A PROBABILISTIC TOPIC MODELING APPROACH FOR EVENT DETECTION IN SOCIAL MEDIA

By

Courtland VanDam

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Computer Science

2012

ABSTRACT

A PROBABILISTIC TOPIC MODELING APPROACH FOR EVENT DETECTION IN SOCIAL MEDIA

By

Courtland VanDam

Social media services, such as Twitter, have become a prominent source of information for event detection and monitoring applications as they provide access to massive volume of dynamic user content. Previous studies have focused on detecting a variety of events from Twitter feeds, including natural disasters such as earthquakes [41] and hurricanes [20] and entertainment events, such as sporting events [27] and music festivals [35]. A key challenge to event detection from Twitter is identifying user posts, or tweets, that are relevant to the monitored event. Current approaches can be grouped into three categories—keyword filtering, supervised classification, and topic modeling. Keyword filtering is the simplest approach but it tends to produce a high false positive rate. Supervised classification approaches apply generic classifiers, such as support vector machine (SVM), to determine if a tweet is related to the event of interest. Their performance depends on the quality of features used to represent the data. Topic modeling approaches such as latent Dirichlet allocation (LDA) [4] can automatically infer the latent topics within the tweets. However, due to the unsupervised nature of the algorithm, they are not as effective as supervised learning approaches. The approach developed in this thesis combines probabilistic topic modeling with supervised classification to leverage the advantages from each approach. This supervised topic modeling approach, called subtopicLDA, utilizes label information to help guide the topic model to select topics that best fit the label information. The model is evaluated for its effectiveness in detecting foodborne illness related tweets.

ACKNOWLEDGMENTS

I would like to thank my advisor and my committee for guiding me through this research process. I would like to thank James Daly, Jason Berg, Andrew Cole, Zachary Graham, and John Bellamy for labeling data.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Social media platforms such as Twitter[1], Facebook[2], and YouTube[3] have enabled millions of users to post real-time status updates as events unfold around them. Due to their broad coverage and dynamic user-generated content, they have become a prominent source of information for event detection and monitoring applications. In particular, Twitter has emerged as a popular social media platform for event detection because their streaming data can be easily collected through an application programming interface (API). For example, Twitter has been successfully used to detect a variety of events, from natural disasters such as earthquakes [41], hurricanes [20], floods [47], and wildfires [47] to sporting [20, 27] and entertainment events [13, 20]. More recently, there have been increasing interests in using Twitter feeds as a potential data source for health monitoring, with the majority of the work focusing on influenza monitoring [1, 8, 9, 14, 24, 31, 32, 38]. This thesis investigates the problem of identifying tweets (Twitter postings) pertaining to foodborne illness events using a supervised learning approach. The possibility of using Twitter for monitoring food safety and foodborne illness was suggested by Newkirk et al. in [30], but they did not perform any experiments to demonstrate this potential.

[1] https://twitter.com/
[2] https://www.facebook.com/
[3] https://www.youtube.com/

1.1 Event Detection from Twitter Data

There are three main challenges that must be addressed when using social media such as Twitter to detect health related events. The first challenge is to determine which tweets are relevant to the event. Some tweets may have common terms relevant to a given event but describe something other than the event in question. For instance, a tweet containing the term "earthquake" could be about an actual earthquake or it could refer to a conference on earthquakes [41]. To identify relevant tweets, a variety of methods based on *keyword filtering*, *supervised classification*, and *topic modeling* approaches have been developed. Keyword filtering is a simple approach based on the assumption that every time a keyword appears in a tweet, it is related to the monitored event. The effectiveness of this approach depends on the choice of keywords used. If the keywords are too general, this may lead to a high false alarm rate whereas if they are too narrow, they may not cover all the tweets relevant to the event of interest. Supervised classification methods employ discriminative classifiers such as support vector machines (SVM) to help remove a significant proportion of unrelated tweets. Since these are mostly generic, off-the-shelf classifiers, applicable to any input data, they are not designed to exploit specific properties of the Twitter data. The classifiers typically examine the words that appear in the tweets, without considering their contexts or the underlying topics that generate those words. Any missing or ambiguous words will have an adverse impact on the performance of the classifier. Topic modeling approaches are designed to discover the underlying topics from which the words are generated. Due to the probabilistic nature of such models, they can handle uncertainties due to missing or ambiguous terms. However, since topic modeling is mostly an unsupervised learning approach, the discovered topics need to be manually analyzed during post-processing to discover which topic contains

words that best describe the event.

The second main challenge is to determine *when* did the event occur. Depending on the type of event, the start time could be as small as a few minutes before it was first discussed, such as for an earthquake, or as large as several months after it was discussed, such as a general election. Defining the precise time of an event such as an influenza outbreak is trickier. A common strategy is to use the frequency of the tweets to determine the start time of the event. For example, the mode of the frequency distribution could be used to signify the time of the event.

The third main challenge is to determine *where* the event occurred. The location of an event can be a city or span a wider geographical region. The simplest way to determine where an event occurred using Twitter is to use the location information listed in the users' profiles. The assumption here is that the user profiles contain the actual locations where users spent most of their time. This is not always the case, as the location listed could be fictional or the event may occur at a different place than where the users typically reside. Thus, one may need to parse the content of the tweets to find out where the event took place.

The focus of this work is only on the first challenge, determining which tweets are relevant to the event. Classifying Twitter data to identify relevant tweets is a challenge because the tweets are inherently noisy. The tweets can describe about anything and some tweets use the same terms that describe an event of interest, such as symptoms of an illness, to describe their opinion of something in their life. Tweets may also contain abbreviations for words or misspelled words which adds noise to the data. Additionally, since the lengths of the tweets are short, there is only limited information available for the classifier to correctly identify them. This makes determining whether a tweet relates to the event a challenging

problem. Using generic classifiers to classify tweets may not produce good results because the classifiers often struggle with short, noisy text. To overcome these challenges, this work uses a probabilistic topic modeling approach to discover the hidden topics from which the tweets are generated, and thus, allows for more accurate identification of relevant tweets.

1.2 Topic Modeling

Topic modeling has been successfully applied to model different aspects of Twitter data. For example, it has been used to classify users [18, 36, 49], to compare Twitter's textual content against traditional news medium [51], to predict the topic of the next tweet of a user based on the topics of previous tweets [48], and to find the most relevant terms for searching in Twitter [50]. Previous research has also used topic modeling to determine which tweets are relevant for event detection. Most notably, Paul and Dredze proposed an Ailment Topic Aspect Model (ATAM) to cluster Twitter postings according to their types of ailments [31, 32]. Unlike other works that consider only one ailment at a time, Paul and Dredze have employed topic modeling to identify 20 different types of ailments. However, since ATAM is an unsupervised learning approach, a post-processing step is needed to manually determine which illness each of the model's variables represent by looking at the most prevalent terms associated with each ailment. For example, to monitor an influenza outbreak event, Paul and Dredze aggregated all the tweets assigned to the influenza ailment.

