Hierarchical Event Detection and Clustering in Micro-Blogs using Topic Models

A Project Report Submitted

for the Course

CS499 Project II

by

Harshil Lodhi (Roll No. 11010121)
Nishant Yadav (Roll No. 11010147)
Shobhit Chaurasia (Roll No. 11010179)

to the

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781039, INDIA

April 2015

CERTIFICATE

This is to certify that the work contained in this project report entitled "**Hierarchical Event Detection and Clustering in Micro-Blogs using Topic Models**" submitted by **Harshil Lodhi** (**Roll No.: 11010121**), **Nishant Yadav** (**Roll No.: 11010147**), and **Shobhit Chaurasia** (**Roll No.: 11010179**) to Department of Computer Science and Engineering, Indian Institute of Technology Guwahati towards the requirement of the course **CS499 Project II** has been carried out by him/her under my supervision.

Guwahati - 781 039
April 2015

(Dr. Sanasam Ranbir Singh)
Project Supervisor

ABSTRACT

With the growth of social media, information sharing on micro-blogging platforms such as Twitter has exploded. This huge knowledge base can be leveraged to extract useful information such as real-world events. The dynamic nature of this corpus can be exploited to not only detect, but also model and track the evolution of events over time. The nature of the problem alludes to clustering of tweets based on abstract topics which could be zoomed in to pin-point specific event instances. With this intuition in mind, we have formulated the problem as an instance of Topic Modelling. In this thesis, we present our work on detection of event instances from Twitter data using Topic Models such as LDA. We define an *event* as an abstract idea which has a topic, a temporal dimension, and a set of entities such as location, person, organization etc. associated with itWe have proposed a hierarchical 2-level pipeline for extracting event instances from Twitter data. The first level segregates the tweets into topic clusters where each cluster corresponds to some high-level topic. The tweets in a topic cluster are then segregated based on time. Since event instances are associated with a named entity like location, person etc., within each sub-cluster we extract the named entities and group the tweets based on these entities. Further, entity based post processing steps are applied such as merging of related entities to get final tweet groups. Each group then represents a set of tweets talking about a high level topic (such as *bomb blast*) within a give time-frame having a specific set of entities, hence representing an event instance according to our definition of an *event*. Further, with these event instances as nodes, we propose a simple graph formulation to model the tracking of events over different time-frames.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Humans have a curiosity to know more about their surrounding environment. This need for information is the main factor that has contributed towards the survival of human race. For instance, the news of the outbreak of an epidemic is immediately followed by the adoption of additional health care measures by people living in the affected regions. This huge appetite for information was originally satisfied by written media like newspapers, telegraphs etc. In the present generation, the growth of Internet has completely changed the way information is shared and received. This increase in popularity and need for information sharing has led to the emergence of social networking platforms like Twitter.

Twitter has a huge user base sharing all kinds of information at a very high rate. Information shared on Twitter range from personal information like what they are eating, to local events like festival celebrations, to events having worldwide impact like forest fires. Since the users of Twitter are spread all over the world, and people usually Tweet about events almost instantaneously, it can be considered as a large media company having its reporters spread all over the world reporting events 24*7. This fact makes the study of Twitter data very important in order to model the evolution of important events through tweets. However, along with the diversity and richness of information pervading the Twitter ecosystem comes an equally huge amount of uninteresting, insignificant and noisy information, such as updates about daily chores of a user. Mining useful information from this exploding space of tweets calls for meticulous organization and structuring of data. With this motivation, we wish to explore the problem of detection and tracking of events using micro-blog posts.

Chapter 2

Problem Definition

Our overarching aim is to detect events using microblogs in social media (such as Twitter), track and model their evolution over time. Our focus is on the following two sub-problems.

2.1 Using LDA to detect events

The first step towards an attempt to extract useful information from Twitter data is to detect and extract real-world events from tweets. To this end, we are exploring Topic Models. In particular, we are using Latent Dirichlet Allocation [3]. LDA is a generative process used for inferring the topics present in a text corpora and classifying the documents according to these topics. Our aim is to use LDA on Twitter posts to cluster related posts across millions of tweets under different *topics*. Topics when associated with spatial and temporal data along with the associated entities represent events instances. Our aim is to segregate the tweets within a high-level topic into different clusters where each cluster corresponds to an event instance. We aim to tackle the problem of event detection in a hierarchical 2-level fashion, where the top level represents high-level topics or event classes such as *bomb blast*. The lower level corresponds to a set of tweet clusters, where each cluster representing an event. The advantage of this hierarchical approach over traditional one-layer approach is that further levels based on new attributes could be added to the pipeline to narrow down on event instances with more specificity.

2.2 Evolution of events

Once an event such as bomb blast, hurricane, and presidential speech have been identified through tweets, the next step is to track the evolution of these events over time. We are interested in investigating how they develop within their *topic*, as well as analyzing how their correlation to events in other topics changes over time.

Chapter 3

Understanding Twitter Ecosystem

In this world of big data, no one can ignore the impact that Twitter has in terms of data availability and data processing. On a typical day, more than 500 million tweets are posted. This amount of data was never available before. The traditional media like newspaper, articles, magazines are very different from Twitter data. Tweets provide almost real-time information and discussions of current events. However, tweets are highly fragmented and noisy, and contain non-interesting events as well, such as personal musings of users about their day-to-day activities. Moreover, the informal, ill-framed, and unstructured nature of messages adds to the noise. All these characteristics of tweets make it difficult for the traditional systems which were based on carefully written and well-structured news articles to process Twitter data. Tweets mostly contain different spellings and misspellings for a single word. Because of the 140 character limit, most of the users refrain from the use of proper punctuation and stop-words in their tweets, resulting in grammatically, semantically and syntactically messy texts.

