8-2015

Event identification and analysis on Twitter

Qiming DIAO
Singapore Management University, qiming.diao.2010@smu.edu.sg

Event Identification and Analysis on Twitter

by

Qiming DIAO

Submitted to School of Information Systems in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Information Systems

Dissertation Committee:

Jing JIANG (Supervisor / Chair)
Assistant Professor of Information Systems
Singapore Management University

Hady W. LAUW
Assistant Professor of Information Systems
Singapore Management University

Ee-Peng LIM
Professor of Information Systems
Singapore Management University

Wee Sun LEE
Associate Professor of Computer Science
National University of Singapore

Singapore Management University
2015

Event Identification and Analysis on Twitter

Qiming DIAO

Abstract

With the rapid growth of social media, Twitter has become one of the most widely adopted platforms for people to post short and instant messages. Because of such wide adoption of Twitter, events like breaking news and release of popular videos can easily capture people's attention and spread rapidly on Twitter. Therefore, the popularity and importance of an event can be approximately gauged by the volume of tweets covering the event. Moreover, the relevant tweets also reflect the public's opinions and reactions to events. It is therefore very important to identify and analyze the events on Twitter. In this dissertation, we introduce our work which aims to (1) identify events from Twitter stream, (2) analyze personal topics, events and users on Twitter, and (3) summarize the events identified from Twitter.

First of all, we focus on event identification on Twitter. We observe that the textual content coupled with the temporal patterns of tweets provides important insight into the general public's attention and interests. A sudden increase of topically similar tweets usually indicates a burst of attention in some events that has happened offline (such as a product launch or a natural disaster) or online (such as the spread of a viral video). Based on these observations, we propose two models to identify events on Twitter, which are extended from LDA and a non-parametric model. These two models share two common assumptions: (1) similar tweets emerged around the same time are more likely about some events, and (2) similar tweets published by the same user over a long term are more likely about the user's personal background and interests. These two assumptions help separate event-driven tweets from the large proportion of personal-interests-driven tweets. The first model needs to predefine the number of events because of the limitation of topic models. However, events emerge and die out fast along the time line, and the number can be

countable infinite. Our non-parametric model overcomes this challenge.

In the first task described above, we aim to identify events underlying the Twitter stream, and we do not consider the relation between events and users' personal interest topics. However, the concept of events and users' personal interest topics are orthogonal in that many events fall under certain topics. For example, concerts fall under the topic about music. Furthermore, being *social* media, Twitter users play important roles in forming topics and events on Twitter. Each user has her own topic interests, which influence the content of her tweets. Whether a user publishes a tweet related to an event also largely depends on whether her topic interests match the nature of the event. Modeling the interplay between topics, events and users can deepen our understanding of Twitter content and potentially aid many predication and recommendation tasks. For the second task, we aim to construct a unified model of topics, events and users on Twitter. The unified model is a combination of a topic model, a dynamic non-parametric model and matrix factorization. The topic model part is to learn users' personal interest topics. The dynamic non-parametric model is to identify events from the tweets stream, and finally matrix factorization is to model the interaction between topics and events.

Finally, we aim to summarize the events identified on Twitter. In the previous two tasks, we utilize topic models and a dynamic non-parametric models to identify events from tweets stream. For both methods, events are learnt as clusters of tweets featured by multinomial word distributions. Therefore, users need to either read the clusters of tweets or the word distribution to interpret the events. However, the former is time-consuming and the latter cannot accurately represent the events. In this case, we propose a novel graph-based summarization method that generates concise abstractive summaries for the events.

Overall, this dissertation presents our work on event identification first. Then we further analyze events, users and personal interest topics on Twitter, which can help better understand users' tweeting behavior on events. Finally, we propose a summarization method to generate abstractive summaries for the events on Twitter.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

With the rapid growth of social media on the Web and the fast adoption of smart mobile devices, the way people consume information has been fundamentally changed. For the younger generation, traditional media such as newspapers, TV and radio have been replaced by new media such as Twitter. Moreover, social media allow users to actively participate in generating content. In particular, Twitter as a microblog site allows people to publish short, instant textual posts anywhere and anytime, making content generation ever easier. Till July 2014, Twitter contains more than 600 millions active users, and these users publish around 58 millions of tweets per day. A consequence of the wide adoption of Twitter is that major events can easily catch eyes of the majority and cause heated discussion. Thus, the popularity and importance of an event can be approximately gauged by the volume of tweets covering the event. These events can be either news related (e.g. traffic accident, election) or totally online (e.g. the spread of a viral video). Moreover, the relevant tweets also reflect the public's opinions and reactions to events such as elections and scandals. It is therefore very useful to find popular events and their relevant tweets from Twitter.

Identifying events and their relevant tweets is non-trivial. As Twitter is arguably

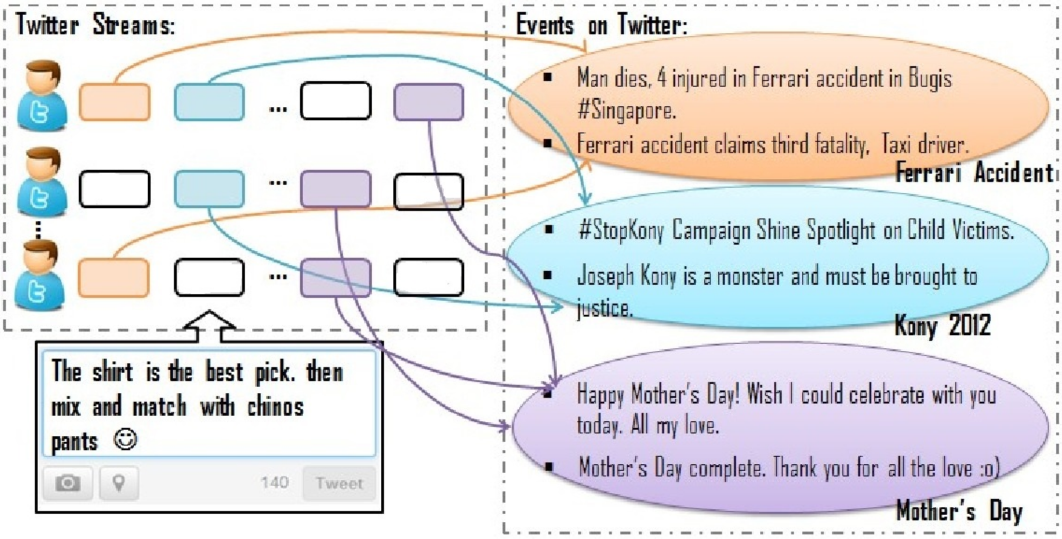

Twitter Streams:
The shirt is the best pick. then mix and match with chinos pants ☺
140
Tweet
Events on Twitter:
Man dies, 4 injured in Ferrari accident in Bugis #Singapore.
Ferrari accident claims third fatality, Taxi driver.
Ferrari Accident
#StopKony Campaign Shine Spotlight on Child Victims.
Joseph Kony is a monster and must be brought to justice.
Kony 2012
Happy Mother's Day! Wish I could celebrate with you today. All my love.
Mother's Day complete. Thank you for all the love :o)
Mother's Day

Figure 1.1: Example events on Twitter and some representative tweets. Note that tweets can be both event related (colored) and personal life related (in white).

the most popular microblog site for user to post and share, most tweets are about daily routines and personal interests, while only a small proportion of tweets underlying the Twitter torrent is event related. These personal interests are longstanding and user specific. Figure 1.1 illustrates that user can tweet about both daily routines and events. According to a Twitter study by PearAnalytics[1], only 3.6% of tweet are news-related and 8.7% have pass-along value. This makes Twitter different from the traditional news stream and news-oriented forum, which used to be the best source of information to detect and summarize events. For these traditional news dataset, event detection is well studied under Topic Detection and Tracking (TDT) in the information retrieval community (e.g. [54, 7, 55]). These work focuses on evolutionary clustering of streaming news articles. Nevertheless, identifying events on Twitter stream is more challenging, since we need to unearth event related tweets from huge tweet flow.

The problem we study is similar to but different from *event detection* on Twitter that has been a hot research topic in recent years. Existing work on event detection from Twitter usually focuses on *early*, *online* detection of major events [44, 38, 50, 9, 32]. For example, Sakaki et al. studied realtime detection of earthquake events

[1] https://www.pearanalytics.com/wp-content/uploads/2012/12/Twitter-Study-August-2009.pdf

for Japan [44]. Petrović et al. studied how to detect the first tweet covering a new event [38]. These studies stress the importance of detecting the onset of an event at the moment or shortly after the event happens, which is critical for monitoring social media for unexpected events such as natural disasters, terrorist attacks and outbreaks of contagious diseases. In contrast, our focus is to identify *all* tweets related to an event.

Furthermore, being *social* media, Twitter users play important roles when publishing event related tweets. Each user has her own personal interests which can be inferred from her past interest-related tweets. These user-specific interests will influence the type of event she concerns. In this case, the concepts of personal interests and events are orthogonal in that many events fall under certain topics. Analysis on the interplay of personal interests and events can deepen our understanding of Twitter content and potentially aid many recommendation and prediction tasks.

Finally, when a significant event happens, it will cause heated discussion. However, it is time consuming for a user to read all the related tweets to interpret the event. Therefore, we aim to summarize the identified events to help users better interpret these events.

1.2 Research Objectives

In this dissertation, we therefore aim to (1) identify events from Twitter stream, (2) analyse personal topics, events and users on Twitter, (3) summarize the identified events on Twitter. Formally, we define an event to be something non-trivial that happens at a certain time. An event can be either planned or unexpected. Example of events include plane crashes, concerts, elections etc. We will refer to the first task as *Event Identification*, the second task as *Unified Analysis for Topics, Events and Users on Twitter* where topics stand for users' personal interests, and the third task as *Event Summarization*.

1.2.1 Event Identification

In this part, the key challenges are the following: (1) Existing methods which aim to detect events from news stream assume that all documents are event-related. On Twitter, however, many tweets are not related to a significant event. The majority of tweets are about people's interests and daily routines; (2) Events on Twitter are always bursty. Due to the nature of Twitter, people usually use Twitter to spread or comment on breaking news rather than old events.

To deal with these challenges, we propose two statistical model based approaches to model the generation of Twitter stream and identify events from Twitter:

- We propose a topic model designed for finding topical bursts from microblogs. Our model is based on the following two assumptions: (1) If a post is about a global event, it is likely to follow a global topic distribution that is time dependent; (2) If a post is about a personal topic, it is likely to follow a personal topic distribution that is more or less stable over time. Separation of "global" and "personal" posts is done in an unsupervised manner through hidden variables. Finally, we apply a state machine to detect topical bursts within each topic, and each burst stands for an event.

- We propose a new non-parametric generative model to identify events from Twitter. The basic assumptions is similar. We assume that posts about personal interests are likely to follow a user-specific topic distribution. However, events on Twitter tend to emerge and die out fast. Thus, the number of events can be countably infinite over the timeline. In this case, we propose a dynamic non-parametric model to capture the events on Twitter.

1.2.2 Unified Analysis for Topics, Events and Users on Twitter

In this part, we propose a unified model for topics, events and users on Twitter. Petrovic et al. [39] recently point out that Twitter stream does not lead news stream for major news events, but Twitter stream covers a much wider range of events than news stream. Our model aims to better understand these additional events and their relation with users' personal interests. Therefore, we point out two important concepts (1) *Topics*. These are longstanding themes that many personal tweets revolve around. Example topics range from music and sports to more serious ones like politics and religion. (2) *Events*. As what we have described, these are things that take place at a certain time and attract many people's short attention in social media. We use topic model and dynamic non-parametric model to capture topics and events retrospectively. Moreover, a user's tweeting behaviour on an event is similar to adoption of an item during purchasing. Thus we use event-topic affinity vectors inspired by PMF-based collaborative filtering. It uses the latent topics to explain users' preferences on events and subsequently infers the association between topics and events.

1.2.3 Event Summarization

In previous parts introduced above, we use topic model and dynamic non-parametric model to capture the events on Twitter. Both approaches model an event as a cluster of tweets featured by a coherent multinomial word distribution. Therefore, to understand an event, a user need to either read the cluster of tweets or the word distribution. However, the former is very time consuming, and the latter cannot accurately represent the event. In this case, we propose a word graph based novel summarization framework. Within the graph, the nodes represent words with directed edges representing relative positional information between the words within each tweet. Thus, we cast the summarization task as finding paths with high scores in the graph, which is an instance of the maximum arborescence problem for directed graphs.

1.3 Our Contributions

Our contributions in this dissertation are as follows:

1.3.1 Event Identification on Twitter

- We study the problem of finding topical bursts from Twitter streams. Because existing work on burst detection from text streams may not be suitable for microblogs, we propose a new topic model that considers both the temporal information of tweets and users' personal interests. We then apply a Poisson-based state machine to identify bursty periods from the topics discovered by our model. We compare our model with standard LDA as well as two degenerate variations of our model on a real Twitter dataset. Our quantitative evaluation shows that our model can more accurately detect unique topical bursts where each of them represents an event. A limitation of this work is that the number of topics is pre-determined, while the number of events can be infinite over the timeline. Therefore, our next piece of work looks into dynamic non-parametric model which can better capture the nature of events.

- We study the problem of event identification from Twitter stream. The Recurrent Chinese Restaurant Process is appealing for our task because it provides a principled dynamic non-parametric model. However, our preliminary experiment shows that RCRP is not directly applicable in our task for two reasons: (1) events emerge and die out fast on Twitter, (2) most tweets are topical and only a small proportion of them are event-related. Therefore, we propose a novel duration-based probability discount to RCRP to capture the burstiness character of events on Twitter. We then propose a probabilistic model to identify both events and topics simultaneously from Twitter. Our experiments demonstrate that our proposed model can identify events accurately, which shows the effectiveness of duration-based discount. Finally, we qualitatively show some interesting studies on users and event-topic correlations.

1.3.2 Unified Analysis for Topics, Events and Users on Twitter

- We propose a unified model to study topics, events and users jointly. The base of our method is a combination of a LDA-like model and the Recurrent Chinese Restaurant Process, which aims to model users' longstanding personal topic interests and events over time simultaneously. We use an inner popularity bias parameter and event-topic affinity vectors to interpret an event's inherent popularity and its affinity to different topics. Our experiments quantitatively show that our proposed model can effectively identify meaningful events and accurately find relevant tweets for these events. Furthermore, the event-topic association inferred by our model can help an event recommendation task and organize events by topics.

1.3.3 Event Summarization

- We propose a novel graph-based summarization framework that generates concise abstractive summaries for events. The key idea is to use a directed word graph structure to represent natural language tweet text and cast this summarization problem as finding appropriate paths in the graph. Then we can cast the summarization task as finding the highest scoring spanning tree, which is an instance of the maximum arborescence problem for directed graphs. Our experiments show that our framework has better agreement with human summaries compared with baseline methods.

1.4 Road Map

The remaining part of the dissertation is structured as follows. We first review related work in Chapter 2. We then present our work on event identification in Part I.

This part includes two chapters: Chapter 3 presents our work about finding topical bursts where each burst represents an event. We use a topic model based approach to identify topics from Twitter, followed by a two state machine to detect bursts within each topic. In this work, the number of topics need to be predefined, and post-processing is needed to detect the bursts. To better model the generation process of events on Twitter, we present our dynamic non-parametric model based approach in Chapter 4, in which the model can directly identify events and better capture their emergence. Next, Part II describes our work about unified analysis for topics, events and users on Twitter. This part includes Chapter 5, which further explores the underlying motivation of users' tweeting behavior on events by studying the relation between events and personal interests. Finally, in Part III, we introduce our work about event summarization. This part includes Chapter 6, which introduces our word graph based summarization framework in detail.

Chapter 2

Literature Review

For the past few decades, people can easily access a vast amount of textual information provided by various sources like newspaper, online media, and microblogs. When a significant event happens, it tends to be “bursty” because the event can easily catch the majority’s eyes and cause heated discussion. Therefore, the first concern of event identification is such “burstiness” property. To detect bursty patterns from data streams, Kleinberg et al. [29] proposed a state machine to model the arrival times of documents in a stream. Different states generate time gaps according to exponential density functions with different expected values, and bursty intervals can be discovered from the underlying state sequence. A similar approach by Ihler et al. [28] models a sequence of count data using Poisson distributions. However, these methods can only be applied to detect bursty patterns, given a single stream about a certain topic (e.g. transportation, sports, politics, etc).

News stream provide various daily event-oriented information, which covers almost every area of public life. Topic Detection and Tracking (TDT) is a relatively old research area, which mainly focuses on event identification on news stream. However, each article in a news stream is event-related, which makes it quite different from microblogs.

Due to the fast growth of Web 2.0, social media, especially Twitter, fundamentally changed the way people seek information. Nowadays, when hearing or seeing an event, the first reaction of the majority is to post it on Twitter or other social media sites. Such property makes social media a good source for event identification. In addition, relevant tweets about an event can reflect the publics' opinions and reactions to events such as elections and scandals. It is therefore very useful to find popular events and their relevant tweets from Twitter. There have been quite a few work about event identification on twitter in recent years. These studies focus on early event detection, while we study event identification of events from a given segment of Twitter stream in a retrospective manner.

Overall, the inspiration of this dissertation comes from various tasks which include Topic Detection and Tracking (TDT) and event detection on Twitter. Moreover, the approaches we use are based on Beyesian statistical models. Finally, we explore abstractive summarization for the events on Twitter. Therefore, we provide literature review of the following areas:

- Topic Detection and Tracking
- Event Detection on Twitter
- Topic Modeling
- Summarization

2.1 Topic Detection and Tracking

Topic Detection and Tracking (TDT) is a relatively old research problem in the information retrieval community. A topic is defined as "a seminal event or activity, along with all directly related events and activities [6]." More specifically, topic detection involves detecting the occurrence of a new event such as a plane crash, a murder, a jury trial result, or a political scandal in a stream of news stories from multiple sources. Topic tracking is the process of monitoring a stream of news stories to find those that track (or discuss) the same event as one specified by a user. Much work has been done along this direction [7, 55, 54, 29], and these studies focus mostly on news articles, which used to be the best source of information to detect and summarize events. These studies are mainly based on two approaches: *document-pivot* and *feature-pivot*. The former aims to represent documents as vectors and calculate similarities between documents, and then cluster documents into events [7, 55, 54]. The latter aims to identify the features of the hidden events from the stream first, and then detect events by clustering these features [29]. Nevertheless, identifying events on Twitter stream is more challenging and our approach is quite different, due to several reasons: (1) Only a small proportion of tweets is event-related in Twitter streams, while most news articles are event-oriented. (2) Twitter content is user generated, where each user has her specific characteristics. We use a probabilistic approach which detects events and considers users' personal interests at the same time.

2.2 Event Detection on Twitter

There have been quite a few studies on event detection on Twitter [44, 38, 50, 9, 32, 51]. Sakaki et al. trained a classifier to recognize tweets reporting earthquakes in Japan [44]. Weng and Lee proposed a method that first characterizes temporal patterns of individual words using wavelets and then groups them into events [50]. Petrović et al. proposed the first story detection task on Twitter [38]. Becker et al. explored supervised approaches to distinguishing between messages about real-world events and non-event messages for Twitter stream analysis [9]. Xie et al. proposed a sketch-based topic model together with a set of techniques to achieve real-time detection of bursty topics on Twitter [51]. As these studies focus on early event detection, their major concerns are storage of past posts and efficient ways of computing similarities between posts. However, recently, Petrović et al. pointed out that Twitter does not necessarily lead traditional news media on major events, which suggests that early event detection on Twitter may not be as critically important as thought to be. Moreover, these work does not aim to identify *all* relevant tweets, nor do they analyze the association of events with personal interests. In comparison, our work focuses on modeling topics, events and users as well as their relations.