ATAM is a variant of the Latent Dirichlet Allocation (LDA) [4] model, which was originally developed for inferring hidden topics within a document corpus. These models assume each document is a bag of words generated from a mixture of randomly drawn topics. In turn, the probability of a word appearing in the document is conditioned on the topic for that

word, which is randomly chosen from the topic distribution associated with the document. A drawback to LDA and ATAM is that they are both unsupervised, so the topic distribution of a document may not be correlated to its class label. There have been several attempts to extend LDA to incorporate the document's label, forming a supervised version of LDA. Many of these methods, such as LabeledLDA [37], assume a topic distribution is generated for each document (rather than for each class). As a result, two documents can have very different topic distributions even though they belong to the same class. Despite their greater flexibility, such models have more parameters to be estimated from the data and require additional overhead to store a separate topic distribution for each document. Instead, the topic model proposed in this thesis, called subtopicLDA, considers a topic distribution for each class. Two documents from the same class will have the same topic distribution but their words may differ since each topic can have a different word distribution (and the same word can be generated from different topics). In subtopicLDA, the expected value for the topic distribution of each class can be used to predict the label of a new document based on which class has the highest probability. The number of comparisons is limited to the number of classes, which tends to be significantly smaller than the number of documents in the training set.

1.3 Case Study: Foodborne Illness Outbreak

To demonstrate the effectiveness of our proposed subtopicLDA model for event detection, we consider the problem of identifying relevant tweets about foodborne illnesses. Why study foodborne illness? Foodborne illness is a common health concern that can be costly to individuals and society as a whole. Although most people who become ill with foodborne

illness are able to get better on their own without medical attention, some who get sick need to be hospitalized or may die from foodborne illness. In 2011, 47.8 million people became ill with foodborne illness. Of those, 128 thousand needed hospitalization and 3 thousand died from complications triggered by the foodborne illness[12]. In this study, the main bacteria considered are salmonella, listeria, and escherichia coli (e. coli), which are among the most common bacteria that cause foodborne illness. The symptoms of salmonella, listeria, and e. coli are similar to symptoms of other bacteria that contribute to foodborne illness. These symptoms are used to determine if a user is ill, and they are used as keywords for data collection.

A second reason to study foodborne illness, specifically in social media, is the long delay time between when a patient first becomes ill to when an outbreak is confirmed. From Figure 1.1, the time between when the patient first becomes ill to when they seek medical attention can take several days. Once a bacteria is identified, determining which food contained the bacteria can take a few days to several weeks. By using social media, it may be possible to shorten parts of this cycle and assist the Center for Disease Control (CDC) in determining the existence of an outbreak sooner. If many users post their symptoms after consuming the same type of food, this information could be aggregated to detect a foodborne illness outbreak.

Foodborne illness has several characteristics that make detecting it more challenging than monitoring other illnesses such as influenza. The biggest difference is that foodborne illness outbreaks do not follow a seasonal pattern, unlike flu rates, which tend to increase during the winter months and decrease during the summer months. Foodborne illness can happen anytime of the year, since similar foods are available year round from different regions. For

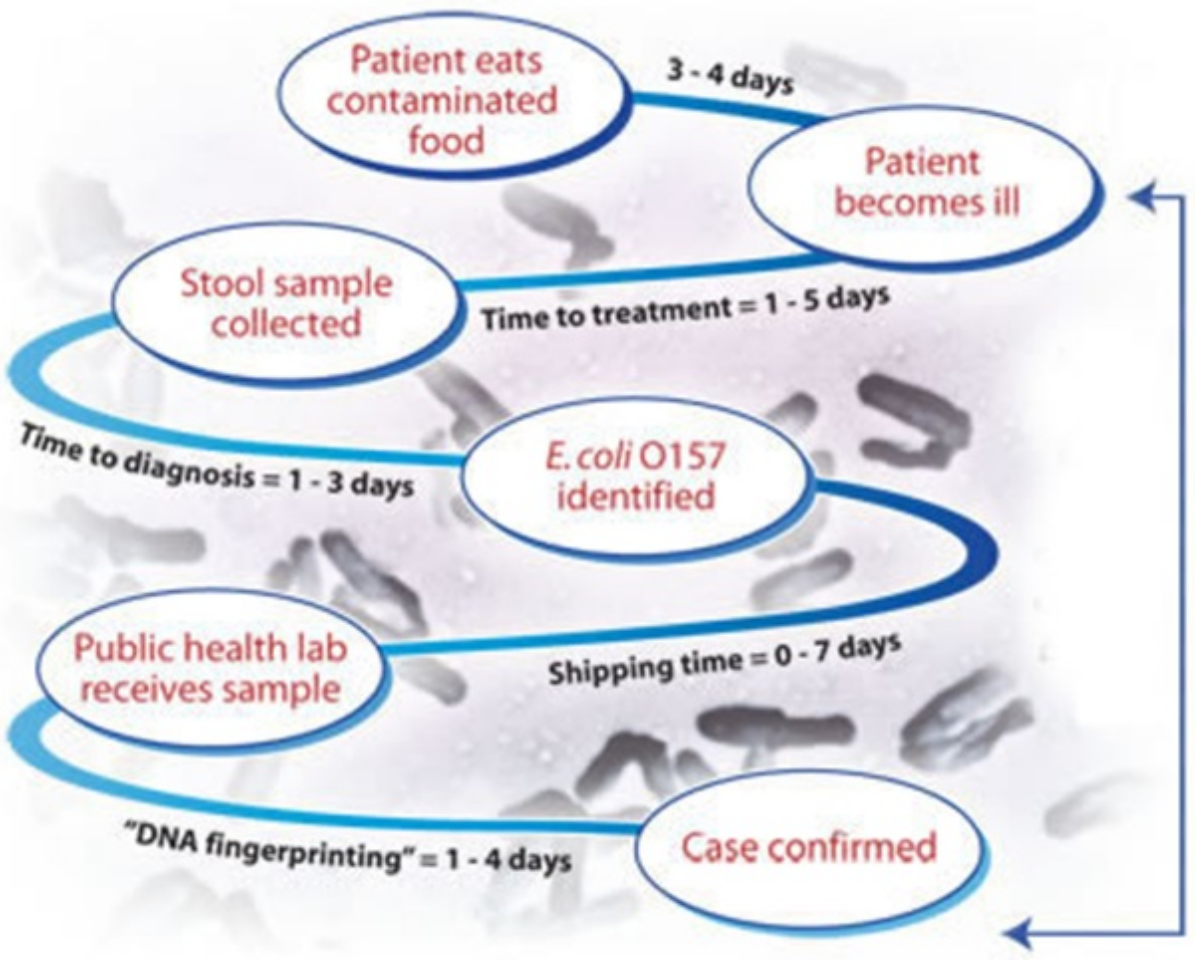

Figure 1.1: CDC Reporting Timeline for E. Coli. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation.

example cantaloupe, which was found to contain listeria in Fall 2011[4], is grown in Colorado, California, North Carolina[5], as well as several other states. Since the growing season differs depending on the location, foodborne illness can occur anytime during the year. In Colorado the listeria outbreak occurred in September, but in North Carolina the outbreak occurred in July. Additionally, the spread of foodborne illness and seasonal influenza could be different. Influenza outbreak tends to affect people over a large area, whereas foodborne illness affects a smaller region (either the city where the food was distributed in a restaurant, or in the few states that the produce was distributed.)