If we consider tweets about users' daily mundane tasks, these tweets come up on Twitter in a large volume on any given day. Intermixed with this uninteresting volume of tweets is a set of equally bursty and information-rich tweets about important events. So, to differentiate between these two classes of tweets, one cannot directly use the frequency; temporal/spatial features of these class of tweets need to be considered. Tweets consisting of daily mundane tasks are evenly distributed across the timeline while tweets about important events are concentrated to a certain part of the temporal and/or spatial dimension.

Events To define an *event*, we need to state what a topic is. Quoting from [18], "A topic is a subject discussed in one or more documents". An *event* is an abstract idea which has a topic, a temporal dimension, and a set of entities such as location, person, organization etc. associated with it. For example, "Death of Steve Jobs" was trending at Twitter. This event has a topic "death" which is associated with an entity "Steve Jobs" and it has a temporal dimension since the burst of tweets appeared in the first week of October, the time when Steve jobs died. The temporal and spatial dimensions can be found explicitly from the tweet content and/or from the tweet's meta-data. A topic alone cannot define an event. So, to make sense from the data, one first needs to find out the different topics from the given data and then, figure out whether there is an associated entity, or a temporal/spatial dimension to it or not.

Chapter 4

Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a probabilistic generative model which automatically and jointly clusters words into topics and documents into mixture of topics. LDA assumes that documents are a mixture of topics which give out words with certain probabilities. Fig. 4.1 represents the plate notation of LDA. The generative process is defined below:

1. Decide on the total number of words that a document will have, lets say N and there are M no. of documents.
2. Choose $\theta_i \sim \text{Dir}(\alpha)$, where $i \in \{1, \ldots, M\}$ and $\text{Dir}(\alpha)$ is the Dirichlet distribution for parameter α.
3. Choose $\phi_k \sim \text{Dir}(\beta)$, where $k \in \{1, \ldots, K\}$.
4. For each of the word positions i, j, where $j \in \{1, \ldots, N_i\}$, and $i \in \{1, \ldots, M\}$.
   - Choose a topic $z_{i,j} \sim \text{Multinomial}(\theta_i)$.
   - Choose a word $w_{i,j} \sim \text{Multinomial}(\phi_{z_{i,j}})$.

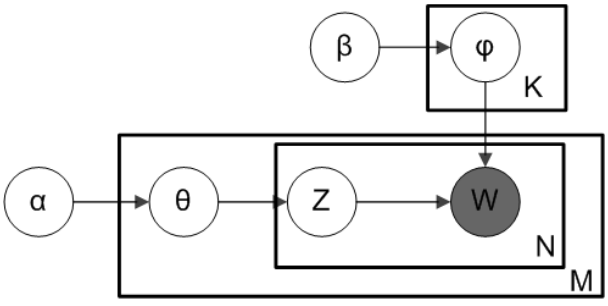

Figure 4.1: LDA Plate Notation

Chapter 5

Literature Review

In recent years, a lot of work has been done in mining hot/trending topics from Twitter stream. Twitter itself shows Trending Topics in its feed. [6] have proposed extensions to this framework by classifying Trending Topics on Twitter using text-based and network-based classifiers. Recently, the focus in literature has shifted beyond the identification of trending topics; the problem of real-world event detection using social media has started receiving overwhelming attention. Researchers have explored the domain of event identification using social media updates, specifically Twitter. While some efforts have focused upon event detection in general [1], other efforts have been directed towards detection of particular class of events such as earthquakes [12], news reports [13]. To this end, numerous techniques have been exploited. While [1] have explored ensemble-based clustering methods for learning similarity metrics for clustering related tweets, [2] have focused on online identification of events using classification techniques. One special technique which has gained our attention, and which we have been exploring is using Topic Models on Twitter data for detection of events.

Topic modeling of the news-wire data has seen a lot of success. Topic modeling techniques such as LDA performs very well on the news data where a document is actually a mixture of a large number of topics. However, the standard LDA doesn't work well on the Twitter data, the major problem being that if we consider a tweet as a single document, then the document is too sparse for the LDA. The most commonly exploited technique to improve the performance of LDA on Twitter data is the aggregation of tweets which are similar in some sense or the other, such as temporally, linguistically, spatially and semantically, so as to feed the topic model with more content-rich documents enabling

better topic inferences.

Several attempts have been made to extend LDA to better capture topics in micro-blogs and tweets. [5] have analyzed different tweet aggregation techniques for training the topic models, in particular, LDA and the Author-Topic (AT) model [11]. The AT model extends LDA by modeling each word in the document as being latently associated to a topic z as well as an author x. An author x is a multinomial (θ) over topics, which in turn is a multinomial distribution (ϕ) over the words w of the vocabulary. Unlike LDA, where only the words are observed, in AT model, both words and authors are observed. Three schemes have been discussed for training the topic models.

1. **MSG scheme:** The focus here is the tweet itself. LDA is trained on all the tweets. For training purposes, each tweet is considered individually as a document.