2.3 Topic Modeling

Topic models provide a principled and elegant way to discover hidden topics from large document collections [13, 23]. Latent Dirichlet Allocation (LDA) is a widely adopted topic model [13] and many extensions of LDA have been proposed. For the rest of this section, we mainly review work in topic modeling that is relevant to this dissertation.

Temporal Topic Modeling and the Recurrent Chinese Restaurant Process

Standard topic models do not consider temporal information. There have been many extensions to topic model to capture the temporal aspects of topics [12, 48, 26]. Blei and Lafferty considered the evolution of topics based on discretization of time [12]. Wang and McCallum model continuous time using a Beta distribution [48]. The models proposed in [49] and in [15] assume a topic distribution within each time epoch. However, these models need to pre-define the number of topics. Intuitively, the number of events should reach countable infinite over time in text streams. The Recurrent Chinese Restaurant Process overcomes this limitation by allowing topics to emerge and disappear along the timeline [5]. Ahmed et al. proposed a unified model which combines the Recurrent Chinese Restaurant Process with LDA to detect events in news streams [3]. Tang et al. further extended the model by capturing user interests in some news-centric social media streams [46]. These studies are closely related to our approach.

Collaborative Filtering with Topic Modeling

Users' tweeting behavior on events is similar to how user adopt item during purchasing. Therefore, our study is also related to the work on collaborative filtering based on probabilistic matrix factorization (PMF) [45]. Recently there has been some work combining topic model with PMF to recommend items with textual content such as news articles and advertisements [47, 1]. They use topics to interpret the latent structure of users and items. We borrow their idea but our items are events, which are not known and have to be discovered by our model.

Topic Modeling on Twitter

Tweets generated from all over the world are expected to be about a variety of topics. Figuring out which tweet is about which topic is interesting, because it enables personalization, discovery, and targeted recommendation. However, standard LDA might not work well on Twitter, since tweets are short and words used in Twitter are informal. There has been some work in this area [53, 56, 25]. Hong and Davison find that, by training a topic model on aggregated messages, they can obtain a higher quality of learned model which results in significantly better performance in two real-world classification problems [25]. Zhao et al. also suggest to aggregate messages in user level [56]. Yang et al. propose a deployed topic modeling system that infers topics of short noisy texts at high precision in real-time [53]. Our task is related to this branch of work, while we focus on event identification rather than purely modeling Twitter content.

2.4 Summarization

Abstractive summarization is also related to our task. Existing work in abstractive summarization can be categorized into two categories: (1) approaches using prior knowledge [41, 18], and (2) approaches using Natural Language Generation (NLG) systems [43, 27]. The first line of work needs considerable amount of manual effort to define schemas such as frames and templates that can be filled with information extraction technology. The second category of work uses deeper NLP analysis. However, neither branch of work can be directly applied in our task for two reasons: (1) Events keep happening everyday on Twitter, and the number of events can be countablely infinite. Therefore, we cannot rely on methods with manual efforts. (2) The language used in tweets is not formal. Thus deep NLP analysis may not be applicable on Twitter.

Part I

Event Identification

Chapter 3

Finding Bursty Topics from Microblogs

Microblogs such as Twitter reflect the general public's reactions to major events. Bursty topics from microblogs reveal what events have attracted the most online attention. Although bursty event detection from text streams has been studied before, previous work may not be suitable for microblogs because compared with other text streams such as news articles and scientific publications, microblog posts are particularly diverse and noisy. To find topics that have bursty patterns on microblogs, we propose a topic model that simultaneously captures two observations: (1) posts published around the same time are more likely to have the same topic, and (2) posts published by the same user are more likely to have the same topic. The former helps find event-driven posts while the latter helps identify and filter out "personal" posts. Our experiments on a large Twitter dataset show that there are more meaningful and unique bursty topics in the top-ranked results returned by our model than an LDA baseline and two degenerate variations of our model. We also show some case studies that demonstrate the importance of considering both the temporal information and users' personal interests for bursty topic detection from microblogs.

3.1 Introduction

With the fast growth of Web 2.0, a vast amount of user-generated content has accumulated on the social Web. In particular, microblogging sites such as Twitter allow users to easily publish short instant posts about any topic to be shared with the general public. The textual content coupled with the temporal patterns of these microblog posts provides important insight into the general public's interest. A sudden increase of topically similar posts usually indicates a burst of interest in some event that has happened offline (such as a product launch or a natural disaster) or online (such as the spread of a viral video). Finding bursty topics from microblogs therefore can help us identify the most popular events that have drawn the public's attention. In this chapter, we study the problem of finding bursty topics from a stream of microblog posts generated by different users. We focus on retrospective detection, where the text stream within a certain period is analyzed in its entirety.

Retrospective bursty event detection from text streams is not new [29, 19, 49], but finding bursty topics from microblog steams has not been well studied. In his seminal work, Jon Kleinberg proposed a state machine to model the arrival times of documents in a stream in order to identify bursts [29]. This model has been widely used. However, this model assumes that documents in the stream are all about a given topic. In contrast, discovering interesting topics that have drawn bursts of interest from a stream of topically diverse microblog posts is itself a challenge. To discover topics, we can certainly apply standard topic models such as LDA [13], but with standard LDA temporal information is lost during topic discovery. For microblogs, where posts are short and often event-driven, temporal information can sometimes be critical in determining the topic of a post. For example, typically a post containing the word "jobs" is likely to be about employment, but right after October 5, 2011, a post containing "jobs" is more likely to be related to Steve Jobs' death. Essentially, we expect that on microblogs, posts published around the same time have a higher probability to belong to the same topic.

To capture this intuition, one solution is to assume that posts published within the same short time window follow the same topic distribution. Wang et al. proposed a PLSA-based topic model that exploits this idea to find correlated bursty patterns across multiple text streams [49]. However, their model is not immediately applicable for our problem. First, their model assumes multiple text streams where word distributions for the same topic are different on different streams. More importantly, their model was applied to news articles and scientific publications, where most documents follow the global topical trends. On microblogs, besides talking about global popular events, users also often talk about their daily lives and personal interests. In order to detect global bursty events from microblog posts, it is important to filter out these "personal" posts.

In this chapter, we propose a topic model designed for finding bursty topics from microblogs. Our model is based on the following two assumptions: (1) If a post is about a global event, it is likely to follow a global topic distribution that is time-dependent. (2) If a post is about a personal topic, it is likely to follow a personal topic distribution that is more or less stable over time. Separation of "global" and "personal" posts is done in an unsupervised manner through hidden variables. Finally, we apply a state machine to detect bursts from the discovered topics.

We evaluate our model on a large Twitter dataset. We find that compared with bursty topics discovered by standard LDA and by two degenerate variations of our model, bursty topics discovered by our model are more accurate and less redundant within the top-ranked results. We also use some example bursty topics to explain the advantages of our model.

3.2 Method

3.2.1 Preliminaries

We first introduce the notation used in this chapter and formally formulate our problem. We assume that we have a stream of D microblog posts, denoted as $d_1, d_2, \ldots, d_D$. Each post d_i is generated by a user u_i, where u_i is an index between 1 and U, and U is the total number of users. Each d_i is also associated with a discrete timestamp t_i, where t_i is an index between 1 and T, and T is the total number of time points we consider. Each d_i contains a bag of words, denoted as $\{w_{i,1}, w_{i,2}, \ldots, w_{i,N_i}\}$, where $w_{i,j}$ is an index between 1 and V, and V is the vocabulary size. N_i is the number of words in d_i.

We define a bursty topic b as a word distribution coupled with a bursty interval, denoted as (ϕ^b, t_s^b, t_e^b), where ϕ^b is a multinomial distribution over the vocabulary, and t_s^b and t_e^b ($1 \leq t_s^b \leq t_e^b \leq T$) are the start and the end timestamps of the bursty interval, respectively. Our task is to find meaningful bursty topics from the input text stream.

Our method consists of a topic discovery step and a burst detection step. At the topic discovery step, we propose a topic model that considers both users' topical interests and the global topic trends. Burst detection is done through a standard state machine method.

3.2.2 Our Topic Model

We assume that there are C (latent) topics in the text stream, where each topic c has a word distribution ϕ^c. Note that not every topic has a bursty interval. On the other hand, a topic may have multiple bursty intervals and hence leads to multiple bursty topics. We also assume a background word distribution ϕ^B that captures common words. All posts are assumed to be generated from some mixture of these $C + 1$ underlying topics.

In standard LDA, a document contains a mixture of topics, represented by a topic distribution, and each word has a hidden topic label. While this is a reasonable assumption for long documents, for short microblog posts, a single post is most likely to be about a single topic. We therefore associate a single hidden variable with each post to indicate its topic. Similar idea of assigning a single topic to a short sequence of words has been used before [22, 56]. As we will see very soon, this treatment also allows us to model topic distributions at time window level and user level.

As we have discussed in Section 3.1, an important observation we have is that when everything else is equal, a pair of posts published around the same time is more likely to be about the same topic than a random pair of posts. To model this observation, we assume that there is a global topic distribution θ^t for each time point t. Presumably θ^t has a high probability for a topic that is popular in the microblogsphere at time t.

Unlike news articles from traditional media, which are mostly about current affairs, an important property of microblog posts is that many posts are about users' personal encounters and interests rather than global events. Since our focus is to find popular global events, we need to separate out these "personal" posts. To do this, an intuitive idea is to compare a post with its publisher's general topical interests observed over a long time. If a post does not match the user's long term interests, it is more likely related to a global event. We therefore introduce a time-independent topic distribution η^u for each user to capture her long term topical interests.

We assume the following generation process for all the posts in the stream. When user u publishes a post at time point t, she first decides whether to write about a global trendy topic or a personal topic. If she chooses the former, she then selects a topic according to θ^t. Otherwise, she selects a topic according to her own topic distribution η^u. With the chosen topic, words in the post are generated from the word distribution for that topic or from the background word distribution that captures white noise. We use π to denote the probability of choosing to talk about a

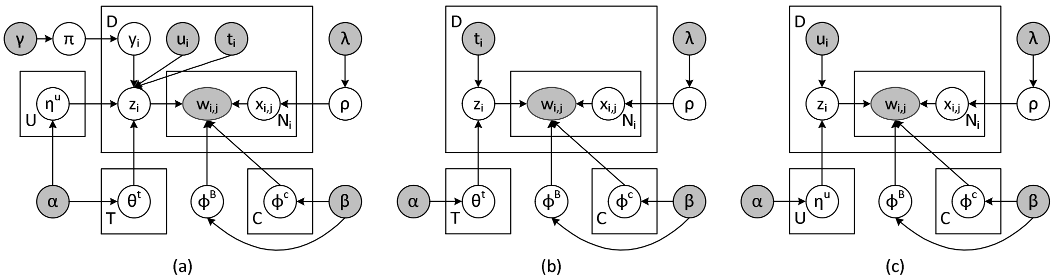


Figure 3.1: (a) Our topic model for burst detection. (b) A variation of our model where we only consider global topical trends. (c) A variation of our model where we only consider users' personal topical interests.

global topic rather than a personal topic.

Formally, the generation process is summarized in Figure 3.2. The model is also depicted in Figure 3.1(a).

There are two degenerate variations of our model that we also consider in our experiments. The first one is depicted in Figure 3.1(b). In this model, we only consider the time-dependent topic distributions that capture the global topical trends. This model can be seen as a direct application of the model by Wang et al [49]. The second one is depicted in Figure 3.1(c). In this model, we only consider the users' personal interests but not the global topical trends, and therefore temporal information is not used. We refer to our complete model as *TimeUserLDA*, the model in Figure 3.1(b) as *TimeLDA* and the model in Figure 3.1(c) as *UserLDA*. We also consider a standard LDA model in our experiments, where each word is associated with a hidden topic.

3.2.3 Learning

We use collapsed Gibbs sampling to obtain samples of the hidden variable assignment and to estimate the model parameters from these samples.

First, for the i-th post, we know its publisher u_i and timestamp t_i. We can jointly sample y_i and z_i based on the values of all other hidden variables. Let us use $\mathbf{y}$ to denote the set of all hidden variables y and $\mathbf{y}_{\neg i}$ to denote all y except y_i. We use

1. Draw $\phi^B \sim \text{Dirichlet}(\beta), \pi \sim \text{Beta}(\gamma), \rho \sim \text{Beta}(\lambda)$
2. For each time point $t = 1, \ldots, T$
   (a) draw $\theta^t \sim \text{Dirichlet}(\alpha)$
3. For each user $u = 1, \ldots, U$
   (a) draw $\eta^u \sim \text{Dirichlet}(\alpha)$
4. For each topic $c = 1, \ldots, C$,
   (a) draw $\phi^c \sim \text{Dirichlet}(\beta)$
5. For each post $i = 1, \ldots, D$,
   (a) draw $y_i \sim \text{Bernoulli}(\pi)$
   (b) draw $z_i \sim \text{Multinomial}(\eta^{u_i})$ if $y_i = 0$ or $z_i \sim \text{Multinomial}(\theta^{t_i})$ if $y_i = 1$
   (c) for each word $j = 1, \ldots, N_i$
      i. draw $x_{i,j} \sim \text{Bernoulli}(\rho)$
      ii. draw $w_{i,j} \sim \text{Multinomial}(\phi^B)$ if $x_{i,j} = 0$ or $w_{i,j} \sim \text{Multinomial}(\phi^{z_i})$ if $x_{i,j} = 1$

Figure 3.2: The generation process for all posts.

similar symbols for other variables. We then have:

$$
\begin{aligned}
& p(y_i = p, z_i = c | \mathbf{z}_{\neg i}, \mathbf{y}_{\neg i}, \mathbf{x}, \mathbf{w}) \\
& \propto p(y_i = p | \mathbf{y}_{\neg i}) \cdot p(z_i = c | y_i = p, \mathbf{z}_{\neg i}, \mathbf{x}, \mathbf{w}) \\
& \propto p(y_i = p | \mathbf{y}_{\neg i}) \cdot p(z_i = c | y_i = p, \mathbf{z}_{\neg i}) \cdot p(\mathbf{w}_i | z_i = c, \mathbf{x}_i, \mathbf{z}_{\neg i}, \mathbf{x}_{\neg i}, \mathbf{w}_{\neg i}) \\
& \propto \int p(y_i = p | \pi) p(\pi | \mathbf{y}_{\neg i}) \mathrm{d}\pi \cdot \left(\int p(z_i = c | \eta^{u_i}) p(\eta^{u_i} | \mathbf{z}_{\neg i}) \mathrm{d}\eta^{u_i} \right)^{1-p} \\
& \cdot \left(\int p(z_i = c | \theta^{t_i}) p(\theta^{t_i} | \mathbf{z}_{\neg i}) \mathrm{d}\theta^{t_i} \right)^{p} \cdot \int \prod_j p(w_{i,j} | \phi^c)^{x_{i,j}} p(\phi^c | \mathbf{z}_{\neg i}, \mathbf{x}_{\neg i}, \mathbf{w}_{\neg i}) \mathrm{d}\phi^c \\
& \propto \frac{M^{\pi}_{(p)} + \gamma}{M^{\pi}_{(\cdot)} + 2\gamma} \cdot \frac{M^{l}_{(c)} + \alpha}{M^{l}_{(\cdot)} + C\alpha} \cdot \frac{\prod_{v=1}^{V} \prod_{k=0}^{E_{(v)}-1} (M^{c}_{(v)} + k + \beta)}{\prod_{k=0}^{E_{(\cdot)}-1} (M^{c}_{(\cdot)} + k + V\beta)}, \qquad (3.1)
\end{aligned}
$$

where $l = u_i$ when $p = 0$ and $l = t_i$ when $p = 1$. Here every M is a counter. $M^{\pi}_{(0)}$ is the number of posts generated by personal interests, while $M^{\pi}_{(1)}$ is the number of posts coming from global topical trends. $M^{\pi}_{(\cdot)} = M^{\pi}_{0} + M^{\pi}_{1}$. $M^{u_i}_{(c)}$ is the number of posts by user u_i and assigned to topic c, and $M^{u_i}_{(\cdot)}$ is the total number of posts by u_i. $M^{t_i}_{(c)}$ is the number of posts assigned to topic c at time point t_i, and $M^{t_i}_{(\cdot)}$ is the total number of posts at t_i. $E_{(v)}$ is the number of times word v occurs in the i-th post and is labeled as a topic word, while $E_{(\cdot)}$ is the total number of topic words in the i-th post. Here, topic words refer to words whose latent variable x equals 1. $M^{c}_{(v)}$ is the number of times word v is assigned to topic c, and $M^{c}_{(\cdot)}$ is the total number of words assigned to topic c. All the counters M mentioned above are calculated with the i-th post excluded.

We sample $x_{i,j}$ for each word $w_{i,j}$ in the i-th post using:

$$\begin{aligned}
&p(x_{i,j} = q|\mathbf{y}, \mathbf{z}, \mathbf{x}_{\neg\{i,j\}}, \mathbf{w}) \\
&\propto p(x_{i,j} = q|\mathbf{x}_{\neg\{i,j\}}) \cdot p(w_{i,j}|x_{i,j} = q, \mathbf{x}_{\neg\{i,j\}}, \mathbf{y}, \mathbf{z}, \mathbf{w}_{\neg\{i,j\}}) \\
&\propto \int p(x_{i,j} = q|\rho)p(\rho|\mathbf{x}_{\neg\{i,j\}})\mathrm{d}\rho \cdot \int p(w_{i,j}|\phi^{l})p(\phi^{l}|x_{i,j} = q, \mathbf{x}_{\neg\{i,j\}}, \mathbf{y}, \mathbf{z}, \mathbf{w}_{\neg\{i,j\}})\mathrm{d}\phi^{l} \\
&\propto \frac{M^{\rho}_{(q)} + \lambda}{M^{\rho}_{(\cdot)} + 2\lambda} \cdot \frac{M^{l}_{(w_{i,j})} + \beta}{M^{l}_{(\cdot)} + V\beta}, \qquad (3.2)
\end{aligned}$$

where $l = B$ when $q = 0$ and $l = z_i$ when $q = 1$. $M^{\rho}_{(0)}$ and $M^{\rho}_{(1)}$ are counters to record the numbers of words assigned to the background model and any topic, respectively, and $M^{\rho}_{(\cdot)} = M^{\rho}_{(0)} + M^{\rho}_{(1)}$. $M^{B}_{(w_{i,j})}$ is the number of times word $w_{i,j}$ occurs as a background word. $M^{z_i}_{(w_{i,j})}$ counts the number of times word $w_{i,j}$ is assigned to topic z_i, and $M^{z_i}_{(\cdot)}$ is the total number of words assigned to topic z_i. Again, all counters are calculated with the current word $w_{i,j}$ excluded.

3.2.4 Burst Detection

Just like standard LDA, our topic model itself finds a set of topics represented by ϕ^{c} but does not directly generate bursty topics. To identify bursty topics, we use the following mechanism, which is based on the idea by Jon Kleinberg [29] and Ihler et [28]. In our experiments, when we compare different models, we also use the same burst detection mechanism for other models.

We assume that after topic modeling, for each discovered topic c, we can obtain a series of counts $(m_1^c, m_2^c, \dots, m_T^c)$ representing the intensity of the topic at different time points. For LDA, these are the numbers of words assigned to topic c. For TimeUserLDA, these are the numbers of posts which are in topic c and generated by the global topic distribution θ^{t_i}, i.e whose hidden variable y_i is 1. For other models, these are the numbers of posts in topic c.