This remainder of this thesis is organized as follows. The next chapter presents some key terms that will be used throughout this paper. Related work is presented in Chapter 3. Framework for the proposed subtopicLDA model is described in Chapter 4. In Chapter 5, a case study on how subtopicLDA performs on determining which tweets relate to foodborne illness events is discussed. Conclusion and future work are discussed in Chapter 6.

[4]http://www.fda.gov/safety/recalls/ucm271879.htm

[5]http://www.fda.gov/NewsEvents/Newsroom/PressAnnouncements/ucm313743.htm

Chapter 2

Key Terms

This section defines more specifically the terms that are used throughout this thesis.

- **Foodborne Illness (FBI)** - This is the technical term for food poisoning. Foodborne illness is an ailment caused by consuming a harmful bacteria in food and have the following symptoms: vomit, fever, diarrhea, and/or upset stomach. If a user seems to have alcohol poisoning, they are not considered as to have FBI. Fever is a common symptom for many ailments, so if this is the only symptom the user is considered to have a different ailment such as the seasonal flu.

- **Patient** - A patient is a Twitter user who is suffering from foodborne illness. If the user has a dependent, such as a child, who are suffering from foodborne illness, then the child is the patient.

- **Tweet** - A tweet is a post on Twitter. Tweets are limited to 140 characters. This includes any urls the user may include in their post.

- **Retweet** - A retweet is a tweet that a different user posted, and the current user is reposting under their username. Retweets start with "RT" followed by the username of the user who posted the original tweet

- **Comment** - A comment is a tweet that starts with @ and the username who the comment is directed towards, followed by the message of the comment.

- **Hashtag** - Label that a user adds to their tweet. This label starts with # followed by the name of the label. The names are not predefined. Any user can invent their own hashtag, and multiple users can use the same hashtag to refer to different topics [34].

- **Profile Location** - Location information that a user places in their profile. This field in a user's profile is a text box, which allows the user to put whatever location they want. The location name can be written in vernacular, as an abbreviation, or be a fictional location.

- **Tweet Location** - Location information that is appended to a tweet. This information is added by the device used to post the tweet, such as a cell phone, which adds the latitude and longitude of where the tweet was posted. Though it provides a more accurate way to identify the location of a user, less than 1% of the tweets have this information.

- **API** - Application Programming Interface. This is an interface that allows easy collection of data from a website. An API allows a website to control who is collecting information, what information is available for collection and how much information can be collected by any individual from that website. For example, Twitter has 2 APIs for accessing information from Twitter, a Search API and a Streaming API[1]. The Streaming API allows for more information to be collected from Twitter than the Search API.

[1] https://dev.twitter.com/

Chapter 3

Related Work

Related work for event detection using topic modeling falls into three categories: research on event detection approaches, review of probabilistic language modeling approaches, and discussion on Latent Dirichlet Allocation and supervised models based on LDA.

3.1 Event Detection

Event detection using social media is a prominent research topic. Previous work has focused on a variety of events, from sporting events to natural disasters, and health-related events. In addition to the type of event, we may categorize previous works according to their temporal and spatial scales. The temporal scale ranges from short events that both start and end within a day to long events that have durations of multiple days and are discussed over weeks and months. Previous studies have suggested that strategies to detect short term events may not be effective for long term events[20]. The spatial scale ranges from local events that are specific to a city or state, to large scale events that affect multiple states, a country, or an entire region. Again, methods developed for local scale events are not always applicable to events that occur over a larger area[20, 47]. Tables 3.1 and **??** summarize the related work on event detection from social media, including the spatial and temporal scale of the monitored events.

Author	Temporal	Spatial	Impact	Algorithm	System
Achrekar et al.[1]	Long	US State	Large	Autoregression with Exogenous Inputs	SNEFT architecture
Corley et al. [8]	Long	Global	Large	Autocorrection function to determine if frequency above random	
Culotta [9]	Long	Global	Large	Multiple Linear Regression, Logistic Regression	
de Quincey & Kostkova [10]	Long	Global	Global	Counting	
Gabrilovich et al.[13]	Medium, Large	Local, Global	Medium	Cluster by topic, KL-divergence distance measure	Newsjunkie
Ginsberg et al. [14]	Long	US State	Large	Counting by Week	Google Flu Trends
Iyengar et al. [20]	Short, Long	Local, Region	Medium	Support Vector Machine, Hidden Markov Model, Cuts algorithm	
Lampos and Cristianini [24]	Long	United Kingdom	Large	Linear Regression	
Marcus et al. [27]	Short	Local	Global	Counting	TwitInfo
Paul & Dredze [31]	Long	Global	Large	Topic Modeling	Ailment Topic Aspect Model
Paul & Dredze [32]	Long	US State	Large	Ailment Topic Aspect Model	
Pozdnoukhov & Kaiser [35]	Short	Ireland	Small, Medium	Markov-modulated, Nonhomogenous Poisson Process	
Rath et al. [38]	Long	US	global	Hidden Markov Model	
Ritterman et al. [39]	Long	Global	Global	Support Vector Regression	

Table 3.1: Summary of Related Work on Event Detection

Table 3.1 (cont'd.)

Author	Temporal	Spatial	Impact	Algorithm	System
Sakaki et al. [41]	Short	Local	Medium	Poisson distribution for time, Kalman filter for location, Support Vector Machine for remove false alarms	Torreter
Seifter et al. [42]	Long	US	US	Google Flu Trends	
Sutton et al. [44]	Medium	Local	Medium	Analysis by Hand	
Vieweg et al. [47]	Long	Local	Local	Counting	

3.1.1 Short Temporal Events

Event detection for short temporal events primarily focuses on finding either the start time of the event, such as an earthquake, or the boundaries of the event, such as a sports game. The discussion of an event starts suddenly during or after the event has occurred (e.g., an earthquake) or the discussion changes verb tense as the event occurs (e.g., a sports game). Short temporal events tend to occur in only one location or a small region, though the event may have popularity worldwide. For example, Marcus et al. mapped the prevalence of different keywords related to their location worldwide [27]. A sports game in one country may receive attention from other countries especially if the teams have followers worldwide. Pozdnoukhov and Kaiser used a Markov-modulated nonhomogenous Poisson process to detect short term local events, such as an Irish music festival, as well as an event that has global interest with a longer duration but is a local event, such as the Harry Potter movie premiere in Ireland [35].

