \delta_h(|\tau_i - \tau_j|). \tag{3}$$

DEFINITION 6. *Temporal term significance (TTS) of term t w.r.t. hashtag h at time τ_i is defined as:*

$$TTS(t, h, \tau_i) = TTF(t, h, \tau_i) \cdot \log\left(\frac{|H|}{THF(t, \tau_i)}\right). \tag{4}$$

We note that due to practical reasons, it may not be possible to pre-compute the TTS scores for each time instant separately. A more computationally affordable solution is to define a suitable time interval and discretize posts' timestamps. TTS scores are then pre-computed for each interval and stored for further usage. Finally, the top-k terms for each hashtag are selected for each time interval based on the TTS value.

As a final step, we include the top-k terms based on the post's timestamp and hashtag h into the enriched post $\hat{p}$. The post's original terms T_p and the newly added hashtag enrichment terms T_{h,τ_p} together form the *social terms* of $\hat{p}$, i.e. $T_{p,soc} = T_p \cup T_{h,\tau_p}$.

3.2.1. Recency Sensitivity of a Hashtag. To control how the importance of terms associated with a particular hashtag h decreases over time, we utilize a *temporal weight function* $\delta_h(\Delta) \in [0, 1]$ for hashtag h and time period Δ (cf. Equations (2)-(4)). Intuitively, different hashtags may have a different *recency sensitivity* (e.g., hot trends vs. long-term trends)[2]. In our work, we propose to model the recency sensitivity of content related to hashtag h as a gaussian, $\mathcal{N}(\mu, \sigma_h^2)$. For a particular term t posted at time τ_i and co-occurring with hashtag h, we model how t's importance decreases over time by a gaussian $\mathcal{N}(\tau_i, \sigma_h^2)$. The temporal weight function $\delta_h(\Delta)$ is then defined as

$$\delta_h(\Delta) = p\big(\Delta \big| \mathcal{N}(\mu = 0, \sigma_h^2)\big), \tag{5}$$

where Δ is a relative time period.

The most important issue is how to determine the temporal variance σ_h^2 for a particular hashtag h. Let us first illustrate our idea with three examples of hashtags that we empirically observe in microblogs. First, *event-related hashtags* become popular for a limited period of time around a particular event (e.g., *#london2012* denoting the London Olympics 2012). Content related to these hashtags includes timely updates and commentary related to the event, hence such content is highly recency-sensitive.

[2]For example, for news-related topics, fresh terms are important and terms older than 1 month may be considered outdated.

Date	Top 10 terms for #irene
2011/08/22	hurrican rico iren puerto intermedi gradual season prepar advisori gov
2011/08/23	hurrican carolina unexpect prepared hq rattl cross app download readi
2011/08/25	evacu plan app west offici cross center local red disast
2011/08/30	donat rescu anim dog night work vermont power ny pet
2011/09/03	outag statu aug insid special outer rever nearest paterson yorker
2011/09/04	presid respons paterson tour ceo op gail volunt mcgovern continu

(a)

τ	2011/08/30 12:08
T_P	RT @cnnbrk: Hurricane #Irene death toll raised to 36.

(b)

τ	2011/08/30 12:08
T_{Soc}	RT @cnnbrk Hurricane #Irene death toll raised to 36 donat rescu anim dog night work vermont power ny pet

(c)

Fig. 3: (a) Top 10 most significant terms for hashtag *#irene*, (b) an example post, (c) example enriched post after hashtag-based enrichment.

Second, we consider *long-term popular hashtags*, such as hashtags related to popular named entities (e.g., *#Obama* or *#London*) Although such hashtags may be popular over a long period of time, content tagged using such hashtags may still be highly recency-sensitive. In this case, however, the recency-sensitivity is not directly observable from the hashtag's trending behavior[3]. Third, *daily activity hashtags* are used in microblogs to denote one's daily activities or feelings (e.g., *#AtWork*, *#ThankGodItsFriday*). Content related to such hashtags may still provide useful semantics for topic discovery, since our hashtag enrichment method essentially extracts a "summary" of such hashtags. However, the recency of such content may be less important.

We note that the above examples do not cover all types of hashtags that appear in microblogs. However, they show that analyzing a hashtag's trending behavior alone is not sufficient to model the hashtag's recency sensitivity (e.g., metrics used in Huang et al. [2010]). Moreover, we cannot rely on supervised methods to determine recency sensitivity due to the dynamic and evolving nature of microblog content. Therefore, we propose a novel method to determine a hashtag's recency sensitivity based on users' re-sharing activity of URL links.

We base our approach on the observation that microblog users often share URL links along with a hashtag. In our Twitter dataset, nearly 50% of tweets containing a hashtag also contain a URL (cf. Section 4.1). For each URL link shared in Twitter, we may obtain a temporal distribution of its re-sharing activity. We then utilize the URL's re-sharing activity to model the importance of recency related to the included hashtag.

Formally, let $\mathcal{T}_u = \{\tau_1, \tau_2, \ldots, \tau_{|\mathcal{T}_u|}\}$ be the set of timestamps of posts mentioning a URL u. We assume that the timestamps follow a gaussian, $\tau \sim \mathcal{N}(\mu_u, \sigma_u^2)$. Here, σ_u^2 represents the temporal variance of sharing activity for u. A small variance indicates a high recency-sensitivity of u (e.g., sharing of a news article), while a large variance indicates content shared over a long period of time (e.g., sharing of a link to a company's official website). Now let U_h denote the set of all URLs that co-occur with hashtag h. We may then determine h's recency sensitivity as a weighted average of variances of all URLs in U_h,

$$\sigma_h^2 = \frac{1}{|\sum_{u \in U_h} w_u|} \sum_{u \in U_h} \sigma_u^2 \cdot w_u, \qquad w_u = \frac{|\mathcal{T}_u|}{\sum_{v \in U_h} |\mathcal{T}_v|}. \tag{6}$$

The weight w_u of URL u reflects the popularity of u by the number of times u is shared. The obtained recency sensitivity of h may then be directly used in Equation 5. Finally, we also need to account for hashtags which do not co-occur with any URLs. For these hashtags, we utilize the global average variance σ_G^2, computed as the average variance of all hashtags from the corpus.

An example of hashtag-based enrichment is shown in Figure 3. In the example, we extract the top 10 significant terms for the hashtag *#irene*, which refer to a hurricane that hit the Carribean and the United States East Coast in late August 2011.

[3]On the one hand, news related to a celebrity may be highly recency-sensitive. On the other hand, information about a travel destination may not be as recency-sensitive.

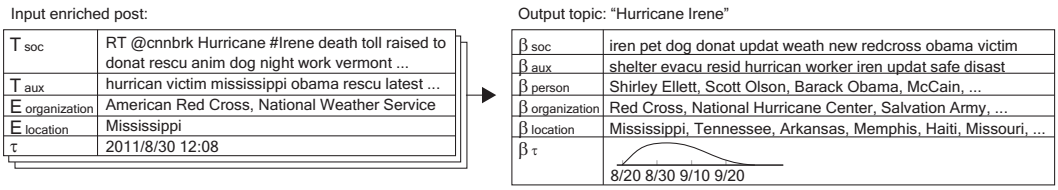


Fig. 4: Example of an enriched post and an output topic from MfTM

We note that there is potential to develop more detailed techniques for each aspect of hashtag-based semantic enrichment. We summarize the main directions for future work as follows. First, we may detect and remove low-quality (e.g., spam) hashtags to avoid the inclusion of noise. Second, the ambiguity of some hashtags may be tackled by considering the context(s) in which they are used. For example, when *#Russia* is used in the context of "news reports", its recency sensitivity could be modeled separately from the context of "travel". Third, the recency sensitivity of long-term hashtags may change over time and thus, could be dynamically adjusted. Fourth, if a hashtag does not occur with any URL, we may refine how recency sensitivity is assigned. To this end, we may consider the hashtag's context (e.g., semantic context or location context) to infer its likely recency sensitivity.