2. **USER scheme:** The focus here is the user. The model is not trained on individual tweets; rather, aggregation of all tweets of a particular user is fed to the model as a document.

3. **TERM scheme:** This is a rather non-intuitive scheme, where tweets are aggregated based on the terms they contain. For each term, all the tweets containing that term are aggregated. Each document thus contains tweets that have a particular term in common.

The inherent differences in the properties of each scheme can potentially lead to different topic proportions being inferred for the same testing set. For example, under *MSG scheme*, model in trained on individual tweets, which are very short. Hence, more numbers of topics are needed for a reasonable performance. Training set for *USER* and *TERM schemes* have sufficiently large documents. The intuition behind the *TERM scheme* is to capture the topics represented directly by the hashtags in tweets. Further, the authors use JS divergence to measure the similarity between the topics inferred by the three different schemes.

[7] have examined additional tweet pooling schemes, including Burst-score wise pooling, Temporal pooling, and hashtag-based pooling. In Burst-score wise pooling, for each burst term, tweets possessing that term are pooled together into one document. Temporal pooling tries to capture the simultaneous tweets posted by users in wake of an occurrence

of a major event by concatenating tweets posted within the same hour to form a document. The resulting topic inferences were found to be best in case of hashtag-based pooling. This result is quite intuitive, since hashtags inherently represent the context of the tweets; they are topics in disguise, and can be viewed as indirect topic assignment by the humans themselves. However, as noted, only a small portions of tweets are hash-tagged. Moreover, hash tags may not cover all the topics related to the tweet.

LDA as a model is unsupervised in nature. It does not need any labeled documents for the purpose of topic inferences. [9] have proposed Labeled LDA, a supervised version of the standard LDA, wherein a set of labels are provided as parameter to the model to be used as observed parameters for assigning topics to the documents. [8] have explored the nature of Twitter messages and classified them into five different categories. Among these categories, the category of interest to us is the *substance category*, which encompasses tweets about events, ideas, things, or people. They have used Labeled LDA to explore latent dimensions in Twitter posts, and further project these dimensions to the five categories. Further, to exploit the supervised learning aspect of Labeled LDA, they have used labels derived from hashtags, emoticons and Twitter meta-data like replies and temporal information to train the model. The combination of the latent features, and labeled features, and their projection to the *substance category* could potentially lead to very accurate topic-to-event mapping which we are aiming to achieve.

Numerous research efforts have proposed approaches which are branches of the topic modeling paradigm, enhanced by use of other novel techniques and orthogonal ideas. [14] have employed Gaussian mixture for choosing bursty words as potential candidates for being associated to an event. To further evaluate their candidacy, they have employed evolutionary clustering to model the temporal evolution of the candidate topics, before declaring them as being tightly-coupled to an event. They have proposed a time dependent HDP, which is an extension of a yet another powerful topic model - Hierarchical Dirichlet Process (HDP). Similarly, [17] have come up with an Aspect Model called GEAM (General and Event-related Aspect Model) for extracting events information from noisy Twitter data.

Chapter 6

Event Detection

Fig. 6.1 shows the complete pipeline of our event detection system. The first and foremost step is to pre-process Twitter data. This step is of paramount importance because Twitter data is extremely noisy and contains a lot of irrelevant and redundant information that have are either not informative and interesting or have no bearing on event extraction process. The processed Twitter data is then fed into the event extraction pipeline. One of the most important feature of Tweets for the purpose of event extraction is hashtag. Hashtags generally suggestive of the context of the Tweet. Albeit noisy, hashtags attempt to capture the situation, event, or incident which the tweet talks about and is representative of the context embedded in the tweet. However, only a small fraction of tweets are hashtagged by the users. This calls for automatic hashtag recommendation for untagged tweets. The hashtags are of prime importance to the Twitter LDA used in the Topic based Clustering module. It segregates the tweets into topic clusters where each cluster could potentially represent a general topic like bomb-blast or election. This is followed by timeline based segmentation of tweets in each topic cluster. To pin-point

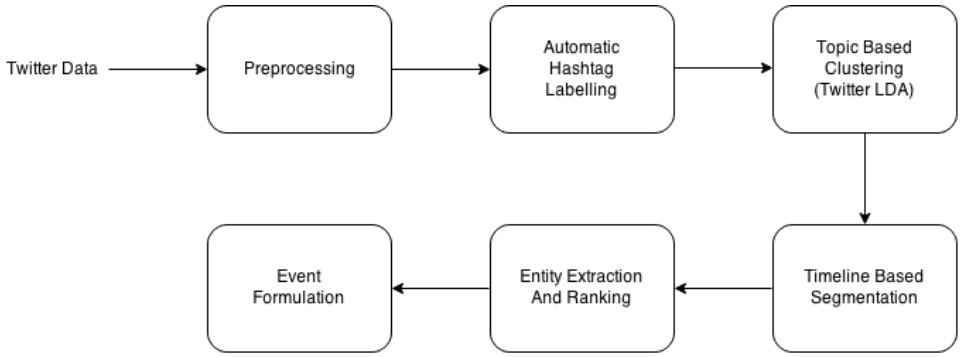

Figure 6.1: Event Detection Pipeline

specific event instances, named entity recognition is employed to extract entity mentions in the tweets. Finally, event instances are formulated by combining the topic, timeline, and entity information extracted from the pipeline. The structure of the pipeline imparts an inherent hierarchy in the event extraction process.