Iyengar et al. detected discrete events with a distinct end time by looking at the verb

phrases in the tweets[20]. They were able to build a support vector machine to classify tweets as before, during, and after the event took place. They used a Hidden Markov Model to predict the window when the event started and finished. For events with a distinct start and end time, such as a sporting event, their approach worked well. For events that do not have an exact start or end time but are continuously discussed with mixed verb tenses, their approach did not perform well.

Earthquake detection is an interesting problem. They can cause damages that are costly to repair and cause numerous injuries and mortalities. Earthquakes happen frequently, making it possible to test detection algorithms in real time. Mendoza et al. presented an approach to detect an earthquake in Santiago, Chile, by counting the number of tweets with the terms "earthquake" or "terremoto" that occur within a 15 minute window[28]. A drawback to their approach is they processed the data after the earthquake had occurred. Mendoza et al. began collecting Twitter data after the first earthquake occurred and analyzed their data at a later time to determine when aftershocks occurred. They found a spike in the number of tweets shortly after each earthquake aftershock occurred. Sakaki et al. uses a similar approach, where they model the number of tweets during an earthquake as following a Poisson distribution [41]. They first classify their tweets containing "earthquake" to determine if they discuss an actual earthquake happening. Then they model the probability of an earthquake based on the number of positively classified tweets from their support vector machine. In their real-time system, called Toretter, they were able to detect an earthquake within a minute after the earthquake occurred, while the national broadcast reported earthquakes within 6 minutes of their occurrence [41].

3.1.2 Long Temporal Events

In long temporal events, such as disease outbreaks, the discussion spans weeks to months where there is no exact time when the discussion changes verb tense and there is not a sudden increase in the frequency of tweets within a small window of time, like a 15-minute window. Long temporal events can affect either one region or multiple regions. For disease outbreak detection, local regions, like US states, tend to be a focus of research, since the Center for Disease Control (CDC) reports some diseases, like influenza, based on the region where an outbreak occurred. These reports from the CDC are often used as the baseline for influenza-like illness outbreak detection [9, 14, 32]. The same illness sometimes affects multiple states and may be part of the same outbreak, especially for influenza. Diseases that previous research has studied include influenza, H1N1, and lyme disease.

Natural disasters such as hurricanes and wildfires also have properties long temporal events. One characteristic that contributes to natural disasters having a long temporal duration is that the disaster moves from one region to another. Hurricanes are a prime example of this. A hurricane will affect different states over the course of a week. A hurricane may affect one state, like North Carolina, for only a couple of days, but the hurricane itself remains active as it continues to move inland. Iyengar et al. found that this property of hurricanes makes determining the event start and end time into a harder problem because the verb tense became mixed as the hurricane moved [20]. Some users that already experienced the hurricane used past tense, while users in the middle of the hurricane would mix past and present tense. Wildfires have this same property on a smaller scale; they tend to move from neighborhood to neighborhood [47].

Influenza-like Illnesses (ILI) is one of the most popular diseases used for surveillance using

social media. Researchers have focused on three internet sources: Google search queries, blogs, and tweets from Twitter. Ginsberg et al. calculated the number of search queries per week related to influenza-like illness (ILI) [14]. From this count, they used linear regression to predict the probability of an epidemic. The Center for Disease Control (CDC) publishes weekly reports of the percentage of doctor visits are related to ILI [5]. The output of the model of Ginsberg et al. found a high correlation (0.9) to the CDC weekly reports, showing an epidemic 1-2 weeks before the CDC report showed an epidemic. Ginsberg et al noticed a drawback to their approach, that all search queries are treated as relevant to the event if they matched some keywords. This would lead to several search queries being considered as relevant to the event when they are unrelated to the event.

Since Google search queries are owned by Google, most researchers focus on data they can collect for free, mainly blogs and Twitter. Corley et al. used blogs from Spinn3r to detect influenza trends [8]. More researchers have focused on gathering data from Twitter [1, 9, 24, 32]. Twitter has more users than Spinn3r [43, 46], and the short length of tweets restricts the number of topics present in a post. Most researchers who use Twitter for influenza surveillance try to model or predict the weekly ILI percentage posted by the CDC by using linear regression. One of the main differences among the approaches is whether they start with a classifier to select tweets that are most relevant. Tweets where the user is claiming to be ill, either by saying the disease they have or by listing their symptoms, is more useful than tweets that do not relate to influenza. Twitter users often use symptom terms to emphasize a point, such as:

> "U know that ability where you re able to sense if someone is engaged in a conversation or not? Some people dont have it. cough mymom cough" [9]

The use of a classifier can help identify relevant tweets from Twitter. To help remove noisy irrelevant tweets, Culotta built a logistic regression model to determine if a tweet is relevant or not [9]. Paul and Dredze used a support vector machine (SVM) to reduce the number of tweets to only those that are health related [32]. Other researchers chose not to build an initial classifier to reduce the number of tweets [1, 24].

Instead, these researchers focus on collecting tweets that either mention the flu or symptoms of the flu. Lampos and Cristianini weigh tweets by how many symptom words it contains, which they call the tweet's flu-score [24]. This approach is susceptible to false alarms when two tweets mention the same relevant terms, but one is clearly from an ill user and the other is from a healthy user. For example:

- Healthy: "*Came Home From School.... I think I have a bad case of the bieber fever*"
- Ill: "*went home. have a fever. #beware*"

Simply counting the number of search terms that occur in a tweet will cause both tweets to have the same flu-score, unless some terms have negative weights, such as "bieber." In this case, flu-score alone does not completely reduce the noise of Twitter.

Achrekar et al. limits the number of unique users by removing tweets that appear within an elapsed time from the user's first influenza-related tweet [1]. They chose an elapse time of 1 week, so any tweet with syndrome terms by a particular user is removed if it occurs within a week of a previous tweet that is included in the dataset that mentions symptoms of influenza. To help reduce how much Twitter noise affects their model, Achrekar et al. include the previous week's CDC ILI percent along with the number of unique users that had flu related tweets, to predict the CDC ILI percentage for the current week.

Other previous research try different approaches to influenza epidemic detection. Rath

et al. uses Hidden Markov Model to detect influenza epidemics by comparing the number of tweets to a Gaussian threshold [38]. They claim an epidemic occurs when the number of tweets exceed the threshold.

Additional illnesses studied in previous research are Lyme disease and swine flu. Seifter et al. uses Google Search Trends to analyze the presense of Lyme disease epidemics [42]. Ritterman et al. use Twitter and prediction markets to predict market prices based on public opinion (from Twitter) of a swine flu pandemic [39]. De Quincey and Kostkova analyze the potential of Twitter to detect a swine flu epidemic based on the number of tweets per day that contain the phrase "swine flu." [10].