3.3. Multi-faceted Topic Model

After obtaining the enriched posts (cf. Definition 2), we propose the *Multi-faceted Topic Model* (MfTM) to discover rich latent topics from the corpus. Traditional topic models, such as Latent Dirichlet Allocation (LDA) [Blei et al. 2003], can be used to learn a set of latent topics from a document corpus. Each topic in LDA is a multinomial distribution over words. In contrast, we aim to model latent topics with finer granularity, such as preference towards specific entities (e.g., specific locations) and the topic's temporal characteristics. Moreover, we integrate auxiliary semantics from linked web documents into each topic. Each of these types of information forms a separate *facet* of a topic. Figure 4 illustrates an enriched post as the input of MfTM and an example latent topic as its output. We will now present the design of MfTM in detail.

We start by discussing how microblog-specific semantics and auxiliary semantics are integrated in MfTM. In general, topic models assume that each document exhibits one or more latent topics (e.g., a post by Barack Obama may relate to 'politics' and 'economy'). Jin et al. [2011] aim to integrate auxiliary information into the topic model, assuming that there are two distinct types of topics: (1) topics learned from the target dataset (i.e., a Twitter corpus), and (2) topics learned from auxiliary data sources (e.g., related news articles). Each document may then exhibit both types of topics. However, our intuition is that a tighter integration of auxiliary semantics would benefit the model. We also aim to avoid topic duplication (e.g., topic 'politics' discovered twice, from Twitter and from auxiliary data). Therefore, instead of a *topic-level* integration, we propose to integrate auxiliary semantics on a finer level, namely on the *word-level*.

Specifically, we assume that each topic is associated with two facets: (1) *social terms* (t_{soc}), which originate from Twitter users, and (2) *auxiliary terms* (t_{aux}), which originate from external documents that are linked from Twitter posts. As an example, let us consider a latent topic 'politics'. On the one hand, Twitter users' comments about politics, including opinion words and slang expressions, may form the *social terms* facet of this topic. On the other hand, words from news articles that report about politics may form the *auxiliary terms* facet of this topic.

When a Twitter user composes a new post, MfTM assumes that she makes the following decisions when writing each word. First, she chooses a topic z to write about.

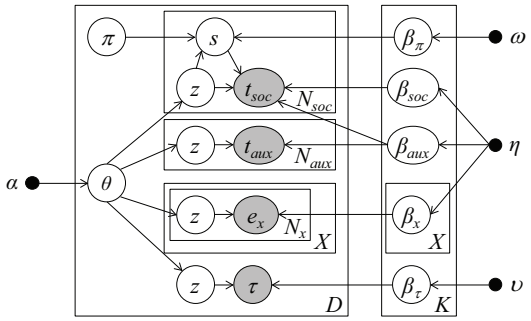


Fig. 5: Graphical Model of MfTM

Second, she makes a binary decision whether to use social or auxiliary terms. We note that each topic has an associated Bernoulli distribution, which influences whether social or auxiliary terms are preferred[4]. Finally, the user chooses a term from the chosen facet of topic z. This procedure aims to mimic user behavior in real situations. For example, a user may read a news article and choose to summarize its main message in a Twitter post (i.e., auxiliary terms are preferred). In another situation, the user may choose to write an opinion about the news event (i.e., social terms are preferred).

Next, MfTM assumes the existence of X entity types in the corpus. For example, the entity types may include *person* (e_p), *organization* (e_o) and *location* (e_l) entities. In MfTM, each entity type follows a multinomial distribution given a latent topic. Each entity type thus forms an additional facet of a topic. Lastly, we also capture the trending behavior of topics by modeling the timestamps of posts as a continuous Beta distribution. As we show in our evaluation, each facet of MfTM provides useful evidence and improves the effectiveness of the model (cf. Sections 4.4.3 and 4.4.5).

Figure 5 illustrates the structure of the model. The generative process of MfTM is summarized in Algorithm 1. Given the hyperparameters $\alpha, \eta, \upsilon, \omega$, the joint distribution of topics β, document-topic mixtures θ, topic assignments $\mathbf{z}$, switch variables $\mathbf{s}$, terms $\mathbf{t}$, entities $\mathbf{e}$ and timestamps $\mathcal{T}$ is given by:

$$\begin{aligned} p(\theta, \beta, \mathbf{z}, \mathbf{s}, \mathbf{t}, \mathbf{e}, \mathcal{T} | \alpha, \eta, \upsilon, \omega) = & p(\theta|\alpha) p(\beta|\eta) p(\beta_\pi|\omega) p(\beta_\tau|\upsilon) p(\mathbf{z_{soc}}|\theta) \times \\ & \textstyle\sum_s p(s|\beta_\pi) p(\mathbf{t_{soc}}|\mathbf{z_{soc}}, \beta_{soc})^{1-s} p(\mathbf{t_{soc}}|\mathbf{z_{soc}}, \beta_{aux})^{s} \times \\ & p(\mathbf{t_{aux}}|\mathbf{z_{aux}}, \beta_{aux}) \textstyle\prod_{x=1}^{X} p(\mathbf{z_x}|\theta) p(\mathbf{e_x}|\mathbf{z_x}, \beta_x) p(\mathbf{z}_\tau|\theta) p(\mathcal{T}|\mathbf{z}_\tau, \beta_\tau). \end{aligned} \tag{7}$$

Since exact inference for this model is intractable, an approximate posterior inference method is needed to estimate the latent parameters. Although Gibbs sampling [Walsh 2004] is a widely adopted inference method for topic models, an online learning method for LDA, namely *stochastic variational inference* (SVI), has been developed recently [Hoffman et al. 2013]. In stochastic optimization, we find the maximum of the variational objective by following noisy estimates of its natural gradient. SVI enables parameter inference on massive and streaming data, since it operates in a sequential, rather than batch fashion. This inference method fits well in the scenario of analyzing microblog posts, which essentially arrive in a streaming fashion. We use SVI as a basis to develop an online learning method for MfTM.

3.3.1. Online Inference for MfTM. We now proceed to present our online inference algorithm for MfTM. Due to space constraints, we only present the major components of the algorithm. Interested readers may refer to Hoffman et al. [2013] for full details of

[4]E.g., in topic 'business', auxiliary terms may be preferred. In topic 'holidays', social terms may be preferred.

Algorithm 1 Generative Process of MfTM

for each topic $k \in \{1, \ldots, K\}$ **do**
  Draw the social terms facet: $\beta^k_{soc} \sim \text{Dir}_{V_{soc}}(\eta)$.
  Draw the auxiliary terms facet: $\beta^k_{aux} \sim \text{Dir}_{V_{aux}}(\eta)$.
  Draw distribution over social/auxiliary switches: $\beta^k_{\pi} \sim \text{Bernoulli}(\omega)$.
  for each entity facet $x \in \{1, \ldots, X\}$ **do**
    Draw facet x of topic k: $\beta^k_x \sim \text{Dir}_{V_x}(\eta)$.
  end for
  Draw the temporal facet of topic k: $\beta^k_{\tau} \sim \text{Beta}(\upsilon)$.
end for
for each document $d \in \{1, \ldots, D\}$ **do**
  Draw document's topic distribution $\theta_d \sim \text{Dir}_K(\alpha)$.
  for each social term at position $n \in \{1, \ldots, N_{soc}\}$ **do**
    Draw topic assignment $z_{d,soc,n} \sim \text{Mult}_K(\theta_d)$.
    Draw switch variable $s \sim \text{Bernoulli}(\beta^z_{\pi})$.
    if $s = 0$ **then**
      Draw term $t_{d,soc,n} \sim \text{Mult}_{V_{soc}}(\beta^z_{soc})$.
    else
      Draw term $t_{d,soc,n} \sim \text{Mult}_{V_{aux}}(\beta^z_{aux})$.
    end if
  end for
  for each auxiliary term at position $n \in \{1, \ldots, N_{aux}\}$ **do**
    Draw topic assignment $z_{d,aux,n} \sim \text{Mult}_K(\theta_d)$. Draw term $t_{d,aux,n} \sim \text{Mult}_{V_{aux}}(\beta^z_{aux})$.
  end for
  for each entity facet $x \in \{1, \ldots, X\}$ **do**
    for each element position $n \in \{1, \ldots, N_x\}$ **do**
      Draw topic assignment $z_{d,x,n} \sim \text{Mult}_K(\theta_d)$. Draw element $e_{d,x,n} \sim \text{Mult}_{V_x}(\beta^z_x)$.
    end for
  end for
  Draw topic assignment of timestamp $z_{d,\tau} \sim \text{Mult}_K(\theta_d)$. Draw timestamp $\tau_d \sim \text{Beta}(\beta^k_{\tau})$.
end for