In the following sections, we discuss each module separately in detail, describing the technical details involved.

6.1 Pre-Processing

The pre-processing step involves cleaning the Twitter data to get rid of noise elements. In particular, HTML tags, URLs, user mentions, re-tweets, and hashtags are removed. Next, stop words that occur very frequently such as *a, the, that, etc.* are removed because they are not informative. Additionally, an online slang dictionary [1] is used to convert Internet slang into their corresponding English words. This step is very important because Twitter posts are full of misspelled and slang words because of the restriction on number of characters in a tweet. A comprehensive dictionary based slang conversion works reasonably well because significant portion of the Internet slang is standardized - *u* for you; *2morrow, 2morow, tomorow* for tomorrow; *bcoz, coz, cz* for because.

6.2 Automatic Hashtag Labelling

As discussed above, hashtags capture the context of a tweet. They can be visualized as annotations given by humans to each tweet. Hence, they could, to some extent, be used to capture topics embedded in tweets. Our Topic Extraction module relies on the abundance and reliability of these tagged tweets. While these tags are not a necessity, the performance of the Topic Extraction process increases drastically if a significant number of tweets are hashtagged. However, as observed in [7], only about 22.3% tweets are labelled; even among these, only 19% are specific.

To circumvent this deficiency of tagged tweets, we have used a simple tweet recommendation system. First, all the pre-hashtagged tweets are put together. Then untagged tweets are then scored against each tweet in the tagged tweet pool. The similarity measure used is cosine similarity based on TF and TF-IDF scores. If the similarity score of

[1] `http://www.noslang.com/dictionary/`

an untagged tweet and a tagged tweet exceeds a threshold c, then the hashtags of the tagged tweet are considered as potential candidates for the untagged tweet. Among all the potential candidates, the top $3-5$ candidates are assigned to the untagged tweet.

6.3 Topic Extraction

The first level in our hierarchical event extraction pipeline segregates tweets into topics. Each topic consists of a set of tweets that are related to each other. Each such cluster is intended to correspond to a broad topic such as *bomb blast*, *Hollywood*, *technology*, etc. The intuition is to cluster the tweets into broad sets of related tweets. Each set could potentially contain numerous instances of events that might or might not be directly connected, but could be grouped superficially under the purview of their topic. This hierarchical approach facilitates the event evolution and tracking process. Attempting to zero down on specific event instances directly at first level could be computationally expensive (due to huge size of Twitter dataset and the Named-Entity extraction), as opposed to segregating tweets into smaller, related sets. Moreover, same entities could be involved in events which are completely unrelated to each other. For example, entity *Guwahati* could occur in sporting event as well as movie screening. Without top level topic segmentation, the event evolution and tracking part might incorrectly fall into relating these two events. With high level topic segmentation, event tracking could be accomplished within the logical domain of a topic.

To this end, we employ Topic Modeling techniques like LDA which infer topic distributions in documents in an unsupervised manner. Applying LDA on Twitter data masqueraded as documents gives us logical clusters of related tweets with each clustering representing a topic. Traditional LDA works well on structured documents like news-wire data, articles, and blogs. Due to the noisy, sparse, extremely short, and unstructured nature of micro-blogs, standard LDA fails to infer topic distributions. Several variants have been proposed in literature for the topic inferences on micro-blogs. We have employed Twitter LDA [18] for topic based clustering of tweets.

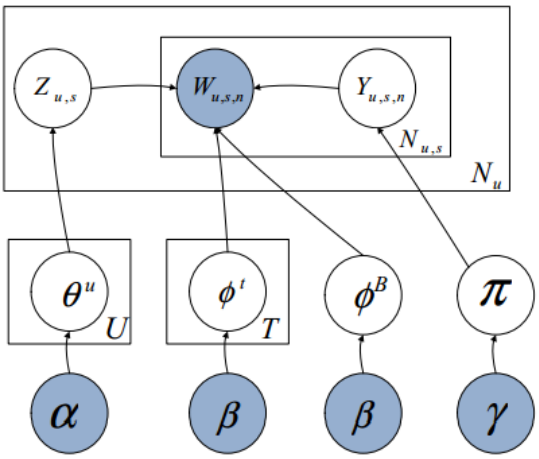

Figure 6.2: Twitter LDA Plate Notation

6.3.1 Twitter LDA

An issue pertaining to the use of LDA on Twitter data is the questionable assumption of considering a tweet as a mixture of topics. Given the extremely short nature of tweets, most tweets consists of a single topic. Some studies have tackled this problem by aggregating the tweets of a user in a single document. This method, though effective, is not guaranteed to help much because of the fact that users generally express a wide variety of different topics in their tweets which may not be related to each other. This analysis will still be good if we want to profile users; but since our aim is to mine events from the topics, the possible solutions that seems feasible will be to aggregate tweets based on time, locality and hashtags.

To this end, [18] have proposed an effective variant of the standard LDA, called Twitter LDA. It is based on the assumption that a tweet will contain a single topic chosen from a topic distribution of a particular user.