Some natural disasters can also be categorized as long temporal events. Iyengar et al. looked at temporal boundary detection for hurricanes to know when it hit each region [20]. They found that the movement of hurricanes from one region to another made determining the start and end time of the hurricane into a challenging problem when looking only at the verb tense of the tweet content. Vieweg et al. use Twitter to monitor situation awareness and the progression of wildfires in Oklahoma and flooding of the Red River in North Dakota [47]. They do not apply any techniques to determine when the event began or ended, except counting tweets by day. They analyze how Twitter is being used to communicate information during these natural disasters.

Previous research on foodborne illness has discussed the potential of social media to monitor food safety, but these works did not test their hypotheses on real data from social media. Newkirk et al. presented a literature review of social media being used for disease surveillance and demonstrate where in the CDC reporting timeline (Figure 1.1) the use of social media can assist with outbreak detection. Astley described how Twitter can be used to alert the public of food recalls [2]. Dixon discussed the benefits and risks of electronic

Author	Topic	Conclusions
Astley [2]	Food safety, Twitter use to prevent potential lethal outbreak	Twitter can help with surveillance and help disseminating information of an existing outbreak
Dixon [11]	Benefis and Drawbacks of social media with foodborne illness	Speculation on the next big outbreak of foodborne illness and the drawbacks of social - media causing false alarms
Newkirk et al. [30]	Food safety and food terrorism surveillance system	Potential to use Twitter to reduce the time from the patients' onset of symptoms to when the data can be imported into the Food and Drug Administration's surveillance system to detect a foodborne illness outbreak

Table 3.2: Related Work with Focus on Literature Review

media in food safety [11]. The benefits include assisting with outbreak detection, while a risk includes erroneous claims that some food is contaminated when it is not.

3.2 Probabilistic Topic Models

In this thesis, a probabilistic topic modeling approach is developed for event monitoring from social media. Probabilistic topic models are designed to model how words are generated in documents within a text corpus. Examples of such models include probabilistic Latent Semantic Indexing (pLSI), Latent Dirichlet Allocation (LDA), and author-topic models. All of these models are derived from a basic language model, which models words and documents as probabilities [29, 33]. Specifically, a document is defined as the joint probability of all of its words occurring. A common approach to deriving the probability of a document is to count the number of times the same set of terms appear in the dataset. Since two documents rarely contain the exact same set of terms, different assumptions have been used to simplify the derivation of the joint probability of all the words occurring in the document. The most common assumption is the bag of words (BOW) assumption, which assumes that each word

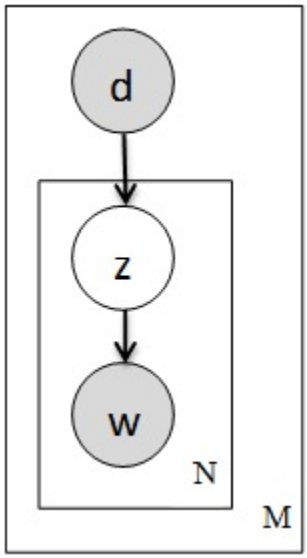

Figure 3.1: Probabilistic Latent Semantic Indexing

is independent of all other words, given the document. The probability of a word occurring in a document $P(w_i)$ can easily be derived using the maximum likelihood method:

$$P(w_i) = \frac{\text{count } w_i \text{ in dataset}}{\text{count all words in dataset}}.$$

The probability of a document is the product of the probability of each term in the document $P(w_1, \ldots, w_N) = \prod_{i=1}^{N} p(w_i)$.

This model is an over simplification of how documents are generated, so more advanced topic models have been proposed to better represent how documents are actually derived. Probabilistic Latent Semantic Indexing (PLSI), also known as Probabilistic Latent Semantic Analysis (PLSA), builds on the basic language model by modeling the probability of each word as conditioned on the topic that generates that word [17]. Figure 3.1 provides a graphical representation of PLSI. The topic is a latent (hidden) variable that is dependent on a prior β. Topics, denoted as z, are called aspects of the model. The model as a whole is an aspect model. Instead of assigning a document to a single topic, a document becomes a

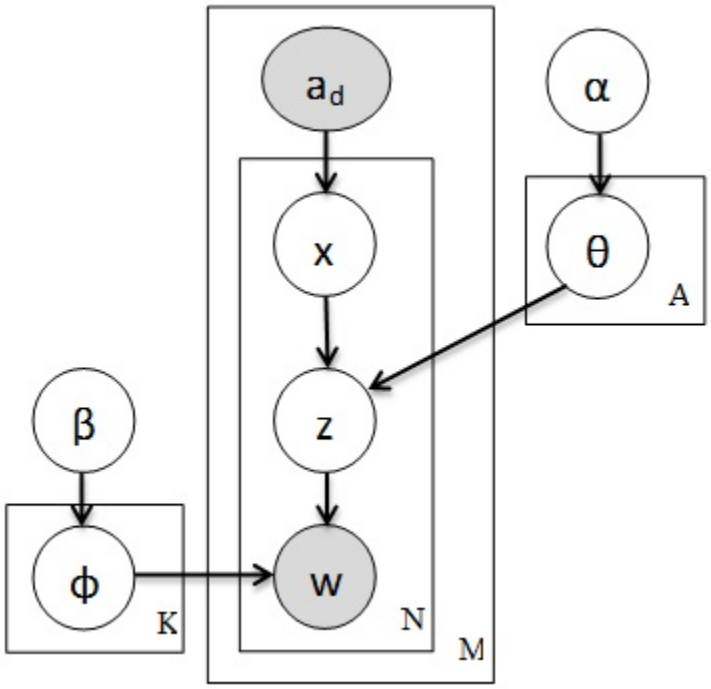

Figure 3.2: Author-Topic Model

mixture of its aspects, or topics, based on the latent aspects of its words [17]. Documents are generated following a three step process. The first step is to determine the number of words in the document N_m. Next, for each word position in the document, a topic is randomly chosen. Finally, a word is chosen based on the word distribution for that topic.

Latent Dirichlet Allocation(LDA) builds on pLSI [4, 16]. In pLSI, the probability for each topic is the same regardless of which document the words and the topics appear in. Similar to pLSI, LDA has one set of probabilities, denoted as ϕ, that model the probability of each term given the topic that generated it, and this probability is conditioned on a prior β. LDA then adds a second probability θ which denotes the probability distribution of the topics given the document. This probability is also conditioned on a prior, which is denoted by α. The graphical model for LDA is presented in Figure 3.3. In LDA, each document does not have to follow the same topic distribution as all other documents. As a consequence,

there is variation in the topics generated for each document. Latent Dirichlet Allocation is further described in detail in the next section.

Another extension of Latent Dirichlet Allocation is author-topic models. Similar to LDA, author-topic models sample a set of topics for each word position in a document based on the topic distribution for that document, and then sample the words from the word distribution for each selected topic. Author-topic models extend LDA to deal with documents written by a set of authors [40]. Figure 3.2 demonstrates this process. For each word position in the document, one of the authors is selected to have written that word. The author decides the topic of the word from the document's topic distribution, and then chooses the word based on the word distribution of the topic. LDA can be considered as a simplified version of the author-topic model assuming every document has the same author. However, since each tweet is written by a single author, it is sufficient to consider LDA instead of author-topic models for event detection in Twitter data.