Table I: Complete conditionals of MfTM

Local hidden variables:
$p(z_{d,soc,n}\|\theta_d, \beta, t_{d,soc,n}) \propto \exp\{\log\theta_d + \log\beta_{soc,t}(1 - s_{d,n}) + \log\beta_{aux,t}s_{d,n}\}$
$p(s_{d,soc,n}\|\beta^k_{\pi}, \beta^k_x, t_{d,soc,n}, z_{d,x,n} = k) = \dfrac{\beta^k_{\pi a} + \beta^k_{aux,t}}{\beta^k_{\pi a} + \beta^k_{aux,t} + \beta^k_{\pi b} + \beta^k_{soc,t}}$
$p(z_{d,x,n} = k\|\theta_d, \beta_x, e_{d,x,n}) \propto \exp\{\log\theta^k_d + \log\beta^k_{x,e}\}$, where $x \in X \cup \{aux\}$
$p(z_{d,\tau} = k\|\theta_d, \beta_{\tau}, \tau_d) \propto \exp\{\log\theta^k_d + \log p(\tau_d\|\text{Beta}(\beta^k_{\tau,a}, \beta^k_{\tau,b}))\}$
$p(\theta_d\|\beta, \mathbf{z_d}) = \text{Dir}\Big(\alpha + \sum_{z\in\mathbf{z_d}} z\Big)$, where $\mathbf{z_d} = \{z_{d,x}\|x \in \{soc, aux\} \cup X \cup \tau\}$
Global hidden variables:
$p(\beta^k_f\|\mathbf{z_f}, \mathbf{t_f}) = \text{Dir}\Big(\eta + \sum_{d=1}^{D}\sum_{n=1}^{N_{d,f}} z^k_{d,f,n}t_{d,f,n}\Big)$, where $f \in \{soc, aux\}$ and $\mathbf{z_f}$ is the set of topic assignments of all terms $\mathbf{t_f}$ within facet f.
$p(\beta^k_x\|\mathbf{z_x}, \mathbf{e_x}) = \text{Dir}\Big(\eta + \sum_{d=1}^{D}\sum_{n=1}^{N_{d,x}} z^k_{d,x,n}e_{d,x,n}\Big)$, where $x \in X$
$p(\beta^k_{\tau}\|\mathbf{z}_{\tau}, \mathcal{T}) = \text{Beta}\Big(\upsilon_a + \mu_k(\frac{\mu_k(1-\mu_k)}{\sigma^2_k} - 1)\ ,\ \upsilon_b + (1-\mu_k)(\frac{\mu_k(1-\mu_k)}{\sigma^2_k} - 1)\Big)$, where μ_k is the sample mean and σ^2_k is the sample variance of timestamps assigned to topic k.
$p(\beta^k_{\pi}\|\mathbf{s}^k) = \text{Beta}\Big(\omega_a + \sum_{s\in\mathbf{s}^k} s\ ,\ \omega_b + \sum_{s\in\mathbf{s}^k}(1-s)\Big)$, where $\mathbf{s}^k$ is the set of switch variables of all social terms assigned to topic k.

Table II: Variational parameters and relevant expectations

Variable	Type	Var. param.	Relevant expectations
$\beta_f^k \ \ (f \in \{soc, aux\})$	Dirichlet	λ_f^k	$\mathbb{E}[\log \beta_{f,v}^k] = \Psi(\lambda_{f,v}^k) - \sum_{y=1}^{V_f} \Psi(\lambda_{f,y}^k)$
$\beta_x^k \ \ (x \in X)$	Dirichlet	λ_x^k	$\mathbb{E}[\log \beta_{x,v}^k] = \Psi(\lambda_{x,v}^k) - \sum_{y=1}^{V_x} \Psi(\lambda_{x,y}^k)$
β_π^k	Beta	λ_π^k	$\mathbb{E}[\log \beta_\pi^k] = \Psi(\lambda_{\pi a}^k) - \Psi(\lambda_{\pi a}^k + \lambda_{\pi b}^k)$
β_τ^k	Beta	λ_τ^k	$\mathbb{E}[\log \beta_\tau^k] = \Psi(\lambda_{\tau a}^k) - \Psi(\lambda_{\tau a}^k + \lambda_{\tau b}^k)$
θ_d	Dirichlet	γ_d	$\mathbb{E}[\log \theta_d^k] = \Psi(\gamma_d^k) - \sum_{j=1}^{K} \Psi(\gamma_d^j)$
$z_{d,x,n}$	Multinomial	$\phi_{d,x,n}$	$\mathbb{E}[z_{d,x,n}^k] = \phi_{d,x,n}^k$
$s_{d,n}$	Bernoulli	$\varphi_{d,n}$	$\mathbb{E}[s_{d,n}] = \varphi_{d,n}$

Algorithm 2 Stochastic Variational Inference for MfTM

Initialize $\lambda^{(0)}$ randomly. Initialize $\gamma^{(0)} = \alpha$.
repeat
  Sample a document d from the data set.
  Initialize intermediate local topic proportion $\hat{\gamma}_d = \theta_d$.
  repeat
    for each $t_{d,soc,n}$, $n \in \{1, \ldots, N_{soc}\}$ **do**
      $\varphi_{d,soc,n} = \frac{\lambda_{\pi a}^k + \lambda_{aux,t}^k}{\lambda_{\pi a}^k + \lambda_{aux,t}^k + \lambda_{\pi b}^k + \lambda_{soc,t}^k}$.
      $\phi_{d,soc,n}^k \propto \exp\{\mathbb{E}[\log \theta_d^k] + \mathbb{E}[\log \beta_{soc,t}^k](1 - \mathbb{E}[s_{d,soc,n}]) + \mathbb{E}[\log \beta_{aux,t}^k]\mathbb{E}[s_{d,soc,n}]\}$.
    end for
    for $f \in X \cup \{aux\}$ **do**
      for each $e_{d,f,n}$, $n \in \{1, \ldots, N_f\}$ **do**
        Set $\phi_{d,f,n}^k \propto \exp\{\mathbb{E}[\log \theta_d^k] + \mathbb{E}[\log \beta_{f,e}^k]\}$.
      end for
    end for
    Set $\phi_{d,\tau}^k \propto \exp\{\mathbb{E}[\log \theta_d^k] + \log p(\tau_d | \lambda_\tau^k)\}$, $\quad \gamma_d = \alpha + \sum_{f=1}^{F} \sum_{n=1}^{N_f} \phi_{d,f,n}$.
  until γ_d converges.
  for $k \in \{1, \ldots, K\}$ **do**
    Set intermediate topic k:
      $\hat{\lambda}_f^k = \eta + D \sum_{n=1}^{N_f} \phi_{d,f,n}^k e_{d,f,n}$ for facet $f \in X \cup \{soc, aux\}$
      $\hat{\lambda}_{\pi a}^k = \omega_a + D \sum_{s \in \mathbf{s}^k} \mathbb{E}[s_{d,soc,n}]$, $\quad \hat{\lambda}_{\pi b}^k = \omega_a + D \sum_{s \in \mathbf{s}^k} (1 - \mathbb{E}[s_{d,soc,n}])$.
      $\hat{\lambda}_{\tau a}^k = \upsilon_a + D\ \mu_k \left(\frac{\mu_k (1 - \mu_k)}{\sigma_k^2} - 1 \right)$, $\quad \hat{\lambda}_{\tau b}^k = \upsilon_b + D\ (1 - \mu_k) \left(\frac{\mu_k (1 - \mu_k)}{\sigma_k^2} - 1 \right)$.
    Update global topic k:
      $\lambda_x^{k\,(i+1)} = (1 - \rho^{(i)}) \lambda_x^{(i)} + \rho^{(i)} \hat{\lambda}_x^k$ for facet $f \in X \cup \{soc, aux\}$.
      $\lambda_{\pi a}^{k\,(i+1)} = (1 - \rho^{(i)}) \lambda_{\pi a}^{k\,(i)} + \rho^{(i)} \hat{\lambda}_{\pi a}^k$, $\quad \lambda_{\pi b}^{k\,(i+1)} = (1 - \rho^{(i)}) \lambda_{\pi b}^{k\,(i)} + \rho^{(i)} \hat{\lambda}_{\pi b}^k$.
      $\lambda_{\tau a}^{k\,(i+1)} = (1 - \rho^{(i)}) \lambda_{\tau a}^{k\,(i)} + \rho^{(i)} \hat{\lambda}_{\tau a}^k$, $\quad \lambda_{\tau b}^{k\,(i+1)} = (1 - \rho^{(i)}) \lambda_{\tau b}^{k\,(i)} + \rho^{(i)} \hat{\lambda}_{\tau b}^k$.
  end for
until forever

SVI. We begin by listing the complete conditionals of the model in Table I. For convenience, we assume that social terms and auxiliary terms share the same vocabulary, $V_{soc} = V_{aux}$, constructed as a union of both vocabularies.