Model Description The generative model makes the following assumptions. Twitter data has T number of topics. When a user tweets, he/she selects a topic from his/her favorite list of topics, these topics will be from the T topics. Then for the selected topic, the user selects a bag of words, one by one from the distribution of words over topics. However, all the words of the tweet need not necessarily describe the topic. Many of them are just common words occurring in tweets of various different topics. So for each of the words, user first decides whether the given word is closely describes the topic or

1. Draw $\phi^{\mathcal{B}} \sim \text{Dir}(\beta), \pi \sim \text{Dir}(\gamma)$
2. For each topic $t = 1, \ldots, T$,
   (a) draw $\phi^t \sim \text{Dir}(\beta)$
3. For each user $u = 1, \ldots, U$,
   (a) draw $\theta^u \sim \text{Dir}(\alpha)$
   (b) for each tweet $s = 1, \ldots, N_u$
      i. draw $z_{u,s} \sim \text{Multi}(\theta^u)$
      ii. for each word $n = 1, \ldots, N_{u,s}$
         A. draw $y_{u,s,n} \sim \text{Multi}(\pi)$
         B. draw $w_{u,s,n} \sim \text{Multi}(\phi^{\mathcal{B}})$ if $y_{u,s,n} = 0$ and $w_{u,s,n} \sim \text{Multi}(\phi^{z_{u,s}})$ if $y_{u,s,n} = 1$

Figure 6.3: Twitter LDA Generative Process

not. Depending upon this the word will be classified as a topic word or a background word. He then chooses the word from its respective distribution.

Formally, let θ_u denotes the topic distribution for a user u. Let ϕ_t be the distribution of words for the topic t and let ϕ_B be the distribution of background words. π is a Bernoulli distribution which denote the word belongs to the background class or to the topic class. α, β, γ are Dirichlet parameter used for generating respective Dirichlet distributions. The plate notation and generative algorithm are given in Fig.6.2 and Fig.6.3.

We have used a variant of Twitter LDA for topic based clustering. Instead of aggregating tweets based on users, pooling tweets based on hashtags gives more coherent and logical topic clusters. Hashtag based tweet aggregation seems more intuitive because hashtags can be thought of as crude topic assignments manually given by users. Although LDA works in an unsupervised fashion, aggregating related tweets into a document could indirectly guide the inference process towards more sound topic clusters.

The Topic-based Clustering module was tested on the dataset using the tweaked version of Twitter LDA for different number of topics, such as $T = 25$, $T = 50$, and $T = 100$. This segmented the tweets into clusters based on high level abstract topics. Each topic cluster could potentially contain numerous event instances. While high level topics correspond to the class of events, specific event instances are almost always associated with one or more entities such as location, person, organization etc. The next logical step would be, thus, to identify named entities in topic clusters to identify event instances. The Entity Extraction module deals with entity extraction and additional processing to pin-point event instances.

6.4 Timeline based Segmentation

Before proceeding to entity based segmentation, we segment each topic cluster based on timeline, creating sub-clusters within each topic corresponding to non-overlapping window lengths of N days. Timeline based segmentation enables modeling of event evolution and temporal tracking.

6.5 Entity Extraction and Ranking

After getting the time segmented cluster of tweets, our aim is to find the important words, places, persons that will represent an event. For our purpose, we used the Twitter NER [10]. As noted in [10], the reason behind using a NER specifically trained for twitter is that the tweets are generally noisy and that traditional NER tools do not work well for micro-texts. We have used the NER to extract the named entities and location information from the tweets. The named entities captured by Twitter NER includes persons, locations, organizations, movies etc. A subtlety to note is that the NER finds the location information from the text by using only the semantics of the text. To further increase the accuracy, we also pipelined NER with a dictionary based location detection mechanism which uses publicly available geo dictionary.

A significant number of tweets will not have any entity mentions. We can safely ignore these tweets because it is highly unlikely for a tweet to talk about an event without mentioning even a single event phrase or an associated entity.

6.5.1 Entity-based Clustering

Once we have extracted entities for each of the tweets, we proceed forward to make a entity set along with frequencies for each of the tweet sub-cluster (i.e. for each time-segment within each high-level topic). Tweets are grouped based on the entities present in them. At this second level, each tweet group comprises of tweets talking about a specific entity within a specific time segment under a high level topic. This level hence captures the high level topic, the temporal aspect, as well as the entity aspect of an event. On a superficial level, each tweet cluster at this level could potentially represent an event instance. However, there are numerous subtle fallacies in this approach, some of which

are discussed and tackled below.

First, we rank each entity cluster using different scoring functions like frequency of occurrence and TD-IDF ranking. Only top e tweet clusters (under the umbrella of a high-level topic, time segment, and an entity) are reported as event instances. Below, we discuss some of the observations of TF and TF-IDF ranking.

TF Ranking Each entity is scored by directly counting the number of tweets that mention that entity. Entities that have been more frequently mentioned are more likely to be involved in a trending event. Hence, focusing on most talked about entities works reasonably well for the purpose of detection. However, the analysis of our test runs on the dataset revealed a subtle flaw in this approach. Some common entities like Twitter, Facebook, Myspace, YouTube appeared among the top entities set in every topic. Manual cross-checking with the dataset revealed that it was indeed true that surprisingly large number of tweets mentioned these entities. While some tweets did indeed talk about an event related to these entities, most of them represented general personal experiences and musings of the users with these social platforms. These overly frequently entities are analogous to stop-words in a text. While, in principle, they could potentially be informative (and hence, represent an event instance), more often than not, they are associated with noise information.