We assume that these counts are generated by two Poisson distributions corresponding to a bursty state and a normal state, respectively. Let μ_0 denote the expected count for the normal state and μ_1 for the bursty state. Let v_t denote the state for time point t, where $v_t = 0$ indicates the normal state and $v_t = 1$ indicates the bursty state. The probability of observing a count of m_t^c is as follows:

$$p(m_t^c | v_t = l) = \frac{e^{-\mu_l} \mu_l^{m_t^c}}{m_t^c!}, \tag{3.3}$$

where l is either 0 or 1. The state sequence $(v_0, v_1, \dots, v_T)$ is a Markov chain with the following transition probabilities:

$$p(v_t = l | v_{t-1} = l) = \sigma_l, \tag{3.4}$$

where l is either 0 or 1.

μ_0 and μ_1 are topic specific. In our experiments, we set $\mu_0 = \frac{1}{T} \sum_t m_t^c$, that is, μ_0 is the average count over time. We set $\mu_1 = 3\mu_0$. For transition probabilities, we empirically set $\sigma_0 = 0.9$ and $\sigma_1 = 0.6$ for all topics.

We can use dynamic programming to uncover the underlying state sequence for a series of counts. Finally, a burst is marked by a consecutive subsequence of bursty states.

Method	P@5	P@10	P@20	P@30
TimeLDA	0.800	0.700	0.600	0.633
UserLDA	0.800	0.700	0.850	**0.833**
TimeUserLDA	**1.000**	**1.000**	**0.900**	0.800

Table 3.1: Precision at K for the various models.

Method	P@5	P@10	P@20	P@30
LDA	0.600	0.800	0.700	N/A
TimeLDA	0.400	0.500	0.500	0.567
UserLDA	0.800	0.500	0.500	0.600
TimeUserLDA	**1.000**	**0.900**	**0.850**	**0.767**

Table 3.2: Precision at K for the various models after we remove redundant bursty topics.

3.3 Experiments

3.3.1 Data Set

We use a Twitter data set to evaluate our models. The original data set contains 151,055 Twitter users based in Singapore and their tweets. These Twitter users were obtained by starting from a set of seed Singapore users who are active online and tracing their follower/followee links by two hops. Because this data set is huge, we randomly sampled 2892 users from this data set and extracted their tweets between September 1 and November 30, 2011 (91 days in total). We use one day as our time window. Therefore our timestamps range from 1 to 91. We then removed stop words and words containing non-standard characters. Tweets containing less than 3 words were also discarded. After preprocessing, we obtained the final data set with 3,967,927 tweets and 24,280,638 tokens.

3.3.2 Ground Truth Generation

To compare our model with other alternative models, we perform both quantitative and qualitative evaluation. As we have explained in Section 3.2.2, each model gives us time series data for a number of topics, and by applying a Poisson-based

Bursty Period	Top Words	Example Tweets	Label
Nov 29	vote, big, awards, bang, mama, win, 2ne1, award, won	(1) why didnt 2ne1 win this time! (2) 2ne1. you deserved that urgh! (3) watching mama. whoohoo	Mnet Asian Music Awards (MAMA)
Oct 5 ∼ Oct 8	steve, jobs, apple, iphone, rip, world, changed, 4s, siri	(1) breaking: apple says steve jobs has passed away! (2) google founders: steve jobs was an inspiration! (3) apple 4 life thankyousteve	Steve Jobs death
Nov 1 ∼ Nov 3	reservior, bedok, adlyn, slap, found, body, mom, singapore, steven	(1) this adelyn totally disgust me. slap her mum? queen of cine? joke please can. (2) she slapped her mum and boasted about it on fb (3) adelyn lives in woodlands , later she slap me how?	girl slapping mom
Nov 5	reservior, bedok, adlyn, slap, found, body, mom, singapore, steven	(1) bedok = bodies either drowned or killed. (2) another body found, in bedok reservoir? (3) so many bodies found at bedok reservoir. alamak.	suicide near bedok reservoir
Oct 23	man, arsenal, united, liverpool, chelsea, city, goal, game, match	(1) damn you man city! we will get you next time! (2) wtf 90min goal! (3) 6-1 to city. unbelievable.	football game

Table 3.3: Top-5 bursty topics ranked by TimeUserLDA. The labels are manually given. The 3rd and the 4th bursty topics come from the same topic but have different bursty periods.

state machine, we can obtain a set of bursty topics. For each method, we rank the obtained bursty topics by the number of tweets (or words in the case of the LDA model) assigned to the topics and take the top-30 bursty topics from each model. In the case of the LDA model, only 23 bursty topics were detected. We merged these topics and asked two human judges to judge their quality by assigning a score of either 0 or 1. The judges are graduate students living in Singapore and not involved in this project. The judges were given the bursty period and 100 randomly selected tweets for the given topic within that period for each bursty topic. They can consult external resources to help make judgment. A bursty topic was scored 1 if the 100 tweets coherently describe a bursty event based on the human judge's understand-

Rank	LDA	UserLDA	TimeLDA
1	Steve Jobs' death	MAMA	MAMA
2	MAMA	football game	MAMA
3	N/A	#zamanprimaryschool	MAMA
4	girl slapping mom	N/A	girl slapping mom
5	N/A	iphone 4s	N/A

Table 3.4: Top-5 bursty topics ranked by other models. N/A indicates a meaningless burst.

ing. The inter-annotator agreement score is 0.649 using Cohen's kappa, showing substantial agreement. For ground truth, we consider a bursty topic to be correct if both human judges have scored it 1. Since some models gave redundant bursty topics, we also asked one of the judges to identify unique bursty topics from the ground truth bursty topics.

3.3.3 Evaluation

In this section, we show the quantitative evaluation of the four models we consider, namely, LDA, TimeLDA, UserLDA and TimeUserLDA. For each model, we set the number of topics C to 80, α to $\frac{50}{C}$ and β to 0.01 after some preliminary experiments. Each model was run for 500 iterations of Gibbs sampling. We take 40 samples with a gap of 5 iterations in the last 200 iterations to help us assign values to all the hidden variables.

Table 3.1 shows the comparison between these models in terms of the precision of the top-K results. As we can see, our model outperforms all other models for $K <= 20$. For $K = 30$, the UserLDA model performs the best followed by our model.

As we have pointed out, some of the bursty topics are redundant, i.e. they are about the same bursty event. We therefore also calculated precision at K for unique topics, where for redundant topics the one ranked the highest is scored 1 and the other ones are scored 0. The comparison of the performance is shown in Table 3.2. As we can see, in this case, our model outperforms other models with all K. We will further discuss redundant bursty topics in the next section.

3.3.4 Sample Results and Discussions

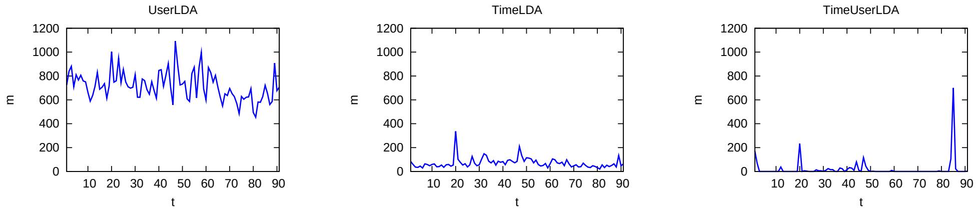


Figure 3.3: Topic intensity over time for the topic on the Circle Line.

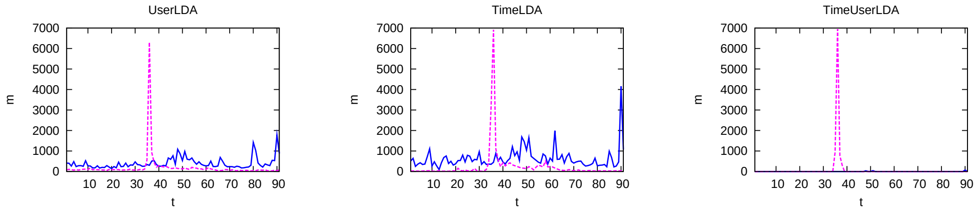


Figure 3.4: Topic intensity over time for the topic about a Korean pop singer. The dotted curves show the topic on Steve Jobs' death.

In this section, we show some sample results from our experiments and discuss some case studies that illustrate the advantages of our model.

First, we show the top-5 bursty topics discovered by the TimeUserLDA model in Table 3.3. As we can see, all these bursty topics are meaningful. Some of these events are global major events such as Steve Jobs' death, while some others are related to online events such as the scandal of a girl boasting about slapping her mother on Facebook. For comparison, we also show the top-5 bursty topics discovered by other models in Table 3.4. As we can see, some of them are not meaningful events while some of them are redundant.

Next, we show two case studies to demonstrate the effectiveness of our model.

Effectiveness of Temporal Models: Both TimeLDA and TimeUserLDA tend to group posts published on the same day into the same topic. We find that this can help separate bursty topics from general ones. An example is the topic on the Circle Line. The Circle Line is one of the subway lines of Singapore's mass transit system. There were a few incidents of delays or breakdowns during the period between September and November, 2011. We show the time series data of the topic related to the Circle Line of UserLDA, TimeLDA and TimeUserLDA in Figure 3.3. As we can see, the UserLDA model detects a much larger volume of tweets related to this topic. A close inspection tells us that the topic under UserLDA is actually related to the subway systems in Singapore in general, which include a few other subway lines, and the Circle Line topic is merged with this general topic. On the other hand, TimeLDA and TimeUserLDA are both able to separate the Circle Line topic from the general subway topic because the Circle Line has several bursts. What is shown in Figure 3.3 for TimeLDA and TimeUserLDA is only the topic on the Circle Line, therefore the volume is much smaller. We can see that TimeLDA and TimeUserLDA show clearer bursty patterns than UserLDA for this topic. The bursts around day 20, day 44 and day 85 are all real events based on our ground truth.

Effectiveness of User Models: We have stated that it is important to filter out users' "personal" posts in order to find meaningful global events. We find that our results also support this hypothesis. Let us look at the example of the topic on the Mnet Asian Music Awards, which is a major music award show that is held by Mnet Media annually. In 2011, this event took place in Singapore on November 29. Because Korean pop music is very popular in Singapore, many Twitter users often tweet about Korean pop music bands and singers in general. All our topic models give multiple topics related to Korean pop music, and many of them have a burst on November 29, 2011. Under the TimeLDA and UserLDA models, this leads to several redundant bursty topics for the MAMA event ranked within the top-30. For TimeUserLDA, however, although the MAMA event is also ranked the top, there is no redundant one within the top-30 results. We find that this is because with TimeUserLDA, we can remove tweets that are considered personal and therefore do not contribute to bursty topic ranking. We show the topic intensity of a topic about a Korean pop singer in Figure 3.4. For reference, we also show the intensity of the topic on Steve Jobs' death under each model. We can see that because this topic is related to Korean pop music, it has a burst on day 90 (November 29). But if we consider the relative intensity of this burst compared with Steve Jobs' death, under TimeLDA and UserLDA, this topic is still strong but under TimeUserLDA its intensity can almost be ignored. This is why with TimeLDA and UserLDA this topic leads to a redundant burst within the top-30 results but with TimeUserLDA the burst is not ranked high.

3.4 Conclusions

In this chapter, we studied the problem of finding bursty topics from the text streams on microblogs. Because existing work on burst detection from text streams may not be suitable for microblogs, we proposed a new topic model that considers both the temporal information of microblog posts and users' personal interests. We then applied a Poisson-based state machine to identify bursty periods from the topics discovered by our model. We compared our model with standard LDA as well as two degenerate variations of our model on a real Twitter dataset. Our quantitative evaluation showed that our model could more accurately detect unique bursty topics among the top ranked results. We also used two case studies to illustrate the effectiveness of the temporal factor and the user factor of our model.

Chapter 4

Recurrent Chinese Restaurant Process with a Duration-based Discount for Event Identification from Twitter

In the previous chapter, we proposed an LDA-based model to capture bursty topics and a two-state machine to identify events. In this chapter, we turn to a nonparametric model to directly model events on Twitter. Recently the Recurrent Chinese Restaurant Process (RCRP) has been successfully used for event identification from news streams and news-centric social media streams. However, these models cannot be directly applied to Twitter based on our preliminary experiments mainly for two reasons: (1) Events emerge and die out fast on Twitter, while existing models ignore this burstiness property. (2) Most Twitter posts are personal interest oriented while only a small fraction is event related. Motivated by these challenges, we propose a new nonparametric model which considers burstiness. We further combine this model with traditional topic models to identify both events and topics simultaneously. Our quantitative evaluation provides sufficient evidence that our model can accurately detect meaningful events. Our qualitative evaluation also shows interesting analysis for events on Twitter.

4.1 Introduction

In the previous chapter, event identification is done through a post-processing step using a two-state machine. In this chapter, we define the task of event identification directly as to identify (gapped) subsequences of tweets from a segment of Twitter stream where each subsequence contains tweets discussing the same event. Figure 1.1 illustrates the problem definition and shows some example events with their representative tweets.

The problem can be regarded as an evolutionary clustering problem, where items are ordered as a stream and clustered depending on not only their similarity but also their closeness in time. For evolutionary clustering of streaming documents, several methods have been proposed, including some from the information retrieval community to address the event detection problem under Topic Detection and Tracking (TDT) (e.g. [54, 7, 55]) and others from the machine learning and data mining communities (e.g. [2, 5]). In particular, Ahmed and Xing proposed a dynamic non-parametric model called the Recurrent Chinese Restaurant Process (RCRP), which performs evolutionary clustering of streaming documents in a principled and elegant way [5]. Being a non-parametric model, it also allows a countably infinite number of clusters and flexibly models the life cycle of each cluster. Because of these appealing characteristics, we choose RCRP as the basis of our solution.

Although RCRP has been successfully applied to find events from news streams [3] and news-centric social media streams [46], Twitter has some major differences from news streams and therefore these existing models are not directly applicable to our problem. (1) Existing models assume that all documents are event-related and must be assigned to a cluster. On Twitter, however, many tweets are not related to any significant event. According to a Twitter study by PearAnalytics[1],

[1] `http://www.pearanalytics.com/wp-content/uploads/2012/12/`

only 3.6% of tweets are news-related and 8.7% have pass-along value. The majority of tweets are about people's personal interests and daily routines. We therefore separate tweets into *topic tweets* and *event tweets*, which capture user's personal life topics and major events on Twitter respectively. We identify the former using a topic model and the latter using a RCRP-based model. Although this assumption is a much simplified view of the wide range of tweets, we find it effective to detect meaningful events and topics. (2) RCRP does not model the phenomenon that events on Twitter are bursty. Because of the nature of microblogs, people usually use Twitter to spread or comment on breaking news rather than old events, which means events on Twitter tend to die out fast. However, RCRP only captures the "rich get richer" phenomenon. We therefore need to introduce some mechanism to favor bursty clusters.

In this chapter, we propose a new non-parametric generative model for identifying events from Twitter. Following [3] and [46], our model distinguishes between longstanding topics and bursty events. In our model, only events are modeled by RCRP and allowed to emerge and disappear along the timeline. Different from the previous models, we separate topical tweets from event-related tweets by considering each user's longstanding topical interests. Moreover, we introduce a novel duration-based probability discount into RCRP, which penalizes longstanding events and hence models the burstiness of events on Twitter.

We evaluate our model on a real Twitter dataset that contains the posts of 500 users published during a period of three months from April to June 2012. Our experiments show that our proposed model can more accurately identify meaningful events than two baseline methods. Our model also finds more relevant tweets and generates better temporal profiles of events.

Our work has the following contributions: (1) We propose a principled unified probabilistic model for event identification on Twitter. Each event forms its own cluster inside the model and no post-processing is needed. (2) Event-related

`Twitter-Study-August-2009.pdf`

tweets can be separated from personal topical tweets automatically within our unified model. (3) We propose a novel duration-based probability discount for RCRP, which allows us to capture the burstiness of events on microblogs.

4.2 Method

We first briefly review RCRP and its application for event identification. Because our preliminary experiments show that existing RCRP based models cannot be directly applied in our task, we then introduce our method, which extends RCRP.

4.2.1 Recurrent Chinese Restaurant Process

The Recurrent Chinese Restaurant Process (RCRP) is a non-parametric model for evolutionary clustering proposed by Ahmed and Xing [5], which basically chains up the Chinese Restaurant Process (CRP) [10] based on the timeline. To model streaming data, RCRP models a restaurant with infinite number of tables and customers coming on different days. When the i-th customer on the t-th day comes in, she can choose a table that either is serving some customers on day t or served some customers on day $t-1$ (or both) with probability $\frac{n_{k,t-1}+n_{k,t}^{(i)}}{N_{t-1}+i-1+\alpha}$, where $n_{k,t-1}$ is the number of customers sitting at table k at the end of day $t-1$, $n_{k,t}^{(i)}$ is the number of customers sitting at table k on day t before customer i comes, and N_{t-1} is the total number of customers served by the restaurant on day $t-1$. This customer can also choose to sit at a new empty table that did not even serve any customer on the previous day with probability $\frac{\alpha}{N_{t-1}+i-1+\alpha}$. With the RCRP metaphor, we can cluster a sequence of items that are divided into epochs. Each resulting cluster not only contains a set of items but also has a duration with a start time and an end time. The RCRP model encourages popular clusters in epoch $t-1$ to remain alive in epoch t. Under RCRP, items from different epochs are no longer exchangeable. When RCRP is applied for document clustering, we further assume that each cluster is associated with a multinomial word distribution. We can then model the generation of documents from a cluster where each document is a bag of words. Such a model allows us to prefer documents with similar word usage to be clustered together.

The RCRP model can be used to cluster news articles into storylines where each story is a series of news articles about the same event [3, 46]. Ahmed et al. [3] proposed a RCRP-LDA model which assumes that there is a fixed number of topics that exist at all times and an infinite number of events that emerge and disappear over time. Each document is assumed to belong to an event, but words inside a document can be either topical or event related. Tang et al. further extended the RCRP-LDA model to incorporate user interests by assuming that each event has a user distribution [46]. They applied their model to some news-centric social media streams such as Digg and online discussion forums.

4.2.2 Our Motivation

However, as we have stated earlier, a major difference between Twitter and news article streams is that the majority of tweets is about trivial events and personal interests, while only a small fraction of them is event related. Therefore, we cannot directly apply existing RCRP based models or TDT methods, which assume each document is related to an event. To illustrate this difference, we apply two representative existing methods for event identification from news articles to a subset of our Twitter data from September to November 2011. The first one is a TDT method from [55], which aims to detect events from news streams retrospectively using hierarchical group average clustering. The second method is from [46] for identifying events from news-centric social media streams using RCRP and LDA. We show some top-ranked events identified by the two methods in terms of top keywords in Table 4.1. For each event, we also plot a temporal profile that shows the volume of identified relevant tweets over time. We can see that the top keywords and the temporal profiles do not clearly indicate any important event. Besides the special property that not every tweet is event-related, another characteristic of events on

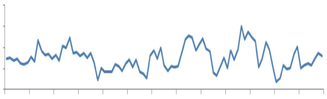
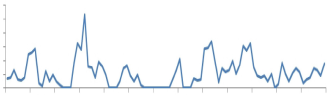
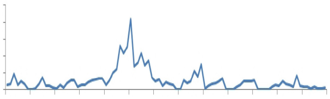
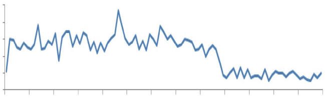
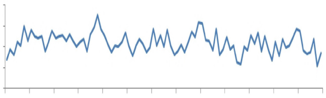
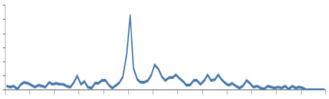
Top words	Temporal profile
time, love, good, people, la, sleep, today, im, day, work	
day, #nowplaying, gonna, video, song, there’s, yeah, wait, find, love	
steve, jobs, love, time, good, feel, rip, happy, damn, miss	
people, life, make, love, moment, person, awkward, hate, smile, things	
sleep, tired, im, home, bed, early, wake, gonna, rain, feel	
iphone, steve, jobs, apple, 4s, ios, app, phone, siri, rip	

Table 4.1: Three of the top-ranked events identified by the models in [55] (top three) and [46] (bottom three) from our data.