In the next step, we summarize parameters of the variational distributions and expected sufficient statistics in Table II. The full inference algorithm including update equations for the variational parameters is presented in Algorithm 2. When processing each document, the algorithm repeatedly updates the local variational parameters γ, ϕ, φ until convergence. After fitting the local parameters, we set each facet of the intermediate topics based on the local parameters. Finally, the intermediate topics are interpolated with the global topics, using the parameter $\rho^{(i)} = (i + \overline{\tau})^{-\kappa}$. The parame-

ter $\kappa \in (0.5, 1]$ is the *forgetting rate*, which controls the weight of fresh content in the model. The *delay* $\overline{\tau} \geq 0$ is used to demote early iterations.

After applying MfTM to a document corpus, we may obtain the topic vector θ_p of a new post p given the latent topics as

$$\theta_p = \alpha + \prod_{n=1}^{N_{soc}} t_{p,soc,n}\ \beta_{soc,t}^{1-\mathbb{E}[\beta_\pi]}\ t_{p,aux,n}\ \beta_{aux,t}^{\mathbb{E}[\beta_\pi]} \prod_{n=1}^{N_{aux}} t_{p,aux,n}\ \beta_{aux,t} \prod_{x=1}^{X} \prod_{n=1}^{N_x} e_{p,x,n}\ \beta_{x,e}\ p(\tau_p|\beta_\tau). \quad (8)$$

4. EVALUATION

In this section, we first describe our evaluation datasets and methodology. Second, we evaluate the proposed online inference algorithm of MfTM and discuss its scalability. Third, we study the effectiveness of our framework on the task of tweet clustering and evaluate the utility of our semantic enrichment components. Clustering quality is further compared against multiple baseline methods. We note that interested readers may refer to our online Appendix [Vosecky 2014] for detailed experimental results.

4.1. Dataset Collection

Due to the lack of a standard dataset for conducting experiments with microblog data, we collected publicly accessible data from Twitter to conduct our experiments. To access Twitter, we utilized the Twitter REST API[5]. In particular, we constructed three evaluation datasets, as described in this section. First, a background Twitter dataset is collected for the purpose of topic modeling. Then, we prepared two labeled datasets to conduct our clustering experiments. Both datasets for clustering span a subset of the time period spanned by the background Twitter dataset.

4.1.1. Background Twitter Dataset. This dataset provides the basis for building the proposed MfTM and baseline topic models. However, due to their inference algorithms, the efficiency of the baseline models is limited and thus, it is not practical to learn them on massive-scale datasets. This poses a limitation on the size of our background corpus. At the same time, our goal is to collect a representative sample of topics in Twitter. To achieve this, we focus both on topics discussed by popular Twitter users and by the general public. On the one hand, popular users are considered since they have a large influence within Twitter and can be selected from diverse topical categories to ensure topical diversity. On the other hand, tweets by the general public constitute the majority of Twitter content and thus, the "voice of the crowd" utilized by hashtag-based semantic enrichment (cf. Section 3.2). We now describe both parts of the dataset.

Popular users. Our basic approach to crawl users that cover diverse topical categories is inspired by previous work [Efron and Golovchinsky 2011; Duan et al. 2010; Rosa et al. 2010]. We selected an initial set of 50 seed users from Listorius[6], a web-based service that categorizes popular Twitter users into various topical categories. The users are randomly selected from 5 different categories (technology, business, politics, celebrities and activism) to ensure topical diversity. Starting with these seed users, we crawled Twitter users' posts in a breadth-first search manner by traversing the followee graph. For each user, we selected the top-20 followees to add to the crawl queue. The followee selection criteria is based on the number of times the user has retweeted or mentioned the followee. We crawl a limited number of followees, since some popular users have a large number of followees, which would create a bias towards certain user categories. Also, a user usually only interacts with very few followees that are of highest interest. Since we are also interested in modeling the temporal characteristics of topics, we crawl up to 1,000 recent posts per user in order to cover a longer time period. In total, this dataset part contains 328,428 tweets by 1,874 users.

[5]https://dev.twitter.com/
[6]http://www.listorious.com

Table III: Twitter dataset statistics

Dataset	Background	ML	HL
No. of tweets	2,126,899	1,542	7,901
No. of users	2,574	1,059	326
% of tweets w/ URL	44.8%	53%	74.3%
% of tweets w/ hashtag	21.1%	24.6%	100%
% of tweets w/ URL and hashtag	11.7%	14.9%	74.3%
% of tweets w/ URL or hashtag	54.4%	62.7%	100%
% of tweets w/ named entities (NE)	38.2%	68.9%	58.1%
... % of tweets with "person" NE	39.4%	52.9%	23.5%
... % of tweets with "organization" NE	49.8%	49.7%	44.1%
... % of tweets with "location" NE	29.4%	50%	34.4%

General users. Twitter's Streaming API[7] provides a sample of the full public Twitter stream. We monitored the stream for one day in April 2013 and selected users who posted English-language tweets and had at least 3,000 posts in total. For each of these users, we crawled up to 3,000 tweets in order to cover a longer time period. In total, this dataset part contains 1,798,471 tweets by 700 users and spans a time period from January 2009 to April 2013.

Additionally, we also extracted all URL links contained in the Twitter corpus and crawled the corresponding web pages. In total, 522,920 web pages were retrieved.

Table III shows the statistics of the background dataset. A high-level analysis has shown that nearly $\frac{1}{2}$ of tweets in our dataset contain URL links and 1 out of 5 tweets contains a hashtag. Performing both web document-based SE and hashtag-based SE is applicable to over $\frac{1}{2}$ of tweets in our dataset. Named entities have been identified for nearly 40% of tweets, showing that our multi-faceted model is applicable to a large proportion of tweets in our dataset.

4.1.2. Clustering Evaluation Datasets. To conduct our clustering experiments, we construct two evaluation datasets. We note that both datasets provide complementary characteristics for our evaluations, both in terms dataset size, time period covered and method of obtaining ground truth labels. This evaluation approach has also been applied in recent work [Tsur and Rappoport 2013].

Manually Labeled Dataset (ML). To construct this dataset, we invite three human reviewers. Each reviewer is asked to choose at least 9 queries to be submitted to the Twitter search engine. For each query, we crawl the top 50 tweets returned by Twitter, corresponding to 1 *page* of Twitter search results. Each reviewer is then asked to read through the list of tweets and assign a topic label to each tweet. Each topic label is a short free-form phrase that describes the main story of the tweet. For example, *"iPhone 5S launch"* may be the topic label for tweet *"Rumor: iPhone 5S to launch June 20, just 8 months after iPhone 5..."*. The reviewers are asked to use a consistent set of topic labels when reviewing the list of tweets for a query. Notably, we choose to use free-form labels over a pre-defined taxonomy. This is mainly because of the diversity and evolving nature of topics in Twitter. In total, we obtained 1,524 labeled tweets for 32 queries, with an average of 47.6 tweets per query. The tweets' topic labels serve as the ground truth when evaluating clustering quality. Based on the topic labels, there are 9.4 ground truth clusters for each query on average.

Hashtag Labeled Dataset (HL). To obtain a larger dataset for comparison of clustering performance, we utilize hashtags in tweets as topic labels. We make use of the fact that Twitter users include hashtags in their tweets to indicate the tweet's topic. Recent work has already utilized hashtags to create labeled test collections for clus-

[7] https://dev.twitter.com/docs/streaming-apis

tering [Jin et al. 2011; Rosa et al. 2010; Tsur and Rappoport 2013]. We first extract 100 most frequent hashtags from our background dataset. We then divide them into 10 batches, each batch containing 10 hashtags. For each hashtag in a batch, we select tweets containing the respective hashtag from the background dataset. Before applying clustering, all hashtags are removed from the tweets to be clustered. Our clustering goal in then to place tweets containing the same hashtag into the same cluster. In total, dataset HL contains 7,901 tweets, each batch containing 790 tweets on average.