TF-IDF Ranking One possible way to mitigate the issue of over-rewarding of the "stop-word entities" is to completely ignore these entities from our pipeline. However, this approach seems to pessimistic since occasionally these entities could indeed correspond to an event instance. A more optimistic way to circumvent this issue is to use TF-IDF ranking of entities, with TF being the number of tweets in a time segment mentioning the entity, and IDF being the inverse of the number of time-segments over all high-level topics in which this entity occurs. While this ranking handles the issue of "stop-word entities", it introduces other issues. Most of the entities that came at top were the ones with TF-IDF score 1, i.e. those that occurred only in a single time segment. This could be handled by ignoring entities present only in very few time segments. Once again, this is an approximation which could lead to discarding of many event instances. This is based on the assumption that it is highly unlikely that an event occurs which is talked about only during very few time periods and the number of tweet mentions drops down

significantly outside those time frames. This could be ensured by meticulously choosing the length of a time segment and TF-IDF threshold.

6.5.2 Merging Entities

Our abstraction of associating each entity based cluster at the second level to an event instance is primitive and flawed. It fails to capture the fact that an event instance could be associated to more than one entities. To this end, some post-processing is done on the entity sets to combine logically connected entities such as Apple and Ipad into a single entity. Without entity merging, we could incorrectly infer that the Apple cluster and the Ipad cluster correspond to two separate event instances. To this end, we propose entity merging schemes based on maximum common subsequence and co-occurrence frequencies. These are discussed below.

Maximum Common Subsequence First and foremost, we merge entities that indeed correspond to the same entity in the real world, but were extracted as separate ones because of common spelling variations or the entity being composed of two or more space separated words, with different users using different components of the entity words to refer to the entity. For example, entities like *Bieber* and *Justin Bieber* actually refer to the same person, but both of them are used to refer to him. Such entities could be merged by computing the maximum common subsequence (potentially dis-contiguous) between each pair of entities and merging those whose maximum common subsequence exceeds a threshold. Advantage of looking for common subsequence over common substring is that it handles spelling variations as well.

Direct co-occurrence frequency Co-occurrence of each pair of entity is computed within a high-level topic cluster by counting the number of tweets t_k in which the two entities e_i and e_j co-occur using the following formula.

$$\frac{\sum_k \delta(e_i, t_k) \cdot \delta(e_j, t_k)}{K} \tag{6.1}$$

where, K is the total number of tweets in the high-level topic cluster, and

$$\delta(e_i, t_k) \;=\; = \begin{cases} 1 & e_i \in t_k \\ 0 & otherwise \end{cases}$$

Direct co-occurrence based merging does not give acceptable results because it is unlikely that a single tweet contains both the entities that we should be logically merged. This can attributed to the restriction on the length of each tweet as well as the general inclination towards brevity in micro-blogs as opposed to redundancy in larger articles. Hence, the co-occurrence scores are subdued.

Hashtag based co-occurrence As noted above, it is unlikely for two related entities to co-occur in a single tweet. However, in a set of tweets related to the two entities, the co-occurrence is expected to be high. In this method, instead of measuring co-occurrence at individual tweet level, we measure co-occurrence at the level of a group of related tweets for gauging the logical relation of two entities. This tackles the sparseness at the level of individual tweets by expanding the domain of search to a collection of tweets.

Hashtags are exploited to construct the set of tweets related to a pair of entities. Since hashtags capture the context of a tweet, tweets with same hashtags are likely to be related, albeit on a potentially superficial level. Hashtags are noisy and the same hashtag could correspond to different situations. This problem could be mitigated to some extent by restricting the domain of search to the high-level topic cluster returned by Twitter LDA or to an even more restrictive time segmented cluster.

6.6 Event Formulation

Each entity cluster at the bottom level (after performing entity merging) represents an event instance. At this level, each cluster is associated with a high level topic (assigned by Twitter LDA) like *bomb blast*, a time segment (due to timeline segmentation), and a set of entities (assigned by Twitter NER and entity merging). An event instance is formulated as an object having a number of attributes which include a set of entities, time frame, high-level topic, and a set of keywords describing an event. The set of keywords describing the event are extracted using Twitter NER, which has a special tag *B-EVENT* for words describing an event. These words generally include verbs, adjectives, and adverbs like *quitting*, *death*, *going up* etc.

Chapter 7

Event Evolution and Tracking

A lot of research is going on to study the evolution of events over time in general. [16] studied common patterns and progression stages in event sequences like medical records, reviews of products and services and web/search logs. They build a generative model to study the common patterns in general event evolution. They assigned different classes to different sequences on the basis of their evolution hierarchy. [4] in their work tracked evolution of news stories for the purpose of event summarization. The core idea of their work is based on visual cues and textual information. News channels often repeat shots of a video multiple times during news broadcast. They have exploited this observation and used visual cues to detect repeating videos in a news to keep track of an event evolving over time. Evolution of events within a topic related to an incident using online news has been studied by [15]. Significant research efforts have been made to study evolution of events in general. However, the effects of LDA-segmented tweets-cum-events on their tracking and evolution has not been studied in the past.

We didn't delved into complex methods of event tracking and evolution since our pipeline has already has identified event's major keywords. We represented each event as a cluster of topical words and use the well known Maximum Weighted Bipartite Matching algorithm for evolution and tracking.