3.3 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA), first proposed by Blei et al., is an unsupervised probabilistic topic model to estimate the probability of observing a document given the underlying topics of the document [4]. The generative process for LDA is similar to pLSI, where the first step is to select the number of words that will appear in the document. In the second step of the generative process, a topic is randomly selected for each word position. Unlike pLSI, where the probability distribution of the topics is assumed to be the same for all documents, each document in LDA has its own topic distribution. Once the topic is generated, a word is randomly chosen from the word probability distribution for that topic. Figure 3.3 shows the

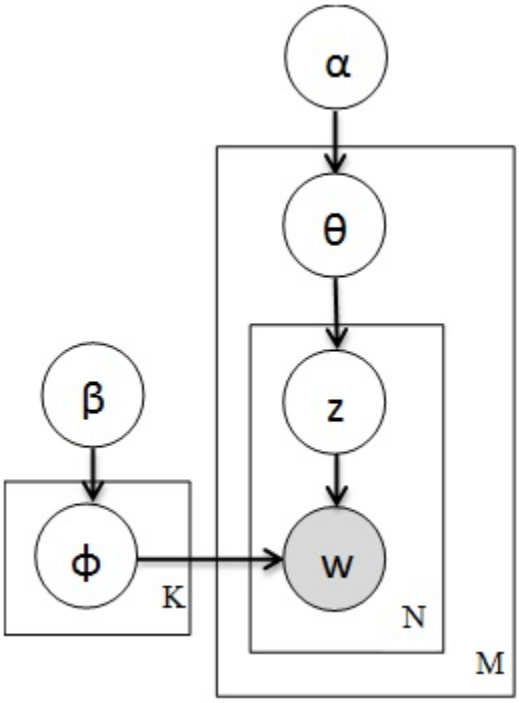

Figure 3.3: Latent Dirichlet Allocation

graphical representation of the LDA model. A pseudo-code of the algorithm is given below.

Algorithm 1 Generative Process for LDA

for each topic $k = 1 \rightarrow K$ **do**
  sample $\phi_k \sim Dir(\beta)$
end for
for each document $m = 1 \rightarrow M$ **do**
  sample $\theta_m \sim Dir(\alpha)$
  sample document length $N_m \sim Poiss(\xi)$
  for each word $n = 1 \rightarrow N_m$ **do**
    sample topic $z_{m,n} \sim Mult(\theta_m)$
    sample word $w_{m,n} \sim Mult(\phi_{z_{m,n}})$
  end for
end for

In Latent Dirichlet Allocation(LDA), each document m has its own topic distribution, θ_m, generated from a Dirichlet prior with hyper-parameter α. The number of words for that document is also randomly chosen from a Poisson distribution. For each word position in the document, a topic z is randomly chosen from a multinomial distribution parameterized

by the mixture of topics for that document θ_m. Once the topic is known, a word is drawn from the word distribution, ϕ_z, for that topic. Drawing a word from the topic distribution is inherited from pLSI. Drawing the topic distribution for each document is unique to LDA (and models built upon LDA). More details on the derivation of LDA can be found in [4, 16, 6].

Author	Theme of Model	Model Based On	Name of Their Algorithm
Hong and Davison [18]	User Modeling	LDA	
Hong et al. [19]	Model Topics of Multiple Text Streams	LDA	
Kinsella et al. [21]	Location	Basic language model[33]	
Paul and Dredze [31]	Ailment Mentions	LDA	Ailment Topic Aspect Model (ATAM)
Paul and Dredze [32]	Ailment Mentions by State	ATAM	
Quercia et al. [36]	Model User Topics	LabeledLDA	TweetLDA
Teevan et al. [45]	Search Queries	LDA	
Wang et al. [48]	Predict Topic of Future Tweet based on Topics Only of Previous Tweet	LDA	TM-LDA
Weng et al. [49]	Topic-sensitive Influential Users	LDA	TwitterRank
Zhao et al. [51]	Topic modeling	LDA	TwitterLDA
Zhao et al. [50]	Keyphrase Extraction	TwitterLDA	

Table 3.3: Uses of LDA in Twitter

Several approaches to incorporate labeled data into the LDA framework have been developed. Blei and McAulliffe proposed supervised LDA, which uses the topic distribution of a document to generate the label of that document [3]. This is advantageous for regression problems when the labels are continuous. Labeled LDA, by Ramage et al, restricts the set of topics to the set of labels that each document contains [37] In their model, a document can have multiple labels. Each label maps to one topic. Li and Perona develop a supervised LDA model that they call Theme Model 1 [25]. This model has the most similarity with

the model presented in this paper. They generate a topic distribution for each document conditioned on a set of priors, that are dependent on the class labels. In the proposed model of this thesis, the topic distributions are generated for each class instead of each document. All class distributions are conditioned on the same priors. However Theme Model 1 does not generate the topic distribution per class and the word distribution per topic from some prior. Lakshminarayanan and Raich call Theme Model 1 as supervised LDA, which they use for image classification [23]. DiscLDA, proposed by Lacoste-Julien et al., is another model similar to the model in this paper. In DiscLDA, they use a transformation matrix to modify the topic distribution of a document, θ, into a mixture of Dirichlet distributions [22]. This transformation depends on the document's label. Similar to this research, DiscLDA also uses θ and the class label to choose the topic of each word in a document.

While most previous research on topic modeling applies models to a variety of domains, some researchers focused on applying LDA or variations of LDA to Twitter data. Pozdnoukhov and Kaiser use topic modeling to detect discrete events from entertainment, such as a music festival [35]. They look at the location and time of the Twitter stream to model the variation of topics over time and by region of Ireland. Hong and Davison apply LDA to tweets from the same user to generate a topic profile for each user [18]. Hong et al. uses two text streams, Twitter and Yahoo News, to model that some topics come from a distribution of common topics that are shared in both streams, while other topics come from a distribution of topics unique to that text stream [19]. Similar to the work by Hong et al., Zhao et al. present Twitter LDA, which explains that a word can be generated from one of two distributions [51]. One distribution of words is created from the distribution of topics, like in regular LDA, while the second possible distribution is that the word is drawn from a set of background words. Zhao et al. then uses this model for keyphrase extraction to model

the popular topics in Twitter [50]. Kinsella et al. uses topic modeling to predict a user's location [21]. They build a topic model to model each location based on the city or state that the tweets come from to determine which terms are unique or more common to that area. They use this model to best determine where a user is from based on the content of their last 100 tweets. Teevan et al. use LDA to model search queries [45]. Topic modeling can assist in discovering which search queries describe the same topic, which Teevan et al. use to compare Twitter search queries with web search queries [45].