Table III summarizes the high-level statistics of both datasets. We note that both clustering datasets span a time period which is a subset of the time period spanned by the background Twitter dataset.

4.2. Modeling Phase

4.2.1. Normalization. We start our pre-processing by normalizing the posts' content. First, posts are converted to lower-case, punctuation and numbers are removed and characters repeated consecutively more than twice are stripped, in order to correct basic misspellings (e.g. the string "goooood" will be converted to "good"). Second, URL links are stored separately for further use and removed from the post. Third, stopwords are removed and all terms are stemmed. Usernames and hashtags contained in the post are retained.

4.2.2. Semantic Enrichment. During web document-based semantic enrichment, we first extract all URL links from the posts in our corpus and crawl the respective web documents. Second, we perform named entity recognition (NER) using the Stanford NER library[8]. In general, our framework is able to accommodate an arbitrary number of named entity types. In this paper, we extract *person*, *organization* and *location* named entities. Apart from web documents, we note that named entities can also be extracted directly from microblog posts. However, the short and informal nature of microblog content results in a poor accuracy of conventional NER tools [Ritter et al. 2011]. In principle, tweet-based named entity extraction can be seamlessly integrated into our framework with the availability of appropriate NER tools.

During hashtag-based semantic enrichment, timestamps of posts are discretized into day intervals. Discretization by day is chosen since a day is a commonly used unit for organizing news and social content (i.e., news articles, blog posts, events documented in Wikipedia, etc.).

We produce several dataset versions with different numbers of *social* and *auxiliary* terms injected during semantic enrichment. Each dataset version includes general terms, extracted named entities and timestamps, so that all facets of MfTM are utilized. The dataset versions differ only with respect to the social and auxiliary terms. We first produce 5 dataset versions with 10, 20, 30, 40 and 50 auxiliary terms per post, respectively. No hashtag-based SE is performed on these datasets. Second, we fix the number of auxiliary terms to be 10 and produce dataset versions with 10, 20, 30, 40 and 50 social terms inserted by hashtag-based SE. These datasets are then used for building different versions of MfTM and evaluating the influence of SE on the models.

Regarding our clustering datasets (i.e., ML and HL), we only perform named entity extraction. No additional auxiliary or social terms are included in the clustered tweets.

4.2.3. Topic Modeling. To prepare training of MfTM, we collect the enriched posts (cf. Sections 3.1 and 3.2) from each user. It has been shown in [Hong and Davison 2010] that grouping all posts of a user as a single document produces more accurate topic models compared with treating each post as a separate document. In our work, all user's posts published during the same day are grouped as a document. The resulting *user-day documents* thus have timestamps discretized into day-intervals.

[8]http://nlp.stanford.edu/ner/

We set the hyperparameters for MfTM in accordance with common practice in topic modeling, $\alpha = \eta = 1/K$, $\omega = \upsilon = (1, 1)$. To select suitable values for the parameters κ and $\overline{\tau}$ in stochastic variational inference, we performed a series of experiments with $K = 50$. We vary each parameter while keeping the others fixed and observe the per-word perplexity of the model (cf. Equation 9). Finally, we set $\kappa = 0.7$ and $\overline{\tau} = 4$.

In addition to training MfTM by means of stochastic variational inference, we also implement a Gibbs sampler for MfTM for comparison. Due to space constraints, we omit the details of the Gibbs sampling procedure. When training both models, we apply our Gibbs sampler on a reduced dataset of 320,000 posts due to longer training time required by the Gibbs sampling procedure.

4.3. Topic Model Inference Evaluation

In this section, we analyze the inference of MfTM from two perspectives. First, we compare the proposed online inference algorithm with standard Gibbs sampling inference. Second, we evaluate the scalability of the online inference algorithm.

4.3.1. Comparison of Online Inference and Gibbs Sampling. Perplexity is a standard metric to evaluate the topic model's capability of predicting unseen data [Rozen-Zvi et al. 2004]. After training the model on the training dataset, we compute the perplexity of heldout data to evaluate the model. A lower perplexity score indicates better generalization performance of the model. Specifically, we calculate the average per-word perplexity of heldout data by the following equation:

$$Perplexity(D_{test}|\mathcal{M}) = exp(-\frac{\sum_{d \in D_{test}} \log p(\overrightarrow{w}_d|\mathcal{M})}{\sum_{d \in D_{test}} N_d}), \tag{9}$$

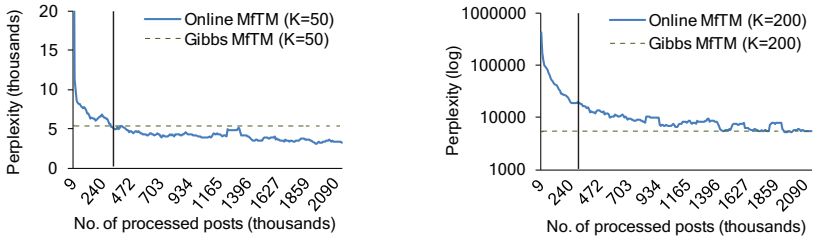
where $\mathcal{M}$ is the model learned from the training dataset, $\overrightarrow{w}_d$ is the word vector for document d and N_d is the number of words in d. We use perplexity to compare the performance of online inference for MfTM and a traditional inference method using Gibbs sampling (GS). For both inference algorithms, we follow common practice to calculate perplexity [Blei et al. 2003; Hoffman et al. 2013]. In the case of Gibbs sampling, the heldout dataset consists of a random 10% sample of the dataset. During online inference, perplexity is calculated using a sliding window of 1000 recent documents.

To illustrate the differences when using GS and our inference algorithm to train MfTM, we show the change in perplexity during the online learning of MfTM in Figures 6 (a) and 6 (b). The dotted line indicates the final perplexity after 1,000 iterations of GS on the dataset. When $K = 50$, we observe that online inference is able to reach the perplexity of the GS-learned model only after processing 200,000 posts. In contrast, with a higher number of topics ($K = 200$), a much larger amount of documents need to be processed by online inference to reach the perplexity of the GS-learned model.

We note that due to the structure of MfTM, it is not possible to use perplexity to compare it against other topic models. On the one hand, our model utilizes auxiliary semantics to influence the topic assignments of terms in a post. On the other hand, named entities and the posts' timestamps need to be considered for topic assignment. These issues make perplexity an inappropriate metric for comparing with other models, which only consider a subset of the data (e.g., only general words). Thus, we evaluate MfTM against other models on a practical task of tweet clustering in Section 4.4.

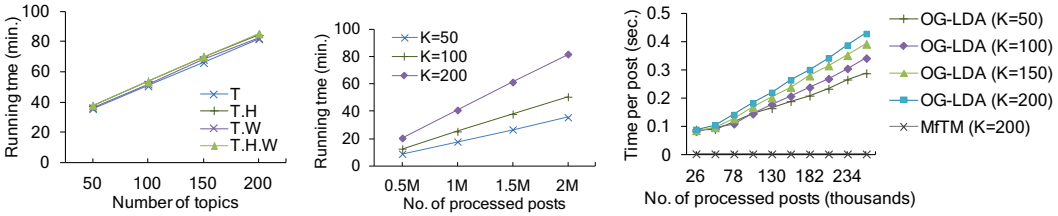
4.3.2. Scalability. To illustrate the runtime requirements of the online inference algorithm for MfTM, we conduct a scalability evaluation. We run the experiments using a standard PC with a dual-core CPU, 4GB RAM and a 600GB hard-drive. For comparison, we run scalability tests of an online Gibbs sampler for LDA proposed by Lau et al. [2012]. We refer to this baseline algorithm as OG-LDA.