Twitter is that they tend to be bursty. Standard RCRP only models the “rich get richer” phenomenon, which can lead to events with long durations.

To address the two problems above, we propose a different RCRP-based generative model for identifying events from Twitter. The proposed model assumes that a tweet is either topical or event-related. It further introduces a duration-based probability discount to favor bursty events.

4.2.3 Preliminaries

We first formally formulate our problem before we go to the detail of our method. We assume that we have a stream of tweets that are divided into T epochs. (In our experiments, we use one day as an epoch.) Let $t \in \{1, 2, \ldots, T\}$ denote the index of an epoch. Each epoch contains a sequence of tweets ordered by their exact time stamps, and each tweet is a bag of words. Let V be the size of our vocabulary and let $w_{t,i,j} \in \{1, 2, \ldots, V\}$ denote the j-th word (represented by its index in the vocabulary) from the i-th tweet in the t-th epoch. We also take note of the authors of these tweets. Let $u_{t,i} \in \{1, 2, \ldots, U\}$ denote the user who published the i-th tweet in the t-th epoch, where U is the total number of users. Our goal is to identify a set of events from these tweets, where each event is a set of tweets. Note that not every tweet has to belong to an event.

Our general idea is to cluster these tweets such that each cluster represents an event. But since not all tweets are event-related, we assume that each tweet is either about a general longstanding topic (a *topical* tweet) or related to an event (an *event-related* tweet). Only the event-related tweets will be clustered using the Recurrent Chinese Restaurant Process. For the topical tweets, we assume that they are closely related to each user's topical interests.

4.2.4 A Duration-based Discount for RCRP

Recall that one problem we have identified with RCRP for Twitter is that RCRP only models the "rich get richer" phenomenon. In other words, popular events tend to attract even more users to tweet about them. However, on microblogs users also tend to follow the newest trends. Once an event becomes old, it may no longer attract much attention. In fact, [31] identified these two factors on both mainstream and social media and termed them *imitation* and *recency*. They argued that any model of the news cycle needs to incorporate some version of these two ingredients. RCRP already captures the imitation factor. What is missing is the recency factor.

We therefore propose the following change to the standard RCRP. Recall that in the RCRP metaphor, when the i-th customer of the t-th epoch comes in, the probability to join an existing table k is proportional to $(n_{k,t-1} + n_{k,t}^{(i)})$, i.e. the number of customers sitting at table k on the current and the previous days before customer i comes. Let $\bar{t}_k$ denote the index of the epoch when table k was first occupied. Based on the recency effect, the earlier a table was first occupied, the older the table is and the less likely it will be chosen. We hence want to discount the probability mass to join table k based on $(t - \bar{t}_k)$. Here we propose a discount of $(n_{k,t-1} + n_{k,t}^{(i)})(1 - e^{-\lambda(t-\bar{t}_k)})$, that is, after the discount, the remaining probability mass is $(n_{k,t-1} + n_{k,t}^{(i)})e^{-\lambda(t-\bar{t}_k)}$. Here $\lambda > 0$ is a parameter we can tune. It is obvious that the older table k is, the smaller $\bar{t}_k$ is and the smaller the probability mass for table k is after the discount. On the other hand, the deducted probability mass will be used for starting a new table.

Formally, define $\Delta_{k,t}^{(i)}$ as $(n_{k,t-1} + n_{k,t}^{(i)})(1 - e^{-\lambda(t-\bar{t}_k)})$, the duration-based probability discount. Then for the i-th customer of the t-th epoch, she can choose to join an existing table with probability $\frac{n_{k,t-1}+n_{k,t}^{(i)}-\Delta_{k,t}^{(i)}}{N_{t-1}+i-1+\alpha}$ or start a new table with probability $\frac{\alpha+\sum_{k'}\Delta_{k',t}^{(i)}}{N_{t-1}+i-1+\alpha}$.

With the discounted RCRP model, customers prefer not only popular tables but also “fresh” tables. This is the major distinction of our proposed model from the standard RCRP. The discount model also maintains the total probability mass as $(N_{t-1} + i - 1 + \alpha)$, which simplifies the model inference later.

4.2.5 The Complete Model

We are now ready to formally present our complete model for event identification from Twitter. We assume that there are A longstanding topics, each associated with a multinomial word distribution ϕ_a. Each user u has a topic distribution θ_u. Events are formed through the Recurrent Chinese Restaurant Process with the duration-based discount, and each event k also has a multinomial word distribution ψ_k.

During the t-th epoch, for the i-th tweet, a binary variable $y_{t,i}$ is first sampled from a user-specific Bernoulli distribution $\pi_{u_{t,i}}$, which indicates a user’s tendency to post topical or event-related tweets. If $y_{t,i}$ equals 0, a topic $z_{t,i}$ is sampled from the user’s topic distribution $\theta_{u_{t,i}}$. Then all words in this tweet are sampled from the

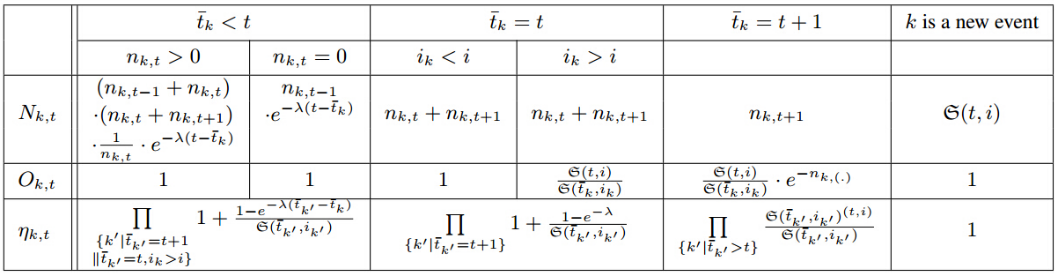
```
1. For each topic $a = 1, \ldots, A$
   (a) draw $\phi_a \sim \mathrm{Dirichlet}(\beta)$
2. For each user $u = 1, \ldots, U$
   (a) draw $\theta_u \sim \mathrm{Dirichlet}(\gamma), \pi_u \sim \mathrm{Beta}(\tau)$
3. For each $t$ and each $i$
   (a) draw $y_{t,i} \sim \mathrm{Bernoulli}(\pi_{u_{t,i}})$
   (b) if $y_{t,i} = 0$
       i. draw $z_{t,i} \sim \mathrm{Discrete}(\theta_{u_{t,i}})$
       ii. for all $j$, draw $w_{t,i,j} \sim \mathrm{Discrete}(\phi_{z_{t,i}})$
   (c) if $y_{t,i} = 1$
       i. draw $s_{t,i}$ from the RCRP with discount
       ii. if $s_{t,i}$ is a new event
           A. draw $\psi_{s_{t,i}} \sim \mathrm{Dirichlet}(\beta)$
           B. set $\bar{t}_{s_{t,i}}$ equal to $t$
       iii. for all $j$, draw $w_{t,i,j} \sim \mathrm{Discrete}(\psi_{s_{t,i}})$
```

Figure 4.1: The generative process of our model.

word distribution $\phi_{z_{t,i}}$. If $y_{t,i}$ equals 1, then an event $s_{t,i}$ is sampled from a Recurrent Chinese Restaurant Process with the proposed duration-based discount. All words in this tweet are then sampled from the word distribution $\psi_{s_{t,i}}$.

We place uniform Dirichlet priors over all the multinomial distributions. The generative process is also described in Figure 4.1.

A major difference between the RCRP-LDA models in [3] and [46] and our model is that the RCRP-LDA models differentiate between topics and events at the *word level*, i.e. they allow a document to contain both topical words and event-specific words, whereas in our model the entire content of a tweet is either topical or event-related.Our preliminary experiment shows that when we apply such a setting to Twitter, many tweets end up containing only topical words but are still wrongly assigned to some event which is not related to the content of the tweet. We therefore



	$\bar{t}_k < t$		$\bar{t}_k = t$		$\bar{t}_k = t + 1$	k is a new event
	$n_{k,t} > 0$	$n_{k,t} = 0$	$i_k < i$	$i_k > i$		
$N_{k,t}$	$(n_{k,t-1} + n_{k,t})$ $\cdot(n_{k,t} + n_{k,t+1})$ $\cdot\frac{1}{n_{k,t}} \cdot e^{-\lambda(t-\bar{t}_k)}$	$n_{k,t-1}$ $\cdot e^{-\lambda(t-\bar{t}_k)}$	$n_{k,t} + n_{k,t+1}$	$n_{k,t} + n_{k,t+1}$	$n_{k,t+1}$	$\mathfrak{S}(t,i)$
$O_{k,t}$	1	1	1	$\frac{\mathfrak{S}(t,i)}{\mathfrak{S}(\bar{t}_k,i_k)}$	$\frac{\mathfrak{S}(t,i)}{\mathfrak{S}(\bar{t}_k,i_k)} \cdot e^{-n_{k,(\cdot)}}$	1
$\eta_{k,t}$	$\prod_{\{k' \mid \bar{t}_{k'}=t+1 \\ \| \bar{t}_{k'}=t, i_k>i\}} 1 + \frac{1-e^{-\lambda(\bar{t}_{k'}-\bar{t}_k)}}{\mathfrak{S}(\bar{t}_{k'},i_{k'})}$		$\prod_{\{k' \mid \bar{t}_{k'}=t+1\}} 1 + \frac{1-e^{-\lambda}}{\mathfrak{S}(\bar{t}_{k'},i_{k'})}$		$\prod_{\{k' \mid \bar{t}_{k'}>t\}} \frac{\mathfrak{S}(\bar{t}_{k'},i_{k'})^{(t,i)}}{\mathfrak{S}(\bar{t}_{k'},i_{k'})}$	1

Figure 4.2: For the formula of sampling events, $N_{k,t}$ $O_{k,t}$ and $\eta_{k,t}$ vary under different conditions.

differentiate between topics and events at the tweet level instead. Also, we do not consider named entities as [3] do because NER on Twitter is less accurate and faces more name variations.

4.2.6 Model Inference

We use collapsed Gibbs sampling to obtain samples of the latent variables based on the conditional distributions derived from our model and finally use these samples to obtain the final hidden label assignment. We find that the conditional probabilities derived from our model are rather complex. This is because unlike the Chinese Restaurant Process, where items are exchangeable, or the Recurrent Chinese Restaurant Process, where items within the same epoch are exchangeable, our model lacks complete exchangeability because of the duration-based discount. While we are able to derive the exact formulas for the conditional probabilities, we find that in terms of efficiency, the exact formulas would incur high computational costs and are not feasible given the large volume of tweets. We then opt for some approximation of the exact sampling formulas. We remove the terms that do not affect the probabilities much and keep the terms that dominate the probability mass. In the content that follows, we first derive the exact formulas for conditional probabilities in detail and then describe the approximation.

For the exact conditional probabilities, we jointly sample $y_{t,i}$, $z_{t,i}$ and $s_{t,i}$. The formulas for $y_{t,i} = 0, z_{t,i} = a$ and $y_{t,i} = 1, s_{t,i} = k$ are different.

Topical:

First of all, for $y_{t,i} = 0$ and $z_{t,i} = a$, we have the following formula:

$$\begin{aligned} p(y_{t,i} = 0, z_{t,i} = a | \mathbf{y}_{\neg(t,i)}, \mathbf{z}_{\neg(t,i)}, \mathbf{w}) \propto \frac{n_{u,0}^{(\pi)} + \tau}{n_{u,(.)}^{(\pi)} + 2\tau} \\ \cdot \frac{n_{u,a}^{(\theta)} + \gamma}{n_{u,(.)}^{(\theta)} + A\gamma} \cdot \frac{\prod_{v=1}^{V} \prod_{l=0}^{E_{(v)}-1} (n_{a,v}^{(\phi)} + l + \beta)}{\prod_{l=0}^{E_{(.)}-1} (n_{a,(.)}^{(\phi)} + l + V\beta)}, \end{aligned}$$

where we use u to represent author $u_{t,i}$. $n_{u,0}^{(\pi)}$ is the number of topical tweets by user u, and it stems from integrating out user's Bernoulli distribution π_u. $n_{u,(.)}^{(\pi)}$ is the total number of tweets by user u. Similarly, $n_{u,a}^{(\theta)}$ is the number of tweets assigned to topic a for this user, resulting from integrating out user's topic distribution θ_u. $n_{u,(.)}^{(\theta)}$ is the same as $n_{u,0}^{(\pi)}$. $E_{(v)}$ is the number of times word type v appears in the current tweet, and $E_{(.)}$ is the total number of words in the current tweet. $n_{a,v}^{(\phi)}$ is the number of times word type v is assigned to topic a, and $n_{a,(.)}^{(\phi)}$ is the number of words assigned to topic a. Note that we calculate all these counting matrixes without considering the current tweet.

Event-related:

Then for $y_{t,i} = 1$ and $s_{t,i} = k$, we use the following formula:

$$\begin{aligned} p(y_{t,i} = 1, s_{t,i} = k | \mathbf{y}_{\neg(t,i)}, \mathbf{s}_{\neg(t,i)}, \mathbf{w}) \propto \frac{n_{u,1}^{(\pi)} + \tau}{n_{u,(.)}^{(\pi)} + 2\tau} \\ \cdot N_{k,t} \cdot O_{k,t} \cdot \eta_{k,t} \cdot \frac{\prod_{v=1}^{V} \prod_{l=0}^{E_{(v)}-1} (n_{k,v}^{(\psi)} + l + \beta)}{\prod_{l=0}^{E_{(.)}-1} (n_{k,(.)}^{(\psi)} + l + V\beta)} \end{aligned}$$

where $n_{u,1}^{(\pi)}$ is the number of event-related tweets by user u, $n_{k,v}^{(\psi)}$ is the number of times word type v is assigned to event k, and $n_{k,(.)}^{(\psi)}$ is the total number of words assigned to event k. These word counters stem form integrating out each event's word distribution, and are set to zero when k is a new born event.

In Table 4.2, we show the values of $N_{k,t}$ $O_{k,t}$ and $\eta_{k,t}$ under various conditions.

These conditions are based on the temporal relation between the current tweet and the candidate event k. Here $n_{k,t}$ is the number of tweets in epoch t assigned to event k, excluding the current tweet. $\Delta^{(i)}_{k,t}$ is as we defined before, i_k is the index of the tweet that started event k in epoch t_k, and $n_{k,(.)} = \sum_{t'=\bar{t}_k}^{T} n_{k,t'}$. To simplify the formula, we use $\mathfrak{S}(t,i)$ to represent $\sum_{k'} \Delta^{(i)}_{k',t} + \alpha$, which reflects the probability to start a new table for the i-th document in epoch t.

Roughly speaking, $N_{k,t}$ contains two factors: (1) The size of event k around epoch t. (2) The time difference between the current time stamp t and the event's start time $\bar{t}_k$. $O_{k,t}$ considers the effect of replacing the cluster starter (the i_k-th tweet in epoch $\bar{t}_k$) with the current tweet. Finally, $\eta_{k,t}$ considers how the current event assignment affects the events which emerge later than the current tweet. In particular, in the condition when $\bar{t}_k$ equals $t+1$, assigning the current tweet to event k will bring the start date $\bar{t}_k$ forward, and $\mathfrak{S}(\bar{t}_{k'}, i_{k'})^{(t,i)}$ is calculated[2] after setting $\bar{t}_k$ to t.

Approximation:

Given the exact conditional probabilities as the previous formulas show, we opt to approximate the formulas by ignoring the factor $\eta_{k,t}$. We omit this influence factor because we find that it has a minor effect on the probability mass but largely increases the computational complexity. After using such approximation, the complexity of our model is similar to a degenerate variation of our model (one of the baselines in section 4.3.2), in which d-RCRP is replaced with RCRP. The differences are: (1) when sampling the i-th tweet at epoch t, our model need to record and track the latent event variables of previous tweets in the same epoch to calculate $\mathfrak{S}(t,i)$; (2) when the i-th tweet at epoch t starts an event during the previous iteration, we need to search for the nearest tweet which belongs to the same event to start the event. Although we use this approximated formula for the exact conditional

[2] $\mathfrak{S}(\bar{t}_{k'}, i_{k'})^{(t,i)}$ will be affected because of the factor $\Delta^{(i_{k'})}_{k,\bar{t}_{k'}}$. Since the start time $\bar{t}_k$ is changed from $t+1$ to t, the value of $\Delta^{(i_{k'})}_{k,\bar{t}_{k'}}$ should be updated to $(n_{k,\bar{t}_{k'}-1} + n_{k,\bar{t}_{k'}})(1 - e^{-\lambda(\bar{t}_{k'}-t)})$.

probabilities, we find that in our experiments the formula works fine and generates meaningful results.

4.3 Experiment

4.3.1 Dataset

We use a Twitter dataset that was previously used in [15] for finding bursty topics. The original dataset contains the tweets published by a large number of Singapore Twitter users. Since the entire dataset is huge, we pick 500 users, including 13 news media users, 2 journalists and 485 random users. We use their tweets between April 1 and June 30, 2012 for our experiments. We use the CMU Twitter POS Tagger[3] to tag these tweets and remove the non-standard words (i.e. words tagged as punctuation marks, emoticons, urls, at-mentions, pronouns, etc.) and stop words. Tweets with less than three words are also discarded. In the end we get 701,878 tweets in total.

4.3.2 Quantitative Evaluation

In the experiments below, we refer to our own model as d-RCRP. We quantitatively evaluate d-RCRP by comparing it with two baseline models:

RCRP: This is a modified version of our own model where we remove the duration-based probability discount, i.e. we use the standard RCRP. Comparison with this model helps us understand the effect of the duration-based discount.

TimeUserLDA: This model is from [15]. Similar to d-RCRP, TimeUserLDA also separates personal topical tweets from event-related tweets. However, it groups the event-related tweets into a *fixed* number of bursty topics and then uses a two-state machine in a postprocessing step to identify events from these bursty topics, i.e. events are not directly modeled within the generative process itself. In contrast,

[3] http://www.ark.cs.cmu.edu/TweetNLP/

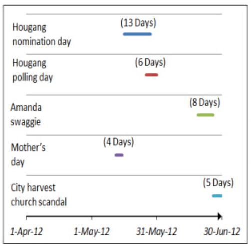
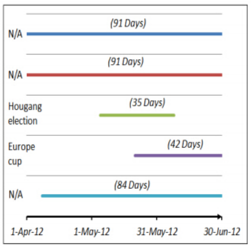
Events	Top words	Life cycle	Events	Top words	Life cycle
Hougang nomination day	#hougangbyelection, hougang, wp, desmond, png		N/A (Malay)	yg, di, yang, aku, dan	
Hougang polling day	#hougangbyelection, hougang, pap, png, desmond		N/A	singapore, prices, oil, asian, stocks	
Amanda swaggie	singapore, amanda, bieber, europe, trending		Hougang election	#hougangbyelection, hougang, wp, desmond, pap	
Mother's day	day, happy, mother's, mothers, love, mom		Europe cup	#euro2012, spain, portugal, euro, germany, italy	
City harvest church scandal	city, harvest, church, kong, founder		N/A	news, home, usa, run, blog	

Figure 4.3: Top five events detected by d-RCRP (left) and RCRP (right). We show each event's name (manually given and N/A indicates a meaningless event), top ranked words, and life cycle (the duration of the event).

d-RCRP and RCRP directly models events.