Three other research groups present variations on LDA to apply to Twitter. Quercia et al. present TweetLDA, which, similar to the work by Hong and Davison, tries to assign a topic distribution to each user based on the tweets of the user [36]. Wang et al. present TM-LDA which uses the topic assignment from LDA to learn a transition matrix of topics to see which topics tend to be followed by which other topics [48]. Their goal is to predict the topics of a future tweet based solely on the topics of the previous tweet by the same user. Weng et al. built TwitterRank, which uses topic modeling to find the most topic-sensitive influential users, the trend-setters that are the first to discuss a new topic, and their followers then discuss the same topic [49].

Paul and Dredze present a model most relevant to this research, which they call the Ailment Topic Aspect Model (ATAM) [31, 32]. Instead of solely concentrating on one ailment, they build a topic aspect model to model 20 ailments in Twitter. These ailments are modeled as a latent variable. Paul and Dredze needed to manually label each ailment by hand, based on the words for each ailments. One ailment that they discover from the terms assigned to it is influenza. To only model relevant tweets, Paul and Dredze initially label the tweets as sick, health, or ambiguous, which becomes the aspect of thier model (an observed label for each tweet) [31]. Paul and Dredze also label tweets as unrelated or not english as part of

a negative class to discard irrelevant tweets. They then trained a SVM to find only health related tweets. To help train their topic aspect model, Paul and Drezde used articles from WebMD to discover the symptoms and treatments for each ailment [32]. To compare their model to others for influenza surveillance, they counted the number of tweets assigned to the flu ailment by ATAM, and normalized over the total number of tweets that week. Their results correlated highly (0.934) to CDC's influenza reports [31].

In this chapter, event detection and topic modeling related work was presented. Some previous research has looked at event detection using Twitter, however only Paul and Dredze [32, 31] used topic modeling to do event detection in their Ailment Topic Aspect Model. In their approach, the ailment was a hidden variable. In the next chapter, subtopicLDA framework is proposed. In contrast to the Ailment Topic Aspect Model, subtopicLDA models the ailment as an observed variable, so human annotation is not needed for a post-processing step.

Chapter 4

Proposed Framework

In this chapter, the framework for subtopicLDA is proposed. SubtopicLDA is a probabilistic generative model, that is an extension of Latent Dirichlet Allocation, and it is designed for event detection using social media. Event detection using social media has three key challenges; determining which posts are relevant to the event, detecting when the event started, and determining where the event occurred. SubtopicLDA is proposed as a solution to the first challenge, determining which content is relevant. Social media platforms, like Twitter, contains short, noisy posts, called tweets. Using the terms directly to build a classifier produces poor results. Topic modeling, however, models the underlying themes that the words discuss, which helps to remove some of the noise present in social media.

Latent Dirichlet Allocation (LDA) is a popular topic model that models the topics both in terms of the individual words as well as of the documents as a whole. LDA has two drawbacks for event detection. It is unsupervised, so specifying which event or topic LDA should focus on is not an option. Label information needs to be incorporated into LDA, which previous work has done, as described in the previous chapter. SubtopicLDA incorporates label information with a similar modification to LDA as these other supervised models by including an observed variable for the label information. The second drawback to LDA is that it assigns a topic distribution to each document, so the model is tends to overfit the training data. Storing the topic distribution for each document also takes a large amount of memory.

Model	Topic Distribution Matrix Dimensions
Latent Dirichlet Allocation (LDA)	M x K
Supervised LDA	M x K
LabeledLDA	M x K
discLDA	(M x L) + (K x L)
Supervised Theme Model 1	(M x K) + (C x K)
SubtopicLDA	C x K

Table 4.1: Storage requirements for topic distribution of Latent Dirichlet Allocation and its supervised extensions

If there are M documents in the training set and K topics, then the topic distribution is stored in an M x K matrix that must estimated for LDA. SubtopicLDA solves this drawback by providing a generalized topic distribution for each class instead of for each document, which takes the mean topic distribution of all documents of the same class. This reduces how much the model overfits the training data. The topic distribution information for LDA is stored in a M x K matrix, where M is the number of documents and K is the number of topics. In subtopicLDA, the topic distribution information is stored in a C x K matrix, where C is the number of classes. Since the number of classes is typically significantly smaller than the number of documents in the training data, subtopicLDA requires less memory and has fewer parameters to estimate than other LDA-based models. Table 4.1 demonstrates the storage requirements for the topic distribution for each model.

4.1 Generative Model

A probabilistic generative model describes the non-deterministic process of how a set of observations are generated. Such models often require a set of hidden states to explain the underlying unobserved information. In Latent Dirichlet Allocation (LDA), the observations are the words that appear in each document and the hidden states are the topics for each word

drawn from the topic distribution for each document. In the generative model for LDA, each document has its own topic distribution, drawn from a Dirichlet prior distribution. For each word in the document, a topic is randomly selected from the document's topic distribution. Each topic has a word probability distribution. Once a topic is drawn, then a word is randomly chosen from the word distribution for that topic. SubtopicLDA, on the other hand, includes the label for each document as another observed variable. Unlike existing supervised LDA methods, the class label determines the topic distribution of a document. In other words, the topic distribution of a document is the same as the topic distribution for the class the document belongs. Once the topic distribution has been generated, the topics and words assigned to each word position in a document follow the same approach as regular LDA. Even though two documents from the same class have the same topic distribution, the topics assigned to each word position in a document may vary between documents of the same class even if the words in the two documents are identical. The generative model for subtopicLDA is described in Algorithm 2

Algorithm 2 Generative Model for SubtopicLDA

for each topic $k = 1 \rightarrow K$ **do**
  sample $\phi_k \sim Dir(\beta)$
end for
for each class $c = 1 \rightarrow C$ **do**
  sample $\theta_c \sim Dir(\alpha)$
end for
for each document $m = 1 \rightarrow M$ **do**
  sample label $y_m \sim Bernoulli(\eta) \in [1, C]$
  sample document length $N_m \sim Poiss(\xi)$
  for each word $n = 1 \rightarrow N_m$ **do**
    sample topic $z_{m,n} \sim Mult(\theta_{y_m})$
    sample word $w_{m,n} \sim Mult(\phi_{z_{m,n}})$
  end for
end for

SubtopicLDA shares many similarities with LDA. The topic distributions, represented