First, we measure the time to train MfTM using different values of K and a fixed dataset size of 2 million tweets. The results in Figure 7(a) indicate a near-linear in-



(a) $K = 50$ (b) $K = 200$

Fig. 6: Perplexity evaluation of online inference of MfTM



(a) Impact of topic amount on training time (b) Impact of dataset size on training time (c) Training time per document

Fig. 7: Scalability evaluation of online inference

crease of training time as K increases. Second, we measure time to process a specified number of posts. Figure 7(b) illustrates that the inference algorithm is suitable for processing streaming data, since it essentially requires constant time to process each post. With $K = 200$, the algorithm required 0.004 sec. to process each post. To relate this figure to real-world data rates, let us consider the current rate of the Twitter public stream API[9]. Our inference algorithm could process its current daily rate of approx. 1.5 million English-language tweets in 100 mins. on a single machine. Importantly, we note that variational inference is well-suited for parallelization, thus the inference of MfTM can be distributed to further improve its performance.

In contrast, the baseline OG-LDA requires an increasing amount of time to process each posts as more posts are processed, starting with 0.08 sec./post and reaching 0.28 sec./post after processing 260,000 posts. This yields OG-LDA unsuitable for practical usage. We further compare scalability of our inference method with an online Gibbs sampler proposed by AlSumait et al. [2008], referred to as OG-LDA2. We use the timing results presented in the respective paper. In the presented results, OG-LDA2 required around 12 mins. to process a batch of 90-250 documents ($K = 50$). This would imply processing time between 2.88 to 8 sec./doc. While their reported running time remains nearly constant as more documents are processed, it is significantly longer than our inference method for MfTM (i.e., 0.004 sec./post). The timing results clearly show that SVI inference enjoys good scalability in face of voluminous data.

4.4. Clustering Evaluation

In order to evaluate the effectiveness of our framework in a practical scenario, we choose the task of tweet clustering. Clustering short texts, such as tweets, is an important and challenging problem due to their short length and lack of context [Jin et al. 2011; Rosa et al. 2010]. The performance of traditional text mining techniques is negatively affected in this situation, since the bag-of-words representation of a tweet

[9]https://dev.twitter.com/docs/streaming-apis

results in sparse instances. In contrast, our framework draws additional semantics from hashtags and URLs in tweets and distinguishes various entities in tweets.

We use Normalized Mutual Information (NMI) as the evaluation metric. NMI is a standard metric for evaluating clustering quality of labeled data and is given by

$$NMI(\Omega, C) = \frac{\sum_k \sum_j P(\omega_k \cap c_j) \log \frac{P(\omega_k \cap c_j)}{P(\omega_k)P(c_j)}}{[H(\Omega) + H(C)]/2}, \tag{10}$$

where Ω is the set of clusters, C is the set of classes, $P(\omega_k)$ is the probability of a post to be in cluster k and $P(c_j)$ is the probability of a post to be of class j. H is entropy, defined as $H(\Omega) = -\sum_k P(\omega_k) \log P(\omega_k)$. We select NMI as an overall evaluation metric due to its ability to balance the quality of the clustering against the number of obtained clusters. We perform clustering for each query in the ML dataset and each batch in the HL dataset and report the average NMI.

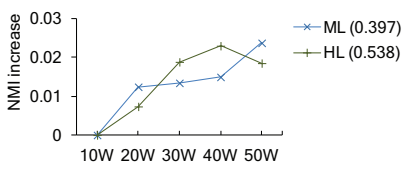
4.4.1. Baselines. We choose the following document representations as baselines.

— *TFIDF.* Traditional vector-space model with TFIDF term weighting.
— *LDA.* Standard topic model proposed by Blei et al. [2003].
— *Twitter-LDA (T-LDA).* Topic model proposed for Twitter data by Zhao et al. [2011].
— *Topics-over-Time (TOT).* Topic model proposed by Wang and McCallum [2006], which models the temporal distribution of each topic as a continuous Beta distribution.
— *Dual LDA (DLDA).* Topic model that jointly models short documents (e.g., Twitter posts) and long auxiliary documents (e.g., web documents) [Jin et al. 2011]. DLDA produces two sets of topics, consisting of K_{aux} auxiliary topics and K_{tar} target topics. We follow the parameterization in Jin et al. [2011] and set $K_{aux} = K_{tar} = K/2$.

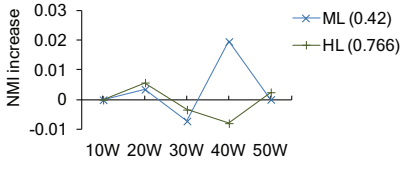
For the clustering task, we build MfTM and all baseline topic models on the background Twitter dataset with a varying number of topics (50, 100, 150, 200). In addition, MfTM is built on each version of our dataset with different extent of semantic enrichment applied (cf. Section 4.2). In this way, we may evaluate the effectiveness of semantic enrichment on MfTM.

4.4.2. Clustering Algorithms. We conduct clustering using the following algorithms.

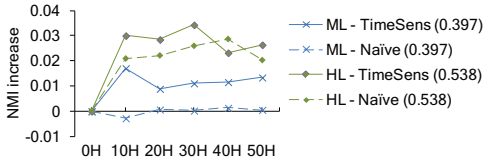
— *K-means.* Traditional algorithm for text clustering. When training K-means on the ML dataset, we set K equal to the number of unique topic labels for the respective query. Similarly for the HL dataset, K is set to the number of unique hashtags in a batch. As distance metrics, we use cosine distance for TFIDF and symmetrized KL-divergence for topic models. Due to the random initialization of K-means, we repeat each clustering 10 times and report the average result.
— *DBSCAN.* A widely adopted density-based clustering algorithm [Ester et al. 1996]. The features of DBSCAN include finding clusters of arbitrary shapes and automatically determining the number of clusters. We use the same distance metrics as with K-means. Since we do not want DBSCAN to remove outliers, we set $minPts = 1$. We tune ϵ separately for TFIDF and the topic models and set as $\epsilon_{TFIDF} = 0.4$, $\epsilon_{topic} = 0.1$.
— *Single-pass Incremental Clusterer (SPIC).* Streaming clustering algorithm designed to handle large-scale and real-time data [Becker et al. 2010]. The algorithm processes each data instance once and assigns it to the nearest cluster. If the nearest cluster is further than a distance threshold δ, the instance will form a new cluster. We use the same distance metrics as with K-means. δ is tuned separately for TFIDF and the topic models on a series of experiments and finally set as $\delta_{TFIDF} = 0.3$, $\delta_{topic} = 0.7$.
— *Direct.* We utilize each topic model to perform "hard clustering" of posts. For each post in the ML and HL datasets, we obtain its topic vector and then assign the post to the most likely topic. Formally, $cluster(p) = \arg\max_k \theta_p$. In this way, we obtain C clusters where C is less or equal to the number of latent topics K.



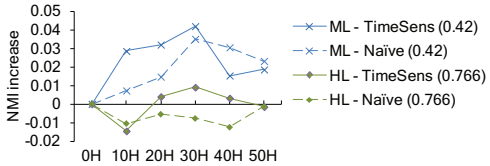
(a) K-means: Effect of web-SE



(b) DBSCAN: Effect of web-SE

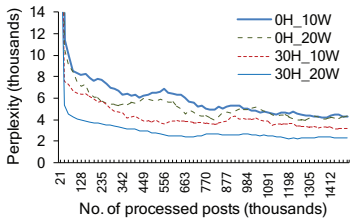


(c) K-means: Effect of hash-SE

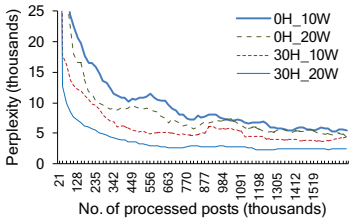


(d) DBSCAN: Effect of hash-SE

Fig. 8: Effect of semantic enrichment on clustering results. Baseline NMI corresponding to each left-most result is indicated in parentheses.



(a) $K = 50$



(b) $K = 100$

Fig. 9: Influence of Semantic Enrichment on Online inference of MfTM

4.4.3. Effect of Semantic Enrichment. Before presenting overall clustering results, we aim to study the effect of semantic enrichment (SE) on the performance of MfTM. We are also interested in choosing a suitable number of auxiliary and social terms to be included in a post during the SE process. The chosen number of SE terms is then used to build a final version of MfTM, to be presented in overall clustering results.