It is worth mentioning that both baselines separate topical tweets and event related tweets. We do not compare with the model in [3] or [46] because these methods are designed for news-centric data and treat all documents as event-related. The results of both model are poor as seen from Table 4.1 in Section 4.1.

For the parameter settings, we empirically set A to 80, γ to $\frac{50}{A}$, β to 0.01, τ to 1, and α to 1. The duration-based discount parameter λ is set to 1. We run 300 iterations before we collect 10 samples with a gap of 5 iterations to obtain our final latent variable assignment.

Event Quality

We first analyze the quality of the detected events. For each method, we rank the detected events based on the number of tweets assigned to them and then pick the top-30 events for each method. We randomly mix these events and ask two human judges to label them. For each event, the judges are provided with 100 randomly selected tweets (or all tweets if an event contains less than 100 tweets) together with their time stamps. The judges are allowed to use external sources to help them. An event is scored 1 if the 100 tweets coherently describe an event or 0 otherwise. The inter-annotator agreement score is 0.639 using Cohen's kappa. The final score of an event is 1 if both judges have scored it 1. Table 4.2 shows the performance in terms of Precision@K, and Table 4.3 shows the top five events detected by d-RCRP and RCRP respectively. The results show that our model outperforms the others consistently.

Method	**P@5**	**P@10**	**P@20**	**P@30**
d-RCRP	**1.000**	**1.000**	**1.000**	**0.800**
RCRP	0.400	0.500	0.600	0.600
TimeUserLDA	**1.000**	0.900	0.800	0.667

Table 4.2: Precision@K for the various models.

A close examination of the events reveals that RCRP identifies several events that are longstanding general topics (as Table 4.3 shows), which verifies that burstiness is an important factor to consider for identifying events on Twitter. It is interesting to see that TimeUserLDA outperforms RCRP. We believe that it is because TimeUserLDA also considers burstiness. However, TimeUserLDA requires a post-processing step whereas d-RCRP achieves event identification inside the generative model itself.

Tweet Quality

Event	**d-RCRP**	**RCRP**	**TimeUserLDA**
Amanda swaggie	**0.91**	0.88	0.79
Mother's day	**0.86**	0.82	0.77
April fools	0.85	0.85	**0.97**
City harvest church scandal	**0.86**	0.85	0.82
Father's day	**0.86**	0.77	0.65

Table 4.3: Precision of tweets for the 5 common events.

The evaluation above is at event-level. We also want to evaluate the relevance of the tweets assigned to each event. To make fair comparison, we select common events identified by all three methods. We further ask two human judges to score the 100 tweets as either 1 or 0 based on their relevance to the event. We obtain a Cohen's kappa of 0.760, which shows high agreement. Table 4.3 shows the precision of the tweets for all 5 common events. We find that for 4 of them, our model obtains the highest precision. The false positive tweets by RCRP are mislabeled mainly because the duration of the event tends to be long. For example, several tweets about Labor Day are clustered into the event of Mother's Day. The false positive tweets by TimeUserLDA are the ones with related words. For example, several "happy birthday" tweets are clustered into the event of Father's Day. For April Fools, after we take a close look at the corresponding tweets, we find that our model does not outperform other models mainly because most tweets of this event adopt similar words, such as ”aprilfool”, ”fraud” and ”prank”, which are quite distinctive and can separate the relevant tweets from other general tweets. Roughly speaking, TimeUserLDA performs well when the event is globally popular (i.e. festivals, or some major events) and the words of the event are distinctive.

Temporal Profile Quality

Besides the quality of the top-ranked events and their tweets, we also evaluate the temporal profiles of the events. Essentially the temporal profile of an event shows how the number of tweets related to an event changes over time. As it is hard for us to obtain the ground truth of the temporal profile of an event through human judgment we use hashtags to help us [26]. Twitter users create specific hashtags when significant events happen. These hashtags are widely used because of the diffusion effect on Twitter's huge network. We rank the hashtags in our data set based on their numbers of tweets. From the top-ranked ones, we pick 7 hashtags that are related to some meaningful events. We obtain a temporal profile of each of these hashtags based on the daily tweet counts. Our hypothesis is that this is close to the real temporal profile of the corresponding event. Then for each of the methods we consider, we pick the corresponding event for each hashtag and also obtain a temporal profile based on the daily tweet counts returned by that method. Finally, we convert the temporal profiles into distributions over time through normalization, and for each hashtag and each method, we compute the JS-divergence between the two distributions, one based on the hashtag and the other based on the method. We believe that the lower the JS-divergence is, the better the temporal profile of an event obtained by the method matches the ground truth. Table 4.4 shows the results. We can see that d-RCRP consistently gives lower JS-divergence scores than the other two methods except for #aprilfools. It shows that the tweets identified by d-RCRP for an event usually better reflect the real evolution of the event on Twitter.

HashTag/Event	d-RCRP	RCRP	TimeUserLDA
#ss4encore	**0.0282**	0.0448	N/A
#bigbangmonster	**0.0055**	0.1749	N/A
#ss4shanghai	**0.0004**	0.0738	N/A
#3years2ne1	**0.0344**	0.0616	N/A
#chc	**0.0419**	0.0465	0.1443
#aprilfools	0.0797	0.0882	**0.0656**
#getwellsoongaga	**0.1416**	0.2178	N/A

Table 4.4: The JS-divergence scores of the three methods. N/A means there is no corresponding event.

4.3.3 Qualitative Evaluation

In this section, we show some example results from our experiments that illustrate the advantages of our proposed model. Moreover, we can do various event-centric analyses (i.e. users' tendency to tweet about events, event-topic correlation), because our unified model considers both personal interests and events on Twitter. These analyses help better interpret events in Twitter.

Case Study

Events	Start date	Top words
candidate announcement	10 May	hougang, choo, desmond, png, candidate
nomination day	16 May	#hougangbyelection, hougang, wp, desmond, png,
polling day	26 May	#hougangbyelection hougang, pap, png, desmond,

Table 4.5: Case study on Hougang by-election.

For events that span a relatively long duration, our model tends to identify the most significant sub-events and treat these sub-events as events. For example, our data set covers the Singapore Hougang by-election, which lasted for around twenty days. There were three major events during this period: Election candidates were announced on May 10, the nomination day was on May 16, and the polling day was on May 26. Table 4.5 shows that our model correctly finds these major sub-events.

User Analysis

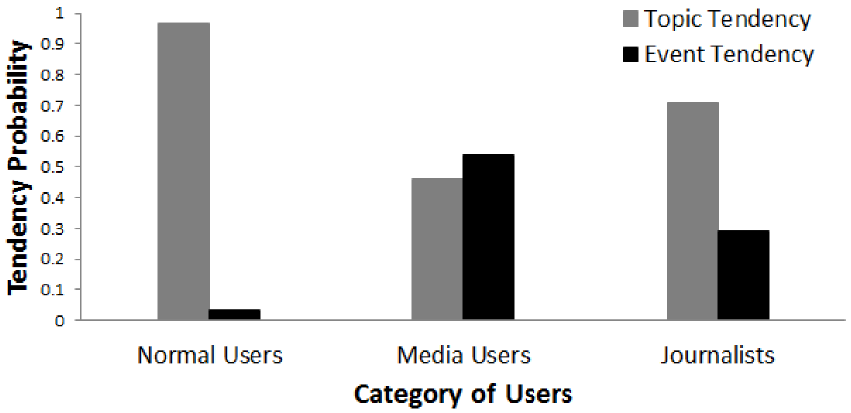


Figure 4.4: User's tendency to tweet on topics or events.

Since our model learns a user's tendency to tweet about topics or events, we can compare such tendency of normal users, media users (e.g. YahooNews) and journalists in our data. For each category of users, we average their Bernoulli parameter π_u and show the results in Figure 4.4. We can clearly see that media users are more likely to tweet about events compared with normal users. It may appear strange that media users also have a high probability to tweet about topics. This is because many news events tweeted by media users do not attract much attention on Twitter, and therefore these news events are not identified as popular events on Twitter but become general topics by our model. We also find that journalists' tendency to tweet about events lies between normal users and media users, which makes sense because journalists play dual roles as both a normal user and a media user.

Event-Topic Correlation Analysis

Our model does not directly model the correlation between events and topics. However, we expect that some events are more related to certain topics than others and therefore more likely to be tweeted by users interested in those topics. E.g. a Korean pop music concert is more related to the general topic on music or entertainment while the event on Eurocup is more related to the topic on sports. We can find such correlations through the following postprocessing. First, we average the topic distributions of all normal users to obtain a background topic distribution of our data. Denote this as θ_{B}. Then for each event, we obtain all users who have tweeted about the event and average these users' topic distributions. We thus obtain a topic distribution θ_k for each event k. By measuring the JS-divergence between θ_{B} and θ_k, we can rank the events.

We show 19 events tweeted by at least 20 users in increasing order of the JS-divergence scores in Table 4.6. We can see that the top-ranked events (with low JS-divergence) are those that tend to be tweeted by all users, while the low-ranked ones (with high JS-divergence) are those that are more related to certain topics than others and therefore tend to be tweeted by a subgroup of users. E.g. event 19 is about Super Junior, a Korean idol group, and this event is likely to be only interesting to K-pop fans. By analyzing the correlation between events and topics, we can potentially recommend relevant events to a user based on her topic interests.

4.4 Conclusions

In this chapter, we study the problem of event identification from Twitter stream. The Recurrent Chinese Restaurant Process is appealing for our task because it provides a principled dynamic non-parametric model. However, our preliminary experiment shows that RCRP is not directly applicable in our task for two reasons: (1) events emerge and die out fast on Twitter, (2) most tweets are topical and only a

Rank	Event Name	Score
1	Mother’s day	0.0068
2	Father’s day	0.0073
3	Indonesia tsunami	0.0106
4	April fool’s day	0.0113
5	Tsunami hit Singapore	0.0114
6	Alex push old lady	0.0162
7	Amanda swaggie	0.0170
8	Ferrari accident	0.0198
9	City harvest church scandal	0.0231
10	Staraward(Rui En)	0.0259
11	Hougang election polling day	0.0263
12	Hougang election nomination day	0.0263
13	Bigbang concert ticket sell	0.0320
14	Bigbang album “Monster”	0.0328
15	Euro cup 2012	0.0341
16	Mozambique fashion week	0.0354
17	Staraward(Jay Park)	0.0362
18	LionsXII 9-0 Sabah FA	0.0543
19	Super Junior new album	0.0805

Table 4.6: Events ranked based on JS-divergence.

small proportion of them are event-related. Therefore, we propose a novel duration-based probability discount to RCRP to capture the burstiness character of events on Twitter. We then propose a probabilistic model to identify both events and topics simultaneously from Twitter. Our experiments demonstrate that our proposed model can identify events accurately, which shows the effectiveness of duration-based discount. Finally, we qualitatively show some interesting studies on users and event-topic correlations.

Part II

Unified Analysis for Topics, Events and Users on Twitter

Chapter 5

A unified model of topics, events and users on Twitter

In this chapter, we try to model topics, events and users on Twitter in a unified way. We propose a model which combines an LDA-like topic model and the Recurrent Chinese Restaurant Process to capture topics and events. We further propose a duration-based regularization component to find bursty events. We also propose to use event-topic affinity vectors to model the association between events and topics. Our experiments shows that our model can accurately identify meaningful events and the event-topic affinity vectors are effective for event recommendation and grouping events by topics.

5.1 Introduction

In this chapter, we consider two concepts that have been repeatedly visited: (1) **Topics**. These are longstanding themes that many personal tweets revolve around. Example topics range from music and sports to more serious ones like politics and religion. Much work has been done to analyze topics on Twitter [42, 25, 56, 30]. (2) **Events**. These are the same as we have discussed in previous chapters.

The concepts of topics and events are orthogonal in that many events fall under certain topics. For example, concerts fall under the topic about music. Furthermore, being *social* media, Twitter users play important roles in forming topics and events on Twitter. Each user has her own topic interests, which influence the content of her tweets. Whether a user publishes a tweet related to an event also largely depends on whether her topic interests match the nature of the event. Modeling the interplay between topics, events and users can deepen our understanding of Twitter content and potentially aid many prediction and recommendation tasks. In this chapter, we aim to construct a unified model of topics, events and users on Twitter. Although there has been a number of recent studies on event detection on Twitter, to the best of our knowledge, ours is the first that links the topic interests of users to their tweeting behaviors on events.

Specifically, we propose a probabilistic latent variable model that identifies both topics and events on Twitter. To do so, we first separate tweets into *topic tweets* and *event tweets*. The former are related to a user's personal life, such as a tweet complaining about the traffic condition or wishing a friend happy birthday. The latter are about some major global event interesting to a large group of people, such as a tweet advertising a concert or commenting on an election result. Although considering only topic tweets and event tweets is a much simplified view of the diverse range of tweets, we find it effective in finding meaningful topics and events. We further use an LDA-like model [13] to discover topics and the Recurrent Chinese Restaurant Process [5] to discover events. Details are given in Section 5.2.1.

Our major contributions lie in two novel modifications to the base model described above. The first is a duration-based regularization component that punishes long-term events (Section 5.2.2). Because events on Twitter tend to be bursty, this modification presumably can produce more meaningful events. More specifically, we borrow the idea of using pseudo-observed variables to regularize graphical models [8], and carefully design the pseudo-observed variable in our task to capture the burstiness of events. The second modification is adding event-topic affinity vectors inspired by PMF-based collaborative filtering [45] (Section 5.2.3). It uses the latent

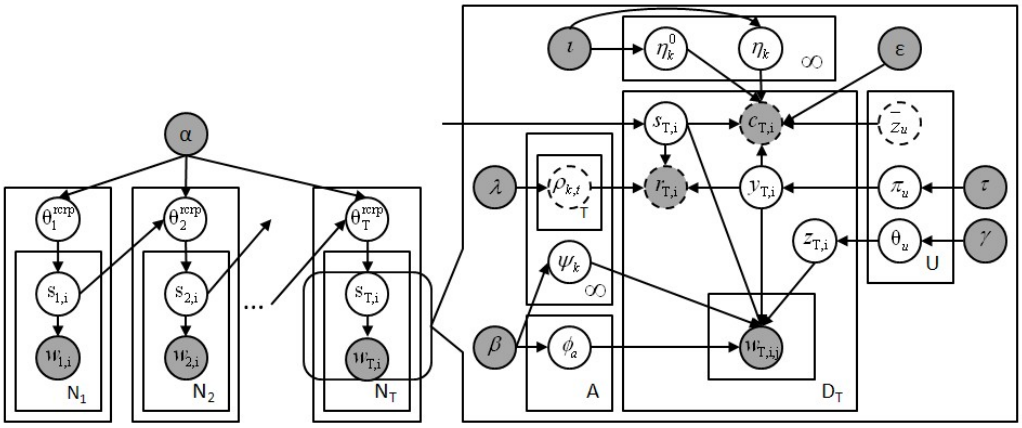


Figure 5.1: Plate notation for the whole model, in which pseudo-observed variables and distributions based on empirical counts are shown as dotted nodes.

topics to explain users' preferences of events and subsequently infers the association between topics and events.

We use a real Twitter data set consisting of 500 users to evaluate our model (Section 5.3). We find that the model can discover meaningful topics and events. Comparison with our base model and with an existing model for event discovery on Twitter shows that the two modifications are both effective. The duration-based regularization helps find more meaningful events; the event-topic affinity vectors improve an event recommendation task and helps produce a meaningful organization of events by topics.

5.2 Our Model

In this section, we present our model for topics, events and users on Twitter. We assume that we have a stream of tweets which are divided into T epoches. Let $t \in \{1, 2, \ldots, T\}$ be the index of an epoch. Each epoch contains a set of tweets and each tweet is a bag of words. We use $w_{t,i,j} \in \{1, 2, \ldots, V\}$ to denote the j-th word of the i-th tweet in the t-th epoch, where V is the vocabulary size. The author of the i-th tweet in the t-th epoch (i.e. the Twitter user who publishes the tweet) is denoted as $u_{t,i} \in \{1, 2, \ldots, U\}$, where U is the total number of Twitter users we consider.

We first present our base model in Section 5.2.1. We then introduce a duration-based regularization mechanism to ensure the burstiness of events in Section 5.2.2. In Section 5.2.3 we discuss how we model the relation between topics and events using event-topic affinity vectors. Finally we discuss model inference in Section 5.2.4.

5.2.1 The Base Model

Recall that our objective is to model topics, events, users and their relations. As in many topic models, our topic is a multinomial distribution over words, denoted as ϕ_a where a is a topic index. Each event is also a multinomial distribution over words, denoted as ψ_k where k is an event index. Because topics are long-standing and stable, we fix the number of topics to be A, where A can be tuned based on historical data. In contrast, events emerge and die along the timeline. We therefore use a non-parametric model called the Recurrent Chinese Restaurant Process (RCRP) [5] to model the birth and death of events. To model the relation between users and topics, we assume each user u has a multinomial distribution over topics, denoted as θ_u.

As we have discussed, we separate tweets into two categories, topic tweets and event tweets. Separation of these two categories is done through a latent variable y sampled from a user-specific Bernoulli distribution π_u. For topic tweets, the topic is sampled from the corresponding user's topic distribution θ_u. For event tweets, the event is sampled according to RCRP. We now briefly review RCRP. Generally speaking, RCRP assumes a Chinese Restaurant Process (CRP) [10] for items within an epoch and chains up the CRPs in adjacent epochs along the timeline. Specifically, in our case, the generative process can be described as follows. Tweets come in according to their timestamps. In the t-th epoch, for the i-th tweet, we first flip a biased coin based on probability π_u to decide whether this tweet is event-related. If it is, then we need to decide which event it belongs to. It could be an existing event that has at least one related tweet in the previous epoch or the current epoch, or it could be a new event. Let $n_{k,t-1}$ denote the number of tweets related to event k at the end of epoch $(t-1)$. Let $n_{k,t}^{(i)}$ denote the number of tweets related to event k in epoch t before the i-th tweet comes. Let N_{t-1} denote the total number of event-related tweets in epoch $(t-1)$ and $N_t^{(i)}$ denote the number of event-related tweets in epoch t before the i-th tweet. Then RCRP assumes that the probability for the i-th tweet to join event k is $\frac{n_{k,t-1}+n_{k,t}^{(i)}}{N_{t-1}+N_t^{(i)}+\alpha}$ and the probability to start a new event is $\frac{\alpha}{N_{t-1}+N_t^{(i)}+\alpha}$, where α is a parameter. As we can see, RCRP naturally captures the "rich-get-richer" phenomenon in social media.

Finally we place Dirichlet and Beta priors on the various parameters in our model. Formally, the generative process of our base model is outlined in Figure 5.2, excluding the lines in bold and blue. We also show the plate notation in Figure 5.1, in which the Recurrent Chinese Restaurant Process is represented as an infinite dynamic mixture model [5] and θ_t^{rcrp} means the distribution on an infinite number of events in epoch t. D_t is the total number of tweets (both event-related and topic tweets), while N_t represents the number event-related tweets in epoch t.