7.1 Problem Formulation

A particular days event clusters obtained at the previous stage of the pipeline are grouped together to form one side of the bipartite graph same as in the bipartite graph matching

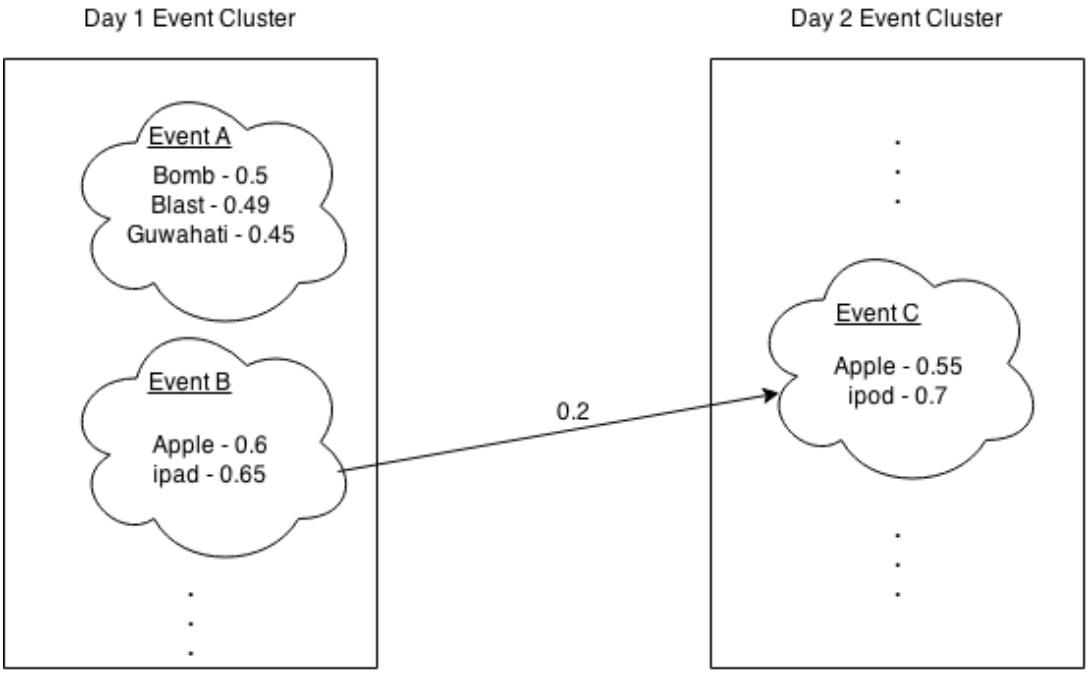

Figure 7.1: Formulation of event tracking as bipartite graph

problem. Now edges are constructed between event clusters of consecutive days events based on the similarity between the event clusters. Thus, forming an event chain over time showing the evolution and also its relation with other event chains on the social media.

The similarity between two event clusters on consecutive days is determined by the ratio obtained from adding the multiplied tweet frequency of the common topical words in the two event clusters and then dividing by the number obtained in the numerator plus the sum of the tweet frequency of the uncommon topical words in the two event clusters. The ratio obtained for the two event clusters determines the weight of the edge between them if above a certain threshold. On the other hand, if the ratio is below a certain threshold then no edge is constructed between the two events. Formally s is the similarity between two clusters and its is defined as ratio of sum of frequency of intersection word set I to the sum of frequency of union word set U.

$$s = \frac{\sum_i f_i \in I}{\sum_j f_j \in U} \tag{7.1}$$

In this way, it is assured that event chain is constructed for related events only and thus is showing the actual progress of the event over time on social media. This whole process can be easily understood from the diagram given below.

We can see from the diagram that there is no edge between event A and event C because of the low similarity ratio between them as should be the actual case. Since, there is no connection between a Bomb Blast and an event related to Apple Inc.. Whereas, there is an event edge between event B and event C as should be the actual case. Since, it shows the product release activity of Apple Inc. which is relevant. So, we can say that our approach for constructing an event chain is in line with the actual event chains that exists in the social media.

Chapter 8

Observations and Results

8.1 Twitter LDA

We compared the results of the normal LDA and the Twitter LDA and found that Twitter LDA is significantly better on the microblogs. As we analyzed our data for different no. of topics like 25, 50 and 100, we found that in reality a single topic was representing a lot of events over the timeline. For eg. the Topic 1 in 8.1 also had events related to both American Idol and Britain's Got Talent. Some of the top keywords like vote,won,trending,talent are perfectly representing both the events.

Topic 1	Topic 2
vote	air
won	france
love	flight
diversity	plane
susan	sad
trending	mcflyforgermany
boyle	families
talent	missing

Table 8.1: Twitter LDA - Topics' top keywords

We also found that there were some very general topics that were representatives of daily activities of the user and were found in the output of the Twitter LDA.

8.2 Entity Extraction and Time Segmentation

After getting the topics from the Twitter LDA, we segmented the tweets in topics based on time and extracted top entities for it. The results we got in terms of entities were found to be representing many events as predicted earlier.

Segment 1	Segment 2	Segment 3
2009-04-06 - 2009-04-21	2009-05-21 - 2009-06-05	2009-06-05 - 2009-06-20
robert - 4	shaun - 37	ashley - 33
youtube - 5	shaun smith - 40	danny - 25
adam - 6	danny - 40	twitter - 169
twitter - 21	adam - 80	tom - 47
miley cyrus - 3	twitter - 252	peter - 35
jessica - 3	kris allen - 69	kate - 56
susan - 3	susan - 90	margaret - 71
adam lambert - 4	adam lambert - 96	adam lambert - 32
simon cowell - 3	susan boyle - 484	david - 38
susan boyle - 58	mtv - 66	teen choice awards - 60

Table 8.2: Time segments of a topic with frequency of each entity in that segment

The three sub-topic clusters shown in Table 8.2 are the three different segments taken from one of the top level topics that we got from topic models.