To achieve this goal, we perform clustering with K-means and DBSCAN using each dataset version (cf. Section 4.2.2) and $K = 50$. In this way, we may gain direct insight into the effects of web document-based semantic enrichment (*web-SE*) and hashtag-based semantic enrichment (*hash-SE*) on the clustering quality. Since K-means and DBSCAN produce different results for each evaluation dataset by absolute values, we instead focus on the relative increase of NMI. Figures 8 (a)-(d) show the influence of web-SE and hash-SE on the ML and HL evaluation datasets. The absolute NMI for each evaluation dataset is shown in parentheses next to its label and corresponds to the respective baseline result (i.e., the left-most value).

The results reveal several differences between the semantic enrichment methods and their impact on the clustering algorithms. When using web-SE, the effect on K-means clustering is overall positive (Fig. 8 (a)). However, web-SE has mixed effects on DBSCAN clustering when more than 20 auxiliary terms are included (Fig. 8 (b)). A possible reason is that DBSCAN clusters are formed based on density. Injecting many auxiliary terms may result in topic vectors that are closer together. Thus, two instances may be put in the same DBSCAN cluster, however they may be in different clusters if no web-SE has been performed. In fact, we observe the same phenomenon of DBSCAN

Table IV: Overall clustering results in NMI. Statistical significance of results of MfTM w.r.t. baselines (p-value by t-test) is shown at signific. levels 0.1 ($'$), 0.05 ($''$), 0.01 ($'''$).

	Dataset ML				**Dataset HL**			
Model	Direct	Kmeans	SPIC	Dbscan	Direct	Kmeans	SPIC	Dbscan
TFIDF[1]	-	0.312	0.487	0.398	-	0.393	0.689	0.548
LDA[2]	0.424	0.369	0.499	0.387	0.526	0.463	0.563	0.545
T-LDA[3]	0.406	0.358	0.463	0.362	0.432	0.307	0.482	0.525
TOT[4]	0.412	0.375	0.338	0.386	0.53	0.469	0.488	0.702
DLDA[5]	0.39	0.398	0.393	0.336	0.397	0.379	0.478	0.487
MfTM	0.442 $^{4'\ 5''}$	0.422 $^{1'''\ 2'''\ 3'''}_{4'''\ 5''}$	0.571 $^{1''\ 2'''\ 3'''}_{4'''\ 5'''}$	0.456 $^{1''\ 2'''\ 3'''}_{4''\ 5'''}$	0.618 $^{2''\ 3'''}_{4'''\ 5'''}$	0.586 $^{1'''\ 2'''\ 3'''}_{4'''\ 5'''}$	0.788 $^{1''\ 2'''\ 3'''}_{4'''\ 5'''}$	0.79 $^{1'''\ 2'''\ 3'''}_{4'''\ 5'''}$

when hash-SE is performed. Based on this observation, we choose 20 as the number of auxiliary terms for web-SE.

Regarding hash-SE, we compare the proposed time-sensitive hash-SE method (denoted "TimeSens") with a "naïve" hash-SE method. In the naïve approach, terms are scored only by Equation (1). Figures 8 (c)-(d) indicate that the best performance is achieved by the proposed time-sensitive method. In fact, the "naïve" method fails to improve the baseline performance of K-means on the ML dataset and similarly, when using DBSCAN on the HL dataset. The results thus confirm that the time-sensitive hash-SE method is both important and effective when performing hash-SE on microblog content, which spans longer time periods. We select the optimal number of terms for hash-SE to be 30, as it achieves the best results using both algorithms.

In summary, our experiments show that both hash-SE and web-SE positively influence clustering quality. In particular, we observe that hash-SE is able to achieve larger relative improvements of NMI compared with web-SE, indicating the benefits of our proposed hash-SE method. As a result of this evaluation, we choose the top-20 terms to be used for web-SE and top-30 terms for hash-SE. A final version of MfTM is built using these settings. Figures 9 (a) and (b) show the perplexity of MfTM with the chosen SE parameters and compare against the baseline settings of SE.

4.4.4. Clustering Results. We now present the overall clustering results in Table IV. For each evaluated model (cf. Section 4.4.1), we present the best clustering result across all values of K. Starting with the baseline TFIDF representation, we observe that TFIDF outperforms most topic model baselines using SPIC and DBSCAN. This behavior is in agreement with the findings by Rosa et al. [2010]. Since LDA is based on a (potentially sparse) bag-of-words representation of tweets, it fails to produce a significant accuracy improvement over TFIDF.

Among the baseline topic models, LDA produces the best results on the ML dataset, with the exception of K-means. We recall that all posts in the ML dataset originate around a specific query time, hence the temporal dimension is less important. In contrast, the TOT model achieves the best results on the HL dataset, with the exception of SPIC. This result suggests that modeling the temporal characteristics of topics benefits the clustering performance. The baseline DLDA model considers both microblog and auxiliary documents to obtain two sets of topics from each respective corpus. However, the results of DLDA are inconsistent across the datasets and clustering algorithms, showing that the integration of auxiliary semantics adopted by DLDA is not sufficient in the microblogging environment.

The representation obtained using MfTM achieves the best overall results compared with all baseline methods. This shows that the multi-faceted topics of MfTM have better potential to place semantically related tweets into the same clusters. Additionally,

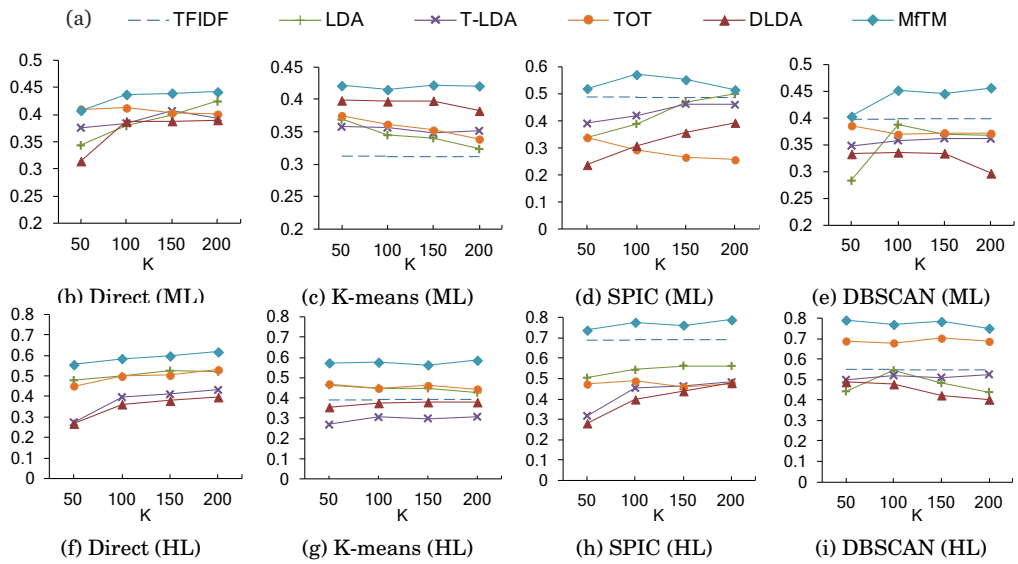


Fig. 10: Clustering results in NMI using different values of K. An overview of notations is shown in (a). For each figure, the dataset name is given in parentheses.

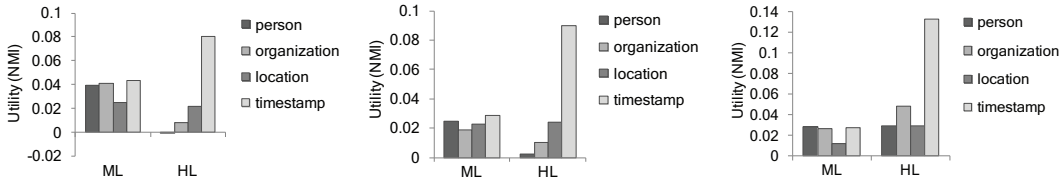
we perform a paired student's t-test to determine if the differences between the results of MfTM and each baseline method are statistically significant. As Table IV indicates, MfTM significantly outperforms all baselines using K-means, SPIC and DBSCAN.