5.2.2 Regularization on Event Durations

As we have pointed out, events on Twitter tend to be bursty, i.e. the duration of an event tends to be short, but this characteristic is not captured by RCRP. While there can be different ways to incorporate this intuition, here we adopt the idea of regularization using pseudo-observed variables proposed recently by [8]. We introduce a pseudo-observed binary variable $r_{t,i}$ for each tweet, where the value of $r_{t,i}$ is set to 1 for all tweets. We assume that this variable is dependent on the hidden variables y and s. Specifically, if $y_{t,i}$ is 0, i.e. the tweet is topic-related, then $r_{t,i}$ gets a value of 1 with probability 1. If $y_{t,i}$ is 1, then we look at all the tweets that belong to event $s_{t,i}$. Our goal is to make sure that this tweet is temporally close to these other tweets. So we assume that $r_{t,i}$ gets a value of 1 with probability $\exp(-\sum_{t'=1,|t'-t|>1}^{T} \lambda|t-t'|n_{s_{t,i},t'})$, where $n_{s_{t,i},t'}$ is the number of tweets in epoch t' that belong to event $s_{t,i}$ and $\lambda > 0$ is a parameter. We can see that when we factor

```
1. For each topic $a = 1, \ldots, A$
    (a) draw $\phi_a \sim \mathrm{Dirichlet}(\beta)$
2. For each user $u = 1, \ldots, U$
    (a) draw $\theta_u \sim \mathrm{Dirichlet}(\gamma), \pi_u \sim \mathrm{Beta}(\tau)$
3. For each epoch $t$ and tweet $i$
    (a) draw $y_{t,i} \sim \mathrm{Bernoulli}(\pi_{u_{t,i}})$
    (b) If $y_{t,i} = 0$
        i. draw $z_{t,i} \sim \mathrm{Multinomial}(\theta_{u_{t,i}})$
        ii. For each $j$, draw $w_{t,i,j} \sim \mathrm{Multinomial}(\phi_{z_{t,i}})$
    (c) If $y_{t,i} = 1$
        i. draw $s_{t,i}$ from RCRP
        ii. If $s_{t,i}$ is a new event
            A. draw $\psi_{s_{t,i}} \sim \mathrm{Dirichlet}(\beta)$
            B. draw $\eta^0_{s_{t,i}} \sim$ Gaussian$(0, \iota^{-1})$
            C. draw $\boldsymbol{\eta}_{s_{t,i}} \sim$ Gaussian$(0, \iota^{-1} I_A)$
        iii. draw $r_{t,i} \sim$ Bernoulli$(\rho_{s_{t,i},t})$, where $\rho_{s_{t,i},t} = \exp(-\sum_{t'=1,|t'-t|>1}^{T} \lambda|t'-t|n_{s_{t,i},t'})$
        iv. draw $c_{t,i} \sim$ Gaussian$(\eta^0_{s_{t,i}} + \boldsymbol{\eta}^T_{s_{t,i}} \cdot \bar{\boldsymbol{z}}_{u_{t,i}}, \epsilon^{-1})$
        v. For each $j$, draw $w_{t,i,j} \sim \mathrm{Multinomial}(\psi_{s_{t,i}})$
```

Figure 5.2: The generative process of our model, in which the duration-based regularization (section 5.2.2) and the event-topic affinity vector (section 5.2.3) are in blue and bold lines.

in the generation of these pseud-observed variables r, we penalize long-term events and favor events whose tweets are concentrated along the timeline. Generation of these variables r is shown in bold and blue in Figure 5.2.

5.2.3 Event-Topic Affinity Vectors

So far in our model topics and events are not related. However, many events are highly related to certain topics. For example, a concert is related to music while a football match is related to sports. We would like to capture these relations between topics and events. One way to do it is to assume that event tweets also have topical words sampled from the event's topic distribution, something similar to the models in [3] and by [46]. However, our prelimiary experiments show that this idea does not work well on Twitter, mainly because tweets are too short. Here we explore another approach inspired by recommendation methods based on probabilistic matrix factorization [45]. The idea is that when a user posts a tweet about an event, we can treat the event as an item and this posting behavior as adoption of the item. If we assume that the adoption behavior is influenced by some latent factors, i.e. the latent topics, then basically we would like the topic distribution of this user to be close to that of the event.

Specifically, we assume that each event k has associated with it an A-dimensional vector $\boldsymbol{\eta}_k$ and a parameter η_k^0. The vector $\boldsymbol{\eta}_k$ represents the event's affinity to topics. η_k^0 is a bias term that represents the inner popularity of an event regardless of its affinity to any topic. We further assume that each tweet has another pseudo-observed variable $c_{t,i}$ that is set to 1. For topic tweets, $c_{t,i}$ gets a value of 1 with probability 1. For event tweets, $c_{t,i}$ is generated by a Gaussian distribution with mean equal to $\eta_{s_{t,i}}^0 + \boldsymbol{\eta}_{s_{t,i}} \cdot \bar{\boldsymbol{z}}_{u_{t,i}}$, where $\bar{\boldsymbol{z}}_u$ is an A-dimensional vector denoting the empirical topic distribution of user u's tweets. This treatment follows the practice of fLDA by [1]. Let $\bar{C}_{u,a}$ be the number of tweets by user u assigned to topic a, based on the values of the latent variables y and z. Then

$$\begin{aligned}\bar{\boldsymbol{z}}_{u,a} &= \frac{\bar{C}_{u,a}}{\sum_{a'=1}^{A} \bar{C}_{u,a'}},\\ c_{t,i} &\sim \text{Gaussian}(\eta_{s_{t,i}}^0 + \boldsymbol{\eta}_{s_{t,i}} \cdot \bar{\boldsymbol{z}}_{u_{t,i}}, \epsilon^{-1}),\end{aligned}$$

where ϵ is a parameter. We generate $\boldsymbol{\eta}_k$ and η_k^0 using Gaussian priors once event k emerges. The generation of the variables c is shown in bold and blue in Figure 5.2.

5.2.4 Inference

We train the model using a stochastic EM sampling scheme. In this scheme, we alternate between Gibbs sampling and gradient descent. In the Gibbs sampling part, we fix the values of η_k^0 and $\boldsymbol{\eta_k}$ for each event k, and then we sample the latent variables $y_{t,i}$,$z_{t,i}$ and $s_{t,i}$ for each tweet. In the gradient descent part, we update the event-topic affinity vectors $\boldsymbol{\eta_k}$ and the bias term η_k^0 of each event k by keeping the assignment of the variables $y_{t,i}$,$z_{t,i}$ and $s_{t,i}$ fixed.

For the Gibbs sampling part, we jointly sample $y_{t,i} = 0, z_{t,i} = a$ (topic tweet) and $y_{t,i} = 1, s_{t,i} = k$ (event tweet) as follows:

Topic tweet:

$$
\begin{aligned}
&p(y_{t,i} = 0, z_{t,i} = a | \mathbf{y}_{\neg t,i}, \mathbf{z}_{\neg t,i}, \mathbf{w}, \mathbf{r}, \mathbf{c}, u_{t,i}) \\
\propto &\frac{n_{u,0}^{(\pi)} + \tau}{n_{u,(.)}^{(\pi)} + 2\tau} \frac{n_{u,a}^{(\theta)} + \gamma}{n_{u,(.)}^{(\theta)} + A\gamma} \frac{\prod_{v=1}^{V} \prod_{i=0}^{E_{(v)}-1} (n_{a,v}^{(\phi)} + i + \beta)}{\prod_{i=0}^{E_{(.)}-1} (n_{a,(.)}^{(\phi)} + i + V\beta)} \\
&\prod_{t',i' \in I_u} \frac{\mathcal{N}(c_{t',i'} | \eta_{s_{t',i'}}^0 + \boldsymbol{\eta}_{s_{t',i'}} \cdot \bar{\boldsymbol{z}}_u^*, \epsilon^{-1})}{\mathcal{N}(c_{t',i'} | \eta_{s_{t',i'}}^0 + \boldsymbol{\eta}_{s_{t',i'}} \cdot \bar{\boldsymbol{z}}_u, \epsilon^{-1})}
\end{aligned}
$$

Event tweet:

$$
\begin{aligned}
&p(y_{t,i} = 1, s_{t,i} = k | \mathbf{y}_{\neg t,i}, \mathbf{z}_{\neg t,i}, \mathbf{w}, \mathbf{r}, \mathbf{c}, u_{t,i}) \\
\propto &\frac{n_{u,1}^{(\pi)} + \tau}{n_{u,(.)}^{(\pi)} + 2\tau} \frac{1}{\boldsymbol{N}} \Bigg(n_{k,t}^{\text{RCRP}} \mathcal{N}(c_{t,i} | \eta_{s_{t,i}}^0 + \boldsymbol{\eta}_{s_{t,i}} \cdot \bar{\boldsymbol{z}}_u, \epsilon^{-1}) \\
&\cdot \exp(- \sum_{\substack{t'=1 \\ |t'-t|>1}}^{T} \lambda |t - t'| n_{k,t'}) \Bigg) \frac{\prod_{v=1}^{V} \prod_{i=0}^{E_{(v)}-1} (n_{k,v}^{(\psi)} + i + \beta)}{\prod_{i=0}^{E_{(.)}-1} (n_{k,(.)}^{(\psi)} + i + V\beta)}
\end{aligned}
$$

in which,

$$n_{k,t}^{\text{RCRP}} = \begin{cases} (n_{k,t-1} + n_{k,t}) \\ \qquad \cdot \frac{n_{k,t}+n_{k,t+1}}{n_{k,t}} & \text{if } n_{k,t-1} > 0, n_{k,t} > 0, \\ n_{k,t-1} & \text{if } n_{k,t-1} > 0, n_{k,t} = 0, \\ n_{k,t+1} & \text{if } n_{k,t+1} > 0, n_{k,t} = 0, \\ \alpha & \text{if } k \text{ is a new event,} \end{cases}$$

where we use u to represent $u_{t,i}$. $n_{u,0}^{(\pi)}$ is the number of topic tweets by user u while $n_{u,1}^{(\pi)}$ is the number of event tweets by user u. They stem from integrating out the user's Bernoulli distribution π_u. $n_{u,(.)}^{(\pi)}$ is the total number of tweets by user u. Similarly, $n_{u,a}^{(\theta)}$ is the number of tweets assigned to topic a for this user, resulting from integrating out the user's topic distribution θ_u. $n_{u,(.)}^{(\theta)}$ is the same as $n_{u,0}^{(\pi)}$. $E_{(v)}$ is the number of times word type v appears in the current tweet, and $E_{(.)}$ is the total number of words in the current tweet. $n_{a,v}^{(\phi)}$ is the number of times word type v is assigned to topic a, and $n_{a,(.)}^{(\phi)}$ is the number of words assigned to topic a. $n_{k,v}^{(\psi)}$ is the number of times word type v is assigned to event k, and $n_{k,(.)}^{(\psi)}$ is the total number of words assigned to event k. These word counters stem form integrating out each event's word distribution and are set to zero when k is a new event. $I_u = \{t', i' | y_{t',i'} = 1, u_{t',i'} = u\}$, which is the set of event tweets published by user u, and u represents $u_{t,i}$ for short. $\bar{\boldsymbol{z}}_u^*$ is the empirical counting vector which considers the current tweet's topic assignment, while $\bar{\boldsymbol{z}}_u$ and all other counters do not consider the current tweet. Finally, $\boldsymbol{N}$ is a local normalization factor for event tweets, which includes the RCRP, event-topic affinity and regularization on event duration.

With the previous Gibbs sampling step, we can get the assignment of variables $y_{t,i}$,$z_{t,i}$ and $s_{t,i}$. Given the assignment, we use gradient descent to update the values of the bias term η_k^0 and the event-topic affinity vectors $\boldsymbol{\eta}_k$ for each current existing event k. First, we can get the logarithm of the posterior distribution:

$$
\begin{aligned}
&\ln P(\mathbf{y}, \mathbf{z}, \mathbf{s}, \mathbf{r}, \mathbf{c} | \mathbf{w}, \mathbf{u}, \text{all priors}) \\
&= \text{constant} - \sum_{k=1}^{\infty} \{ \frac{\iota}{2} (\eta_k^{0^2} + \boldsymbol{\eta}_k \cdot \boldsymbol{\eta}_k) \\
&+ \sum_{u=1}^{U} n_{u,k} \frac{\epsilon}{2} [1 - (\eta_k^0 + \boldsymbol{\eta}_k \cdot \bar{\boldsymbol{z}}_u)]^2 \},
\end{aligned}
$$

where $n_{u,k}$ is the number of event tweets about event k published by user u. The derivative of the logarithm of the posterior distribution with respect to the bias term η_k^0 and the event-topic affinity vector $\boldsymbol{\eta}_k$ are as follows:

$$
\begin{aligned}
\frac{\partial \ln P}{\partial \eta_k^0} &= -\iota \eta_k^0 + \sum_{u=1}^{U} \epsilon n_{u,k} [1 - (\eta_k^0 + \boldsymbol{\eta}_k \cdot \bar{\boldsymbol{z}}_u)], \\
\frac{\partial \ln P}{\partial \boldsymbol{\eta}_k} &= -\iota \boldsymbol{\eta}_k + \sum_{u=1}^{U} \epsilon n_{u,k} [1 - (\eta_k^0 + \boldsymbol{\eta}_k \cdot \bar{\boldsymbol{z}}_u)] \bar{\boldsymbol{z}}_u.
\end{aligned}
$$

5.3 Experiment

5.3.1 Dataset and Experiment Setup

We evaluate our model on a Twitter dataset that contains 500 users. These users are randomly selected from a much larger pool of around 150K users based in Singapore. Selecting users from the same country/city ensures that we find coherent and meaningful topics and events. We use tweets published between April 1 and June 30, 2012 for our experiments. For preprocessing, we use the CMU Twitter POS Tagger[1] to tag these tweets and remove those non-standard words (i.e. words tagged as punctuation marks, emoticons, urls, at-mentions, pronouns, etc.) and stop words. We also remove tweets with less than three words. After preprocessing, the dataset contains 655,881 tweets in total.

Recall that our model is designed to identify topics, events and their relations

[1] `http://www.ark.cs.cmu.edu/TweetNLP/`

Event	Top words	Duration	Inner popularity (η_k^0)
debate caused by Manda Swaggie	singapore, bieber, europe, amanda, justin, trending, manda, hates, swaggie, hate	17 June - 19 June	0.9457
Indonesia tsunami	tsunami, earthquake, indonesia, singapore, hit, warning, aceh, 8.9, safe, magnitude	10 April - 12 April	0.9439
SJ encore concert	#ss4encore, cr, #ss4encoreday2, hyuk, 120526, super, leader, changmin, fans, teuk	26 May - 28 May	0.8360
Mother's Day	day, happy, mother's, mothers, love, mom, mum, everyday, mother, moms	11 May - 14 May	0.9370
April Fools' Day	april, fools, day, fool, joke, prank, happy, today, trans, fool's	1 April - 3 April	0.9322

Table 5.1: The top-5 events identified by Base+Reg+Aff. We show the story name which is manually labeled, top ten ranking words, lasting duration and the inner popularity (η_k^0) for each event.

with users. We therefore would like to evaluate the quality of the identified topics and events as well as the usefulness of the discovered topic distributions of users and event-topic affinity vectors. Because our topic discovery mechanism is fairly standard and a quick inspection shows that the discovered topics are generally meaningful and comparable to those discovered by standard LDA, here we do not focus on evaluation of topics. In Section 5.3.2 we evaluate the quality of the discovered events. In Section 5.3.3 we show how the discovered event-topic affinity vectors can be useful.

For comparison, we consider an existing method called **TimeUserLDA** introduced in our previous work [15]. TimeUserLDA also models topics and events by separating topic tweets from event tweets. However, it groups event tweets into a *fixed* number of *bursty topics* and then uses a two-state machine in a postprocessing step to identify events from these bursty topics. Thus, events are not directly modeled within the generative process itself. In contrast, events are inherent in our generative model. We do not compare with other event detection methods because our objective is not online event detection.

We also compare our final model with two degenerate versions of it. We refer to the base model described in Section 5.2.1 as **Base** and the model with the duration-based regularization as **Base+Reg**. Comparison with these two degenerate models allows us to assess the effect of the two modifications we propose. We refer to the final model with both the duration-based regularization and the event-topic affinity vectors as **Base+Reg+Aff**.

For the parameter setting, we empirically set A to 40, γ to $\frac{50}{A}$, τ to 1, β to 0.01, α to 1, ι to 10, ϵ to 1, and the duration regularization parameter λ to 0.01. When a new event k is created, the inner popularity bias term η_k^0 is set to 1, and the factors in event-topic affinity vectors $\boldsymbol{\eta}_k$ are all set to 0. We run the stochastic EM sampling scheme for 300 iterations. After Gibbs sampling assigns each variable a value at the end of each iteration, we update the values of η_k^0 and $\boldsymbol{\eta}_k$ for the existing events using gradient descent.

5.3.2 Events

First we quantitatively evaluate the quality of the detected events. Our model finds clusters of tweets that represent events. We first assess whether these events are meaningful. We then judge whether the detected event tweets are indeed related to the corresponding event.

Quality of Top Events

Method	**P@5**	**P@10**	**P@20**	**P@30**
Base+Reg+Aff	**1.000**	**1.000**	**0.950**	**0.900**
Base+Reg	**1.000**	**1.000**	**0.950**	0.867
Base	0.000	0.200	0.250	0.367
TimeUserLDA	**1.000**	0.800	0.750	0.600

Table 5.2: Precision@K for the various methods.

Usually we are interested in the most popular events on Twitter. We therefore assess whether the top events are meaningful. For each method, we rank the detected events based on the number of tweets assigned to them and then pick the top-30 events for each method. We randomly mix these events and ask two human judges to label them. The judges are given 100 randomly selected tweets for each event (or all tweets if an event contains less than 100 tweets). The judges can use external sources to help them. If an event is meaningful based on the 100 sample tweets, a score of 1 is given. Otherwise it is scored 0. The inter-annotator agreement score is 0.744 using Cohen's kappa, showing substantial agreement. Finally we treat an event as meaningful if both judges have scored it 1.

Table 5.2 shows the performance in terms of precision@K, and Table 5.1 shows the top 5 events of our model (i.e., Base+Reg+Aff). We have the following findings from the results: (1) Our base model performs quite poorly for the top events while Base+Reg and Base+Reg+Aff perform much better. This shows that the duration-based regularization is critical in finding meaningful events. A close examination shows that the base model clusters many general topic tweets as events, such as tweets about transportation and music and even foursquare tweets. (2) TimeUserLDA performs well for the very top events (P@5 and P@10) but its performance drops for lower-ranked events (P@20 and P@30), similar to what was reported by [15]. A close examination shows that this method is good at finding major events that do not have strong topic association and thus attract most people's attention, e.g. earthquakes, but not good at finding topic-oriented events such as some concerts and sports games. This is because this method mixes topics and events first and only detects events from bursty topics in a second stage of post-processing. In contrast, our model performs well for topic-oriented events. (3) The difference between Base+Reg and Base+Reg+Aff is small, suggesting that the event-topic affinity vectors are not crucial for event detection.

Precision of Event Tweets

Next, we evaluate the relevance of the detected event tweets to each event. To make a fair comparison, we select only the common events identified by all the methods. We pick 3 out of 5 common events shared by all methods within top-30 events (we pick "Fathers' day" to represent public festivals, and ignore the similar events "Mothers' day" and "April fools"). We also pick one event shared by three RCRP

Event	**TimeUserLDA**	**Base**	**Base+Reg**	**Base+Reg+Aff**
Father's Day	0.61	0.63	0.71	**0.72**
debate caused by Manda Swaggie	0.73	0.74	**0.84**	0.80
Indonesia tsunami	0.75	0.75	**0.82**	0.80
Super Junior album release	N/A	0.72	0.78	**0.81**

Table 5.3: Precision of the event tweets for the 4 common events.

based models. We further ask one of the judges to score the 100 tweets as either 1 or 0 based on their relevance to the event. The precision of the 100 tweets for each event and each method is shown in Table 5.3. We can see that again Base+Ref and Base+Ref+Aff perform similarly, and both outperform the other two methods. We also take a close look at the tweets and find that the false positives mislabeled by Base is mainly due to the long-duration of the discovered events. For example, for the event "Super Junior album release," Base finds other music-related tweets surrounding the peak period of the event itself.