We manually searched for these keywords to find out about coherence of topical words. A/c to our observations, the first subclusters represents a amazing performance of Susan Boyle in Simon Cowell was the judge. Adam Lambert was another performer who did a phenomenal performance in American Idol 8. So we have two different events related to stage performance in a single time segment. To merge keywords, we have used the co-occurence of words as already described in 6.5.2

The next two subclusters denote events that succeed the first one. One can easily see the relation between these subclusters based on keywords and this relation will be exploited by our Event Tracking Algorithm.

Chapter 9

Conclusion And Future Work

Earlier many people have tried various adhoc approaches on micro-blogs to detect, track and analyse events. In our work we have explored a different approach by using topic models on tweets to cluster them according to the topics they belong. This clustering of tweets helps in organizing the micro-blog information into high-level abstraction-based clusters. But, a high-level topic alone does not represent a specific event instance. Instead, a topic may have numerous instances of different events which come under the purview of the broad topic. To this end, we have proposed a hierarchical approach by temporally segmenting the tweets in a topic and then finding the named-entities present in the tweets sub-cluster. After merging related entities, each group of tweets represents an event instance. The advantage of the hierarchical approach is that further hierarchies based on additional features can be added to the pipeline to add more specificity to the result. For example, since location is an important attribute associated with an event, spatial segmentation could be added to create a 3-level hierarchical framework. Our tests on a Twitter dataset having 1.6 million Tweets shows promising results, with many event instances extracted from the corpus.

After event detection we track events over time to study their evolution. We have modeled the problem as an instance of maximum bipartite matching. We have achieved this by taking intersection of keywords words describing an event with events on another time-frame. This simple event tracking technique used by us looks very promising for tracking events as we have used weighted edges to show relationship between events. These weighted edges also give a good idea of the degree of correlation between events.

There is a lot of potential in carrying our analysis further. One direction of work

could focus specifically on development of variants of Topic Modeling techniques specific to micro-blogs. Our current framework clusters tweets at the bottom level using direct entity presence. While this approach works reasonably well, it is admittedly crude and simplistic. More involved approaches such as supervised/semi-supervised clustering techniques such as Labelled LDA [9] could be employed at this level to extract even non-popular event instances which might otherwise get subdued by trending ones. Finally, more involved time-series models such as ARIMA could be employed for event tracking.

Chapter 10

Work division

- **Harshil Lodhi:** Tweets cleaning and pre-processing module, Named-Entity based clustering, Entity Merging - Hashtag based co-occurrence method.
- **Shobhit Chaurasia:** Automatic hashtag labeling, Twitter LDA variant, Entity Merging - Direct co-occurrence frequency based method.
- **Nishant Yadav:** Timeline based tweet segmentation, Entity ranking - TF and TF-IDF scoring, event evolution and tracking.

Bibliography

[1] Hila Becker, Mor Naaman, and Luis Gravano. Learning similarity metrics for event identification in social media. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 291–300. ACM, 2010.

[2] Hila Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: Real-world event identification on twitter. pages 438–441, 2011.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[4] Pinar Duygulu, Jia-Yu Pan, and David A Forsyth. Towards auto-documentary: Tracking the evolution of news stories. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 820–827. ACM, 2004.

[5] Liangjie Hong and Brian D Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, pages 80–88. ACM, 2010.

[6] Kathy Lee, Diana Palsetia, Ramanathan Narayanan, Md Mostofa Ali Patwary, Ankit Agrawal, and Alok Choudhary. Twitter trending topic classification. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pages 251–258. IEEE, 2011.

[7] Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 889–892. ACM, 2013.

[8] Daniel Ramage, Susan T Dumais, and Daniel J Liebling. Characterizing microblogs with topic models. 2010.

[9] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.

[10] Alan Ritter, Sam Clark, Oren Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics, 2011.

[11] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press, 2004.

[12] Takeshi Sakaki, Makoto Okazaki, and Yutaka 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. ACM, 2010.

[13] Jagan Sankaranarayanan, Hanan Samet, Benjamin E Teitler, Michael D Lieberman, and Jon Sperling. Twitterstand: news in tweets. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 42–51. ACM, 2009.

[14] Xun Wang, Feida Zhu, Jing Jiang, and Sujian Li. Real time event detection in twitter. In *Web-Age Information Management*, pages 502–513. Springer, 2013.

[15] Christopher C Yang, Xiaodong Shi, and Chih-Ping Wei. Discovering event evolution graphs from news corpora. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 39(4):850–863, 2009.

[16] Jaewon Yang, Julian McAuley, Jure Leskovec, Paea LePendu, and Nigam Shah. Finding progression stages in time-evolving event sequences. In *Proceedings of*

the 23rd international conference on World wide web, pages 783–794. International World Wide Web Conferences Steering Committee, 2014.

[17] Yue You, Guangyan Huang, Jian Cao, Enhong Chen, Jing He, Yanchun Zhang, and Liang Hu. Geam: A general and event-related aspects model for twitter event detection. In *Web Information Systems Engineering–WISE 2013*, pages 319–332. Springer, 2013.

[18] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval*, pages 338–349. Springer, 2011.