To further examine the effectiveness of each data representation, we present the results achieved by different values of K in Figures 10 (a)-(h). Starting with TFIDF, we observe that it outperforms most baseline topic models using SPIC and DBSCAN clustering. This is further evidence that existing topic models are insufficient in addressing the unique characteristics of microblog content and thus fail to produce robust results. As an example, TOT performs well using Direct and DBSCAN clustering, however its performance is inconsistent when K-means or SPIC is used. Similarly, DLDA performs well on the ML dataset using K-means, however it fails when other algorithms are used. In contrast, the proposed MfTM consistently outperforms all baseline models across our datasets and clustering algorithms. This shows that the rich semantics captured by MfTM help to achieve robust clustering results.

The clustering experiments show that the proposed framework is effective in the tweet clustering task and outperforms various state-of-the-art baselines.

4.4.5. Utility of Named Entities and Timestamps in Clustering. In this section, our goal is to evaluate the benefits of utilizing additional semantics associated with the posts to be clustered. By supplying named entities associated with each post to MfTM, the named entity facets of each topic can be utilized (cf. Equation (8)). Similarly, we may utilize the post's timestamp. During this evaluation, we use MfTM trained with $K = 100$.

We start our analysis by supplying all semantics associated with a tweet to be clustered to MfTM (i.e., tweet's terms, 3 types of named entities and timestamp). This is the default setting used in clustering evaluations in the previous sections. We then conduct a series of experiments, each time omitting one type of semantics and measuring the impact on the clustering quality. We devise a simple metric *utility* to measure the increase in NMI when semantics of type s are supplied to MfTM, compared with NMI achieved when s is omitted. Formally, $Utility(s) = NMI_{all} - NMI_{all \setminus s}$. Since this metric shows the relative increase in NMI, it allows us to compare results using dif-



(a) Direct clustering (b) K-means clustering (c) DBSCAN clustering

Fig. 11: Utility of named entities and timestamps in clustering

ferent datasets and clustering algorithms. Figure 11 shows results using the ML and HL datasets and Direct, K-means and DBSCAN clustering.

From the results, we observe that the utility of additional semantics differs by dataset type. The HL dataset consists of tweets that are spread over a longer period of time. Thus, the tweet's timestamp is more important than in the ML dataset. In contrast, the ML dataset covers a short time period, thus other semantics such as named entities become useful. This shows that MfTM can support various characteristics of microblog datasets. Among the named entity types we analyzed, each contributes to the overall clustering quality using the ML dataset. In the HL dataset, "person" named entities have a lower utility, which can be attributed to a relatively low number of tweets containing this named entity type (cf. Table III).

5. CONCLUSION

In this paper, we study the problem of topic discovery in the microblogging environment of Twitter. To tackle the short length of microblog posts and uncover their rich latent semantics, we propose a novel multi-faceted topic modeling framework. Our framework takes into account the users' posts, auxiliary semantics from linked web documents and named entities. Moreover, we exploit a new source of "social chatter" associated with hashtags to aid topic modeling. After applying our pre-processing techniques, we integrate the various semantics within the Multi-faceted Topic Model. As shown in our evaluation, the latent topics discovered by MfTM are beneficial for downstream applications such as tweet clustering. Experiments with multiple clustering algorithms reveal that MfTM consistently outperforms baseline topic models. Our experiments also confirm that the proposed web-document-based and hashtag-based semantic enrichment methods provide important additional semantics for topic modeling. Moreover, MfTM exhibits good scalability in face of voluminous data.

Relevant issues for future work include extending MfTM using social connections, by considering the interests of other users that one interacts with. Regarding hashtag-based semantic enrichment, more detailed treatment of different hashtag types, trending behaviors or the hashtag's context may be considered.

REFERENCES

F. Abel, Q. Gao, G.-J. Houben, and K. Tao. 2011. Semantic enrichment of twitter posts for user profile construction on the social web. In *Proc. of ESWC '11*, 375–389.

L. AlSumait, D. Barbara, and C. Domeniconi. 2008. On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *Proc. of ICDM '11*, 3–12.

H. Becker, M. Naaman, and L. Gravano. 2010. Learning Similarity Metrics for Event Identification in Social Media. In *Proc. of WSDM '10*, 291–300.

D. M. Blei, A. Y. Ng, and M. I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*. 3 (March 2003), 993–1022.

I. Celik, F. Abel, and G.-j. Houben. 2011. Learning Semantic Relationships between Entities in Twitter. In *Proc. of ICWE'11*, 167–181.

Y. Duan, L. Jiang, T. Qin, M. Zhou, and H.-Y. Shum. 2010. An Empirical Study on Learning to Rank of Tweets. In *Proc. of COLING'10*, 295–303.

M. Efron. 2010. Hashtag retrieval in a microblogging environment. In *Proc. of SIGIR'10*, 787–788.

M. Efron and G. Golovchinsky. 2011. Estimation Methods for Ranking Recent Information. In *Proc. of SIGIR'11*, 495–504.

M. Ester, H. P. Kriegel, J. Sander, and X. Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of KDD'96*, 226–231.

Y. Genc, Y. Sakamoto, and J. V. Nickerson. 2011. Discovering context: Classifying tweets through a semantic transform based on wikipedia. In *Proc. of HCI'11*, 484–492.

G. Groh, F. Straub, J. Eicher, and D. Grob. 2013. Geographic aspects of tie strength and value of information in social networking. In *Proc. of LBSN Workshop*, 1–10.

M. Hoffman, C. Wang, and J. Paisley. 2013. Stochastic Variational Inference. *Journal of Machine Learning Research*.14, 1 (May 2013), 1303–1347.

L. Hong and B. D. Davison. 2010. Empirical study of topic modeling in twitter. In *Proc. of SOMA Workshop*, 80–88.

J. Huang, K. M. Thornton, and E. N. Efthimiadis. 2010. Conversational tagging in twitter. In *Proc. of HT'10*, 173–177.

O. Jin, N. Liu, K. Zhao, Y. Yu, and Q. Yang. Transferring topical knowledge from auxiliary long texts for short text clustering. 2011. In *Proc. of CIKM'11*, 775–784.

Y. Jo and A. H. Oh. 2011. Aspect and sentiment unification model for online review analysis. In *Proc. of WSDM'11*, 815–824.

J. H. Lau, N. Collier, and T. Baldwin. 2012. On-line trend analysis with topic models: #twitter trends detection topic model online. In *Proc. of COLING*, 1519–1534.

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

D. Newman, C. Chemudugunta, and P. Smyth. 2006. Statistical entity-topic models. In *Proc. of KDD'06*, 680–686.

M. Paul and R. Girju. 2010. A two-dimensional topic-aspect model for discovering multi-faceted topics. In *Proc. of AAAI'10*, 545–550.

A. Ritter, S. Clark, Mausam, and O. Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proc. of EMNLP'11*, 1524–1534.

K. D. Rosa, R. Shah, B. Lin, A. Gershman, and R. Frederking. 2010. Topical Clustering of Tweets. In *Proc. of SWSM'10*.

M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. 2004. The author-topic model for authors and documents. In *Proc. of AUAI'04*, 487–494.

O. Tsur and A. Rappoport. 2013. Efficient Clustering of Short Messages into General Domains. In *Proc. of ICWSM'13*, 621–630.

J. Vosecky, D. Jiang, K. W. T. Leung, and W. Ng. 2013. Dynamic multi-faceted topic discovery in twitter. In *Proc. of CIKM'13*, 879–884.

J. Vosecky. 2014. Online Appendix to: Integrating Social and Auxiliary Semantics for Multi-Faceted Topic Modeling in Twitter. (April 2014). Retrieved April 12, 2014 from http://www.cse.ust.hk/~wilfred/mftm.html

B. Walsh. 2004. Markov chain monte carlo and gibbs sampling. Lecture Notes, MIT.

X. Wang and A. McCallum. 2006. Topics over Time: A Non-Markov Continuous-Time Model of Topical Trends. In *Proc. of KDD'06*, 424–433.

X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang. 2011. Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In *Proc. of CIKM'11*, 1031–1040.

J. Weng, E.-P. Lim, Q. He, and C. W.-K. Leung. 2010. What do people want in microblogs? measuring interestingness of hashtags in twitter. In *Proc. of ICDM'10*, 1121–1126.

W. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. 2011. Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval*, volume 6611, 338–349. Springer-Verlag.

Received November 2013; revised April 2014; accepted July 2014