In summary, our evaluation on event quality shows that (1) Using the non-parametric RCRP model to identify events within the generative model itself is advantageous over TimeUserLDA, which identifies events by postprocessing. (2) The duration-based regularization is crucial for finding more meaningful events.

Comparison with Duration-based Discount Method

Recall that, in Chapter 4 we introduced a duration-based discount method to capture the burstiness of the events on Twitter. The disadvantage of the duration-based discount method is that the model now lacks exchangeability, which makes it hard for inference. In Chapter 4 we had to use some approximation to simplify the model inference. In this chapter, we use a duration-based regularization method (Base+Reg) to capture the burstiness of Twitter events, where the tweets within the same epoch are exchangeable. It is natural to compare these two different ways of modeling burstiness. In this section, we compare the two methods by evaluating the quality of the top-ranked events, using the data set mentioned in Section 4.3. Table 5.4 shows the precision@K for the two methods.

Method	**P@5**	**P@10**	**P@20**	**P@30**
d-RCRP	**1.000**	**1.000**	**1.000**	0.800
Base+Reg	**1.000**	**1.000**	**1.000**	**0.850**

Table 5.4: Precision@K for the two models.

As the result shows, the two models perform similarly when $K \leq 20$, which suggests that both duration-based discount and duration-based regularization are helpful to capture the significant events. However, when K equals 30, the model with duration-based regularization (Base+Reg) performs better. The main reason is that duration-based regularization forces the event-related clusters to peak at a certain timestamp, which may better match the nature of events on Twitter. However, duration-based discount only penalizes the event-related clusters with long duration. This comparison shows that duration-based regularization has both the advantage of allowing faster computation and the advantage of capturing better events.

5.3.3 Event-Topic Association

Besides event identification, our model also finds the association between events and topics through the event-topic affinity vectors. The discovered event-topic association can potentially be used for various tasks. Here we conduct two experiments to demonstrate its usefulness.

Event Recommendation

Recall that to discover event-topic association, we treat an event as an item and a tweet about the event as indication of the user's adoption of the item. Following this analogy with item recommendation, we define an event recommendation task where the goal is to recommend an event to users who have not posted any tweet about the event but may potentially be interested in the event. Intuitively, if a user's topic distribution is similar to the event-topic affinity vector of the event, then the user is likely to be interested in the event.

Event	TimeUserLDA	Base	Base+Reg	Base+Reg+Aff	Inner popularity (η_k^0)
debate caused by Manda Swaggie	0.3533	0.3230	**0.3622**	0.2956	0.943
Father's Day	0.3811	0.3525	0.3596	**0.4362**	0.917
Big Bang album release	0.1406	0.1854	0.1533	**0.1902**	0.893
City Harvest Church scandal	N/A	0.2832	0.1874	**0.3347**	0.890
Alex Ong pushing an old lady	N/A	**0.1540**	0.1539	0.1113	0.876
final episode of Super Spontan (reality show)	N/A	0.0177	0.0331	**0.2900**	0.862
Super Junior album release	N/A	0.0398	0.0330	**0.5900**	0.792
LionsXII 9-0 Sabah FA (soccer)	0.0711	0.1207	0.2385	**0.3220**	0.773
MAP	N/A	0.1845	0.1901	**0.3213**	

Table 5.5: For the 8 test events that happened in June 2012, we compute the Average Precision for each event. We also show the Mean Average Precision (MAP) when applicable.

Topic	Top words of the topic	Related event	Top words of the event
Food	eat, food, eating, ice, hungry, dinner, cream, lunch, chicken, buy	Ben&Jerry free cone day	free, cone, day, ben, jerry's, today, b&j, zoo, #freeconeday, singapore
Korean Music	music, big, cr, super, bang, junior, love, concert, bank, album	Super Junior encore concert	#ss4encore, cr, #ss4encoreday2, hyuk, 120526, super, leader, changmin, fans, teuk
		Super Junior Shanghai concert	#ss4shanghai, cr, 120414, donghae, eunhyuk, giraffe, solo, hyuk, ryeowook, shanghai
		Super Junior Paris concert	#ss4paris, cr, paris, super, 120406, ss4, junior, siwon, show, update
Malay	aku, nak, tak, kau, ni, lah, tk, je, mcm, nk	final episode of Super Spontan	zizan, johan, friendship, jozan, #superspontan, skips, forever, real, juara, gonna
		LionsXII 9-0 Sabah FA	sabah, 9-0, #lionsxii, lions, singapore, 7-0, amet, sucks, sabar, goal
Soccer	win, game, man, chelsea, match, city, goal, good, united, team		
		Man City crowned English champions	man, city, united, qpr, fuck, bored, lah, love, glory, update

Table 5.6: Example topics and their corresponding correlated events.

Specifically, we use the first two months' data (April and May 2012) as training data to learn all the users' topic distributions. We then use a ransom subset of 250 training users and their tweets in June to identify events in June as well as the event-topic affinity vectors of these events. We pick 8 meaningful events that are ranked high by all methods for testing. For each event, we try to find among the remaining 250 users those who may be interested in the event and compare the results with ground truth obtained by human judgment. Because it is time consuming to obtain the ground truth for all 250 users, we randomly pick 100 of these 250 users for testing purpose. For each test user and each event, we manually inspect the user's tweets around the peak days of the event to judge whether she has commented on the event. This is used as ground truth.

With our complete model Base+Reg+Aff, we can simply rank the 100 test users in decreasing order of $\boldsymbol{\eta}_k \cdot \bar{\boldsymbol{z}}_u$. For the other methods, because we do not have any parameter that directly encodes event-topic association, we cannot rank users based on how similar their topic distributions are to the event's affinity to topics. We instead adopt a collaborative filtering strategy and rank the test users by their similarity with those training users who have tweeted about the event. Specifically, each of these methods produces a topic distribution θ_u for each user. In addition, for each test event these methods identify a list of training users who have tweeted on it. By taking the average topic distribution of these training users and compute its cosine similarity with a test user's topic distribution, we can rank the 100 test users.

Since we have turned the recommendation task into a ranking task, we use Average Precision, a commonly used metric in information retrieval, to compare the performance. Average Precision is the average of the precision value obtained for the set of top items existing after each relevant item is retrieved [34]. We also rank the 8 events in decreasing order of their inner popularity η_k^0 learned by our complete model. The results are shown in Table 5.5. We have the following findings from the table. (1) Our complete method outperforms the other methods for 6 out of the 8 test events, suggesting that with the inferred event-topic affinity vectors we can do better event recommendation. (2) The improvement brought by the event-topic affinity vectors, as reflected in the difference in Average Precision between Base+Reg+Aff and Base (or Base+Reg) is more pronounced for events with lower inner popularity. Recall that the inner popularity of an event shows the inherent popularity of an event regardless of its association with any topic, that is, an event with high inner popularity attracts attention of many people regardless of their topic interests, while an event with low inner popularity tends to attract attention of certain people with similar topic interests. The finding above suggests that the event-topic affinity vectors are especially useful for recommending events that attract only certain people's attention, such as those related to sports, music, etc.

One may wonder for the events with low inner popularity why we could not achieve the same effect by Base or Base+Reg where we consider the topic similarity of test users with training users who have tweeted about the event. Our close examination shows that for these events although Base and Base+Reg may identify relevant event tweets with decent precision, the users they identify who have tweeted about the event may not share similar topic interests. As a result, when we average these users' topic interests, we cannot obtain a clear skewed topic distribution that explains the event's affinity to different topics. In contrast, Base+Reg+Aff explicitly models the event-topic affinity vector and prefers to assign a tweet to an event if its author's topic distribution is similar to the event's affinity vector. Through the training iterations, the users who have tweeted about an event as identified by Base+Reg+Aff will gradually converge to share similar topic distributions.

Grouping Events by Topics

Finally, we show that the event-topic affinity vectors can also be used to group events by topics. This can potentially be used to better organize and present popular events in social media. In Table 5.6 we show a few highly related events for a few popular topics in our Twitter data set. Specifically given a topic a we rank the meaningful events that contain at least 70 tweets based on $\eta_{k,a}$. We can see from the table that the events are indeed related to the corresponding topic. The event "LionsXII 9-0 Sabah FA" is particularly interesting in that it is highly related to both the topic on Malay and the topic on soccer. (LionsXII is a soccer team from Singapore and Sabah FA is a soccer team from Malaysia.)

5.4 Conclusions

In this chapter, we propose a unified model to study topics, events and users jointly. The base of our method is a combination of an LDA-like model and the Recurrent Chinese Restaurant Process, which aims to model users' longstanding personal topic interests and events over time simultaneously. The Recurrent Chinese Restaurant Process is appealing in the sense that it provides a principled dynamic nonparametric model in which the number of events is not fixed overtime. We further use a time duration-based regularization to capture the fast emergence and disappearance of events on Twitter, which is effective to produce more meaningful events. Finally, we use an inner popularity bias parameter and event-topic affinity vectors to interpret an event's inherent popularity and its affinity to different topics. Our experiments quantitatively show that our proposed model can effectively identify meaningful events and accurately find relevant tweets for these events. Furthermore, the event-topic association inferred by our model can help an event recommendation task and organize events by topics.

Part III

Event Summarization

Chapter 6

Event Summarization

In previous chapters, we defined an event as a set of tweets which are published in a short time period and share similar content. Multinomial distributions over words are used to capture the content of the events. In this case, to interpret these events, users need to read either the sets of tweets or the word distributions. The former is time-consuming and the latter cannot accurately represent the event. Therefore, we propose a novel graph-based summarization framework that generates concise abstractive summaries for the events. Evaluation results show that our framework has better agreement with human summaries compared with baseline methods.

6.1 Introduction

We have explored event identification on Twitter using Bayesian statistical models. Specifically, we utilize topic model and dynamic non-parametric model to detect events. For both methods, each event is modeled as a cluster featured by a multinomial word distribution and each tweet has a latent variable that indicates which event it belongs to. Although the word distribution and the set of tweets of each event are often intuitively meaningful, a major challenge is to accurately interpret the event. Specifically, it is time consuming for users to read all tweets related to the events, and the word distribution of the event can not accurately represent the events. It is thus desirable to generate a concise summary to help the users better understand the story of each event. We propose a summarization framework to generate abstractive summaries for the events using their related tweets.

Our work is related to labeling of topic models [11, 13, 24, 40, 36, 35]. To interpret the semantics of topics, in existing work of statistical modeling, people generally either the select top words from word distributions [11, 13, 24], or generate more meaningful labels in a subjective manner [40, 36]. Mei et al. proposed a probabilistic approach to automatically labeling multinomial topic models in an objective way, which considers the representativeness and redundancy of the labels [35]. However, our task is different from this branch of work, since we are not generating the summaries using the multinomial word distributions of events. More specifically, such word distribution lacks readability because of the bag-of-words assumption, while our task is to generate abstractive sentences from the original tweets.

We propose an abstractive summarization framework based on the directed word graph, where the nodes record the words and the edges together with corresponding weights encode the relative positional information between word pairs. Secondly, we cast the summarization task to finding maximum spanning tree of the graph. Finally, we use a greedy algorithm to select the candidate sentences (i.e. paths), which are highly representative and with low redundancy.

6.2 Method

Our models in Chapter 4 and Chapter 5 can identify events directly from the Twitter stream. We use the set of tweets $I_k = \{d_{t',i'} | y_{t',i'} = 1, s_{t',i'} = k\}$ to represent the set of tweets about event k, where $d_{t',i'}$ means the i'-th tweet in timestamp t', $y_{t',i'} = 1$ means it is event-related, and $s_{t',i'}$ is the latent event variable. The notation follows section 5.2. We present a summarization framework which can generate abstractive summaries for I_k. Section 6.2.1 will introduce the generation of word graph for each event, section 6.2.2 will introduce the generation of the candidate sentences, and finally section 6.2.3 will describe the selection of these candidate sentences. Our approach is similar to the work proposed by Ganesan et al. [20]. They proposed an approach which first builds an *Opinosis-Graph* using the positional information and Part-of-Speech (POS) tags of all the words within the textual reviews, and then picks the top-ranked paths in the graph utilizing various scoring functions. However, their methods cannot be directly applied for our problem mainly because of two reasons: (1) Tweets are informal, which makes the POS tagger unreliable; (2) When a significant event happens, it will cause a large number of tweets talking about it, which makes it computationally costly to score all the possible paths. Thus, instead of considering all paths, in our solution we first cast the word graph to a tree structure.

6.2.1 Word Graph Generation

Our key idea is to use a word graph structure to represent natural language tweet and cast this summarization problem as finding appropriate paths in the graph. Graphs have been commonly used for extractive summarization problems (e.g. [17],[37]). However, in these work, the graph is always undirected with sentences as nodes and similarity as edges. Our graph structure is different, in which nodes represent words with directed edges representing relative positional information between the words within each tweet.

For each event k, we construct a directed word graph $G_k = (V_k, E_k, W_k)$. V_k are the nodes which stand for all the unique words appeared in I_k, E_k is the set of edges, and W_k is the set of edge weights which encode the relative positional information between the word pairs. Table 6.1 outlines the steps involved in building a word graph G_k for event k based on the tweet set I_k. An example word graph for the event 'City Harvest Church Scandal' is shown in Figure 6.2.

The word graph contains some unique properties that are crucial in generating

```
• Set $V_k, E_k, W_k = \{\}$
• For each tweet $d'$ in $I_k$
  - $V_k \leftarrow tokenize(d')$
  - For each word pair $< v_{d,j}, v_{d,j'} >$, where $j = 1$ to $length(d') - 1$ and
    $j' = j + 1$ to $length(d')$
    * if $E_k$ contains edge $v_{d,j} \rightarrow v_{d,j'}$
      . $w_{v_{d,j} \rightarrow v_{d,j'}} + = 1/(j' - j)$
    * else
      . AddEdge($v_{d,j} \rightarrow v_{d,j'}$,$E_k$)
      . AddWeight($w_{v_{d,j} \rightarrow v_{d,j'}} = 1/(j' - j)$, $W_k$)
```

Figure 6.1: Word Graph G_k Generation Process

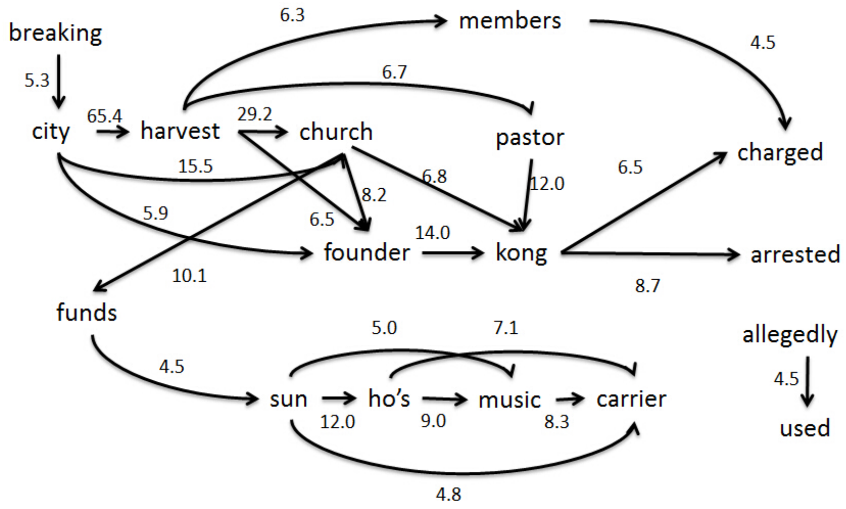


Figure 6.2: Word Graph for City Harvest Church Scandal in Singapore

abstractive summaries: (1) the edge with high weight indicates the corresponding word pair co-occurs closely and frequently; (2) the direction of the edge reflects the readability of the words sequence; (3) gapped positional information between words pairs is captured, which makes it possible to generate new summary sentences.

Given a word graph for a specific event, to generate a summary with N sentences, an intuitive way is to select N valid paths over the graph that can cover as many highly scored edges as possible and reduce the redundancy simultaneously. However, since we record all gapped information between all word pairs, our word graph is densely connected. Thus, to exhaustively enumerate all possible combinations of N paths is computationally very expensive. We therefore go for an approx-

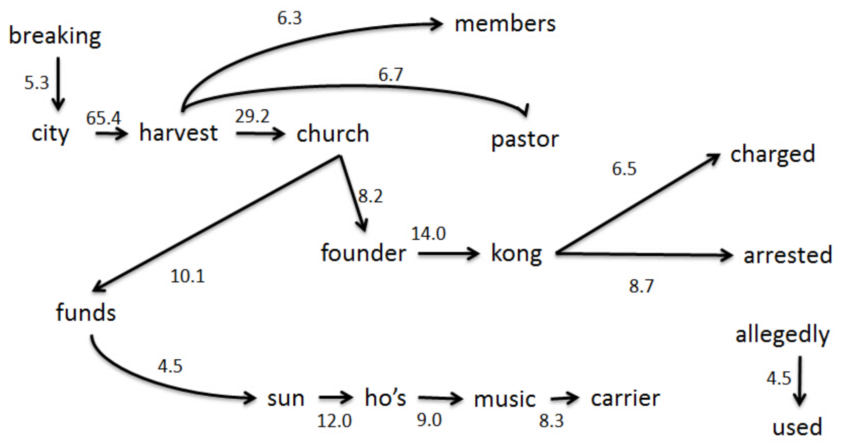


Figure 6.3: Maximum Spanning Tree for City Harvest Church Scandal in Singapore

imate solution where we first find the maximum scoring forest of the word graph and then greedily select highly scored paths from the forest.

6.2.2 Candidate Sentence Generation

Motivated by these properties of the word graphs mentioned above, we can cast the summarization problem as finding paths with high scores in the graph.

Thus, once we get the word graph G_k for each event, we can find the highest scoring spanning tree, which is an instance of the maximum arborescence problem for directed graphs. It can be solved efficiently in quadratic time using the greedy, recursive Chu-Liu-Edmonds algorithm [52, 16]. Note that the word graph G_k may not be a connected graph. In this case, we apply Chu-Liu-Edmonds algorithm to each part of the graph, and we can get a maximum scoring forest. We regard each path (from a root to a leaf) as a candidate sentence for the summary of the event, and we denote the set of all paths as $P_k = \{p_i | p_i = (v_{i,1}, \cdots, v_{i,|p_i|})\}_{i=1}^{|P_k|}$. An example spanning tree of the event 'City Harvest Church Scandal' is shown in Figure 6.3, and the number of paths $|P_k|$ equals to 6 in this case.

```
1. **Input:** $P_k = \{p_i | p_i = (v_{i,1}, \cdots, v_{i,|p_i|})\}_{i=1}^{|P_k|}$
2. **Parameters:** *Threshold*, *N*
3. **Output:** $Out = \{\}$
4. Rank the paths within $P_k$ in descending order, based on the average
   weights along each path.
5. For each path $p_i$ in $P_k$:
   (a) If $Out$ is empty:
         i. AddPath($p_i$, $Out$)
   (b) Else:
         i. For each path $p'_i$ in $Out$:
            A. If $Cosine(p_i, p'_i) >$ *Threshold*: continue and jump to
               step(5).
        ii. AddPath($p_i$, $Out$).
   (c) If $|Out| = N$:
         i. Break and jump to the output step(6).
6. Output the result summaries $Out$.
```

Figure 6.4: Candidate sentences selection

6.2.3 Candidate Sentence Selection

Given the spanning forest of an event, we can score each path within P_k using the average edge weight along the path. However, intuitively, we can not simply select the top-N paths to summarize the event because of the redundancy between each path pairs. For example, the path “city harvest church founder Kong arrested” captures the same meaning as the path “city harvest church founder Kong charged”.

Therefore, we use a greedy algorithm as Figure 6.4 shows. Basically, we want to select the Top-*N* representative paths (i.e. the paths with the highest scores), and reduce the redundancy at the same time. Thus, we compare the similarity with the current candidate path p_i with all currently selected paths Out. If the current path is not similar to all selected paths, we add the path to the output path set Out.

To quantify the similarity between two paths, we measure the cosine similarity between the two paths, while ignoring the shared part of them. We take the two paths “city harvest church founder Kong arrested” and “city harvest church founder Kong charged” as an example. We ignore the shared part, and just measure the similarity between the two remaining parts “arrested” and “charged”. For each remaining part, we learn the language model over all the tweets which contain at least one word in this remaining part. Then, we use the cosine similarity over the two remaining parts as the similarity between the two paths. Specifically, for example we compute the cosine similarity between two language models learnt from the tweets which contain “arrested” and “charged” retrospectively. The intuition behind is that we use all the tweets contains a specific word to represent its semantic meaning, while using external sources like *WordNet* can be an alternative solution.

6.3 Experiments

We evaluate our summarization framework based on the results of Section 5.3. Twenty-seven events are judged as true out of the top-30 events using the proposed method Base+Reg+Aff. For these events, we further ask the two annotators to summarize these events. For each event, they are required to read all the tweets assigned to the event, and then use several sentences or word sequences (including hashtag) to summarize the event. Each sentence is no more than 10 words.

Baseline and Evaluation Metrics

We compare our summarization framework with an existing summarization method [21] (referred to as ILP for short in this part), which utilizes Integer Linear Program for exact inference under a maximum coverage model for automatic summarization. Specifically, it considers information and redundancy at a sub-sentence, “concept” level (e.g. unigram, bigram, etc.), modeling the value of a summary as a function of the concepts it covers. With c_i an indicator for the presence of concept

i in the summary, and its weight w_i, the objective function is:

$$
\begin{aligned}
\text{Maximize: } & \sum_i w_i c_i \\
\text{Subject to: } & \sum_j s_j \leq N & (6.1) \\
& s_j O_{cc_{ij}} \leq c_i, \forall i, j & (6.2) \\
& \sum_j s_j O_{cc_{ij}} \geq c_i, \forall i & (6.3) \\
& c_i \in \{0, 1\}, \forall i \\
& s_j \in \{0, 1\}, \forall j
\end{aligned}
$$

where s_j is an indicator for the presence of sentence j in the summary, and $O_{cc_{ij}}$ is an indicator for the occurrence of concept i in sentence j. The constraints ensure that (1) the number of sentences in the summary is limited by N, (2) selecting a sentence necessitates selecting all the concepts it contains, and (3) selecting a concept is only possible if it is present in at least one selected sentence. Here, we use bigrams as concepts, which is commonly used previous work [21, 14]. The bigrams are weighted by the number of input sentences in which they appear.

Given the human generated event summarization, To compare our framework with ILP, we use ROUGE [33], which is officially applied by Document Understanding Conference (DUC) for document summarization performance evaluation. The summarization quality is measured by counting the overlapping units. Here we use ROUGE-1 and ROUGE-2, which means using unigram and bigram as units retrospectively. ROUGE-1 and ROUGE-2 generate three scores (Recall, Precision and F-measue). For each method, we calculate the three scores by comparing with the two manually generated summaries and then taking the average.

Methods	N	ROUGE-1			ROUGE-2		
		ROUGE-1-P	ROUGE-1-R	ROUGE-1-F	ROUGE-2-P	ROUGE-2-R	ROUGE-1-F
Summarization Framework	1	0.5180	0.4108	**0.4181**	0.3870	0.3067	**0.3054**
	3	0.3098	0.5792	**0.3713**	0.2019	0.4080	**0.2455**
	5	0.2433	0.6967	**0.3373**	0.1458	0.4665	**0.2071**
ILP	1	0.4062	0.5129	0.4180	0.2328	0.3211	0.2439
	3	0.1913	0.7595	0.2943	0.1212	0.5147	0.1870
	5	0.1239	0.7929	0.2081	0.0839	0.5694	0.1410

Table 6.1: ROUGE-1 and ROUGE-2 for our summarization framework and ILP, using $N \in \{1, 3, 5\}$ sentences as summary.

Quality of Summarization

For both our method and ILP, we use top-N sentences to summarize the events, where $N \in \{1, 3, 5\}$. We compare our summarization framework with ILP using ROUGE-1 and ROUGE-2 as evaluation metrics. We average the ROUGE-1 and ROUGE-2 value over the 27 summaries, compared with both manually generated summary, for each method. For our method, we set the similarity threshold (i.e. *Threshold* mentioned in section 6.2.3) to be 0.8. The results is shown in Table 6.1.

From the results, we have several observations: (1) For both ROUGE-1 and ROUGE-2, our method outperforms ILP in terms of F-measure. (2) Compared with our framework, ILP has high recall and low precision. It is mainly because ILP is an extractive summarization method, which aims to selection representative tweets. However, tweet itself contains many informal words, which makes it noisy. Thus, many words within the tweets selected by ILP are not useful(i.e. low precision), although these tweets can capture some important key units of the event (i.e. high recall). However, our method is abstractive, which is able to generate new representative sentence. (3) For ROUGE-2, compared with ILP, our framework performs better, which shows that our method (especially the word graph) can capture the sequential information.

Recall that in section 6.2.3, to reduce the redundancy among the selected sentences, we utilize a similarity threshold (i.e. *Threshold*) to filter out the sentences which is similar to the selected ones. To further test the effect of the threshold, we varies the value of the threshold by fixing number of sentences $N = 5$. The result

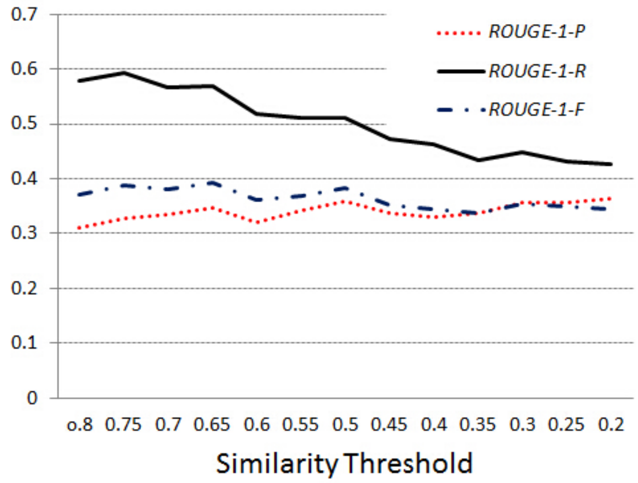


Figure 6.5: ROUGE-1 and ROUGE-2 when varies the similarity threshold for our summarization framework

is shown in Figure 6.5. We can see that, by decreasing the similarity threshold, the precision is increasing while the recall is dropping. It suggests that the redundancy among the selected sentences is increasing.

Readability of Summaries

ROUGE scores measure the quality of the summaries based on the overlapping units shared by the summaries and the ground truth. Here, we further measure the readability of the summaries. We randomly mix the tweets selected by ILP together with the summary sentences generated by our framework ($N = 3$), and ask a human judge for each presented sentence whether it is a tweet (manually written) or not a tweet (automatically generated). The results show that 85.7% of the tweets selected by ILP is judged as real tweets, and 80.1% of the sentences generated by our framework is judged to be real tweets. We take a close look at the summaries, and find that: (1) ILP selects tweets to maximize the information coverage. So it tends to select relatively long tweets. The selected tweets therefore sometimes contain infrequent words. Thus, a small proportion of selected tweets are judged as not true tweets. (2) Our algorithm captures the sequential information using the word graph. Thus, the sentences generated by our framework is relatively readable compared with the real tweets selected by ILP.

6.4 Conclusions

In this part, we propose a word graph based summarization framework to summarize the events on Twitter. The word graph is novel, in which nodes represent words with directed edges representing relative positional information between the words within each tweet. Therefore, we cast the summarization task as finding the paths with highest scores. Then, we utilize the Chu-Liu-Edmonds algorithm and a greedy algorithm to select the representative sentences and reduce the redundancy at the same time. The performance in terms of ROUGE-N shows that our approach outperforms the baseline method.

Chapter 7

Dissertation Conclusion and Future Work

In this chapter, we summarize the findings of this thesis, present an integrated framework for event identification and analysis on Twitter, and point out some future research directions.

7.1 Summary of Contribution

Nowadays, when a significant event happens, the first reaction of the majority is to tweet about it and share their opinions, which makes Twitter the best source to know what happens everyday. It thus triggers event identification and analysis as a crucial task like never before. However, this task is non-trivial because of several challenges: (1) only a small proportion of tweets is event-related, while most of them are about daily routines and personal interests, (2) it lacks study on the interaction between users' tweeting behaviour on events and users' personal interests, especially when both of them are unknown in the tweets flood, (3) a summarization technology is needed to provide concise abstractive summaries for the events on Twitter. To deal with these challenges, this thesis focuses on three tasks, and the corresponding contributions are listed as follows:

Event Identification on Twitter

We used a topic model based approach to identify events on Twitter. The first approach we proposed regards event identification as bursty topics detection from the text streams on Twitter. We introduced a new topic model that considers both the temporal information of tweets and users personal interests, because existing work is not applicable. Then a state machine as a post-processing step is needed to detect the bursty states for the discovered topics. Our experiments showed that our model outperforms standard LDA and its two degenerate variations. However, a limitation of this method is that the number of topics is predetermined, while the events can emerge and die out along the time line. What's more, the model lacks generative ability for the events, so that a post-processing step is needed.

Motivated by the limitations mentioned above, we looked into methods that allow appearance and disappearance of topics along the time line. Recurrent Chinese Restaurant Process provides a principled way where the number of clusters is not fixing over time. However, RCRP cannot be directly applied to our task, since it just captures rich get richer phenomenon while events on Twitter tend to be bursty. Therefore, we proposed a novel duration-based probability discount to RCRP to capture the burstiness character of events on Twitter. We then combined the modified RCRP with topic model to identify both events and topics simultaneously from Twitter. Our experiments demonstrated that our unified model can identify events accurately, which shows the effectiveness of duration-based discount.

Unified Analysis for Topics, Events and Users on Twitter

In the first task, we aimed to separate events from the large proportion of personal interests driven tweets, while ignoring the interaction between the events and personal interests topics. Therefore, we focused on analyzing topics, events and users jointly in this task. Similarly, we combined a modified RCRP with topic model to identify events and topics simultaneously. Then we used an inner popularity bias parameter and event-topic affinity vectors to interpret an events inherent popularity and its affinity to different topics. Our experiments quantitatively showed that our proposed model can effectively identify meaningful events and accurately find relevant tweets for these events. Furthermore, the event-topic association inferred by our model can help an event recommendation task and organize events by topics.

Event Summarization

The approaches mentioned above can detect events by finding all the tweets related to each event. However, it is difficult for users to interpret these events by reading all related tweets. It is thus desirable to generate a concise summary to help the users better understand the story of each event. We proposed a word graph based summarization framework to summarize the events on Twitter. Then we could cast the summarization task as finding the paths with highest scores. Then, we utilized the Chu-Liu-Edmonds algorithm together with a greedy algorithm to select the most representative sentences and reduce the redundancy at the same time. The performance in terms of ROUGE-N showed that our approach outperforms the baseline method.

7.2 Future Direction

In the future, we plan to study a few new directions related to event identification and analysis on Twitter.

- In the event identification task (Part I), we introduced our approaches to identifying events from Twitter streams. Moreover, our work in Chapter 4 can capture the temporal pattern, i.e. birth and death, of the events. However, due to the evolving nature, events often exhibit much more complicated temporal patterns. For example, an event may branch into multiple variant sub-events, and these sub-events overtime often form a evolutionary tree like structure.

Such tree structure can better illustrate the evolving information of a event. To capture the evolutionary tree structure for the events on Twitter, the combination of the Nested Chinese Restaurant Franchise Process [4] and the Recurrent Chinese Restaurant Process[5] can be a future direction. The former one is able to model the hierarchical structure by extending Chinese Restaurant Process to nested structure. The latter makes sure that the path from the root to a leaf follows evolutionary pattern over the time line.

- We explored the interaction among events, users and topics on Twitter in Part II. We proposed a unified model which can identify events, personal interests, and their relations s in a retrospective and offline manner. However, Twitter is appealing in terms of its fast reaction to the events. It can even leads news streams for some events. Moreover, the users always want the freshest events to be recommended. Therefore, it is more practically valuable to develop an online event identification system, and recommend the fresh events to the users based on their personal interests.

- Twitter is a social system with complicated follower and followee networks, which makes it possible to propagate information rapidly. However, our current work only focuses on the textual stream of Twitter, while we ignore the network structure. Intuitively, at the micro level, users' retweet and mention behaviors over the network make an event popular. Therefore, it is interesting to explore what kind of social roles (e.g. whistleblowers) the users play when a certain type of events happens.

With the rapid growth of adoption of Twitter and user generated content in such microblogs, there are always new directions for event identification and event-oriented analysis that trigger urgent and important tasks.

Bibliography

[1] D. Agarwal and B.-C. Chen. fLDA: matrix factorization through latent Dirichlet allocation. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 91–100, 2010.

[2] C. C. Aggarwal and P. S. Yu. A framework for clustering massive text and categorical data streams. In *Proceedings of the SIAM International Conference on Data Mining*, 2006.

[3] A. Ahmed, Q. Ho, J. Eisenstein, E. Xing, A. J. Smola, and C. H. Teo. Unified analysis of streaming news. In *Proceedings of the 20th International Conference on World Wide Web*, pages 267–276, 2011.

[4] A. Ahmed, L. Hong, and A. Smola. Nested chinese restaurant franchise process: Applications to user tracking and document modeling. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1426–1434. JMLR Workshop and Conference Proceedings, 2013.

[5] A. Ahmed and E. P. Xing. Dynamic non-parametric mixture models and the recurrent Chinese restaurant process: with applications to evolutionary clustering. In *Proceedings of the SIAM International Conference on Data Mining*, 2008.

[6] J. Allan. *Topic Detection and Tracking*, chapter Introduction to Topic Detection and Tracking, pages 1–16. 2002.

[7] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 37–45, 1998.

[8] R. Balasubramanyan and W. W. Cohen. Regularization of latent variable models to obtain sparsity. In *Proceedings of SIAM Conference on Data Mining*, 2013.

[9] H. Becker, M. Naaman, and L. Gravano. Beyond trending topics: Real-world event identification on twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media*, 2011.

[10] D. Blackwell and J. MacQueen. Ferguson distributions via Polya urn schemes. *The Annals of Statistics*, pages 353–355, 1973.

[11] D. Blei and J. Lafferty. Correlated topic models. In *Advances in Neural Information Processing Systems*, 2005.

[12] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

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

[14] B. F. D. Gillick and D. Hakkani-Tur. The icsi summarization system at tac 2008. In *Proceedings of the Text Understanding Conference*, 2008.

[15] Q. Diao, J. Jiang, F. Zhu, and E.-P. Lim. Finding bursty topics from microblogs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 536–544, 2012.

[16] J. Edmonds. Optimum branchings. In *Journal of Research of the National Bureau of Standards*, pages 233–240, 1967.

[17] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, pages 457–479, 2004.

[18] S. H. Finley and S. M. Harabagiu. Generating single and multi-document summaries with gistexter. In *Proceedings of the workshop on automatic summarization*, pages 30–38, 2002.

[19] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu. Parameter free bursty events detection in text streams. In *Proceedings of the 31st International Conference on Very Large Data Bases*, pages 181–192, 2005.

[20] K. Ganesan, C. Zhai, and J. Han. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 340–348, 2010.

[21] D. Gillick and B. Favre. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, pages 10–18, 2009.

[22] A. Gruber, M. Rosen-Zvi, and Y. Weiss. Hidden topic Markov model. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2007.

[23] T. Hofmann. Probabilistic latent semantic analysis. In *Proceeding of Uncertainty in Artificial Intelligence*, pages 289–296, 1999.

[24] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of ACM Special Interest Group on Information Retrieval*, pages 50–57, 1999.

[25] 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.

[26] L. Hong, B. Dom, S. Gurumurthy, and K. Tsioutsiouliklis. A time-dependent topic model for multiple text streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 832–840, 2011.

[27] Hongyan and K. R. McKeown. Cut and paste based text summarization. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference,*, pages 178–185, 2000.

[28] A. Ihler, J. Hutchins, and P. Smyth. Adaptive event detection with time-varying poisson processes. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 207–216, 2006.

[29] J. Kleinberg. Bursty and hierarchical structure in streams. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 91–101, 2002.

[30] J. H. Lau, N. Collier, and T. Baldwin. On-line trend analysis with topic models: #twitter trends detection topic model online. In *Proceedings of the 24th International Conference of on Computational Linguistics*, pages 1519–1534, 2012.

[31] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009.

[32] C. Li, A. Sun, and A. Datta. Twevent: segment-based event detection from tweets. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 155–164, 2012.

[33] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, 2004.

[34] C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*, chapter Evaluation in information retrieval. Cambridge University Press, 2008.

[35] Q. Mei, X. Shen, and C. Zhai. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 490–499, 2007.

[36] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: An exploration of temporal text mining. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 198–207, 2005.

[37] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, 2004.

[38] S. Petrović, M. Osborne, and V. Lavrenko. Streaming first story detection with application to Twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–189, 2010.

[39] S. Petrović, M. Osborne, R. McCreadie, R. Macdonald, R. Ounis, and L. Shrimpton. Can twitter replace newswire for breaking news? In *Proceedings of the 7th International Conference on Weblogs and Social Media*, 2013.

[40] H. S. Q. Mei, C. Liu and C. Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proceedings of the 15th International Conference on World Wide Web*, pages 533–542, 2006.

[41] D. Radev and K. McKeown. Generating natural language summaries from multiple on-line sources. In *Computational Linguistics*, pages 469–500, 1998.

[42] D. Ramage, S. T. Dumais, and D. J. Liebling. Characterizing microblogs with topic models. In *Proceedings of the 4th International Conference on Weblogs and Social Media*, 2010.

[43] H. Saggion and G. Lapalme. Generating indicative-informative summaries with sumum. In *Computational Linguistics*, pages 497–526, 2002.

[44] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, pages 851–860, 2010.

[45] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

[46] X. Tang and C. C. Yang. TUT: a statistical model for detecting trends, topics and user interests in social media. In *Proceedings of 21st ACM Conference on Information and Knowledge Management*, 2012.

[47] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448 – 456, 2011.

[48] X. Wang and A. McCallum. Topics over time: a non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 424–433, 2006.

[49] X. Wang, C. Zhai, X. Hu, and R. Sproat. Mining correlated bursty topic patterns from coordinated text streams. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 784–793, 2007.

[50] J. Weng and B.-S. Lee. Event detection in Twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media*, 2011.

[51] W. Xie, F. Zhu, J. Jiang, E.-P. Lim, and K. Wang. Topicsketch: Real-time bursty topic detection from twitter. In *Proc. of the 13th IEEE International Conference on Data Mining*, 2013.

[52] C. Y. J and L. T. H. On the shortest arborescence of a directed graph. In *Sci. Sinica*, pages 1396–1400, 1965.

[53] S.-H. Yang, A. Kolcz, A. Schlaikjer, and P. Gupta. Large-scale high-precision topic modeling on twitter. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1907–1916, 2014.

[54] Y. Yang, J. Carbonell, R. Brown, T. Pierce, B. Archibald, and X. Liu. Learning approaches for detecting and tracking news events. *IEEE Intelligent Systems*, pages 32–43, 1999.

[55] Y. Yang, T. Pierce, and J. Carbonell. A study of retrospective and on-line event detection. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 28–36, 1998.

[56] 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 *Proceedings of the 33rd European conference on Advances in information retrieval*, pages 338–349, 2011.