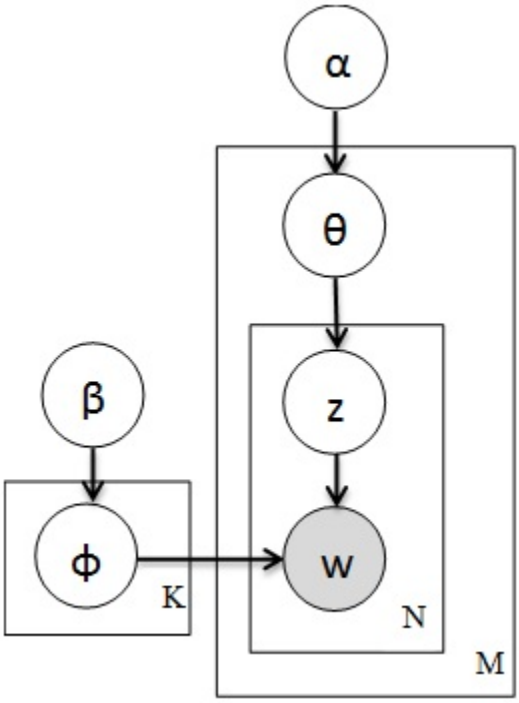

Figure 4.1: Latent Dirichlet Allocation

by θ, are the major variation, which are generated for each class instead of each document. The other addition to subtopicLDA is the label information, y_m, that is randomly generated from a Bernoulli distribution. Figures 4.1 and 4.2 demonstrate the difference between their graphical models. The topic distributions, θ, are moved to their own plate, to represent that only C vectors are estimated for θ instead of M vectors, where C is the number of classes and M is the number of documents. Shaded variables are observed, and variables that only have outgoing edges are the hyper-parameters of subtopicLDA. Since the class label y is observed, its prior η is primarily used for predicting the class of previously unseen documents.

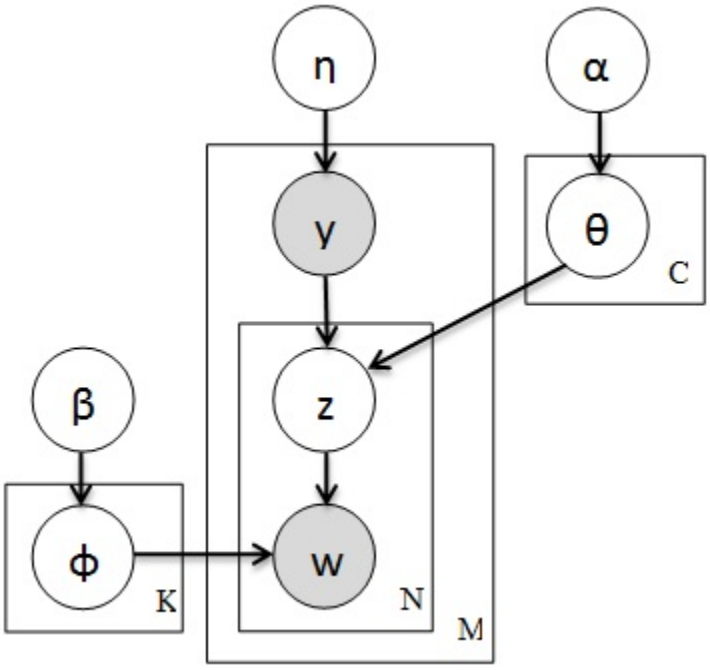

Figure 4.2: Subtopic LDA

4.2 Inference

To calculate the hidden variables of this generative model, first the complete probability needs to be computed, $p(w, y, z, \Phi, \Theta|\alpha, \beta, \eta)$.

$$
\begin{aligned}
&p(w, y, z, \Phi, \Theta|\alpha, \beta, \eta) = p(\Theta|\alpha)p(\Phi|\beta)p(z|y, \Theta)p(w|\Phi, z)p(y|\eta) \\
&= \prod_{m=1}^{M} p(y_m|\eta) \prod_{c=1}^{C} p(\theta_c|\alpha) \prod_{k=1}^{K} p(\phi_k|\beta) \prod_{m=1}^{M} \prod_{n=1}^{N_m} p(z_{m,n}|\theta, y_m) \prod_{m=1}^{M} \prod_{n=1}^{N_m} p(w_{m,n}|\phi, z_{m,n}) \\
&= \prod_{m=1}^{M} p(y_m|\eta) \prod_{c=1}^{C} p(\theta_c|\alpha) \prod_{k=1}^{K} p(\phi_k|\beta) \prod_{m=1}^{M} \prod_{n=1}^{N_m} p(z_{m,n}|\theta_{y_m}) \prod_{m=1}^{M} \prod_{n=1}^{N_m} p(w_{m,n}|\phi_{z_{m,n}}) \quad (4.1)
\end{aligned}
$$

n_c	Number of documents in class c
$n_{m,k}$	Total number of times topic k is assigned to a word in document m
$n_{c,k}$	Total number of times topic k is assigned in any document from class c
$n_{k,m,v}$	Total number of times topic k is assigned to term v in document m
$n_{k,v}$	Total number of times topic k is assigned to term v in all documents

Table 4.2: Table of Count Variables

$n_{c,k}$	$= \sum_{m=1}^{M} n_{m,k} * \delta(y_m = c)$
$n_{k,v}$	$= \sum_{m=1}^{M} n_{k,m,v}$

Table 4.3: How to calculate counts that are dependant on other counts

$$p(y|\eta) \sim Mult(y|\eta) = \prod_{m=1}^{M} \prod_{c=1}^{C} \eta_c^{\delta(y=c)} = \prod_{c=1}^{C} \eta_c^{nc} \tag{4.2}$$

$$p(\theta_c|\alpha) \sim Dir(\theta_c|\alpha) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{c,k}^{\alpha_k - 1} \tag{4.3}$$

$$p(\phi_k|\beta) \sim Dir(\phi_k|\beta) = \frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\prod_{v=1}^{V} \Gamma(\beta_v)} \prod_{v=1}^{V} \phi_{k,v}^{\beta_v - 1} \tag{4.4}$$

$$p(z_{m,n}|\theta_{y_m}) \sim Mult(z_{m,n}|\theta_{y_m}) = \prod_{m=1}^{M} \prod_{k=1}^{K} \theta_{y_m,k}^{n_{m,k}} = \prod_{c=1}^{C} \prod_{k=1}^{K} \theta_{c,k}^{n_{c,k}} = \theta_{c,k}^{n_{c,k}} \tag{4.5}$$

$$p(w_{m,n}|\phi_{z_{m,n},w_{m,n}}) \sim Mult(w_{m,n}|\phi_{z_{m,n}}) = \prod_{m=1}^{M} \prod_{n=1}^{N_m} \prod_{k=1}^{K} p(w_{m,n}|\phi_{z_{m,n}=k,w_{m,n}=v})$$

$$= \prod_{m=1}^{M} \prod_{v=1}^{V} \prod_{k=1}^{K} \phi k, v^{n_{k,m,v}} = \prod_{k=1}^{K} \prod_{v=1}^{V} \phi_{k,v}^{n_{k,v}} \tag{4.6}$$

Where

$$\delta(y = c) = \begin{cases} 1 & \text{if } y = c \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

Filling in the probabilities from the distributions, the resulting probability:

$$
\begin{aligned}
&p(w, y, z, \Phi, \Theta | \alpha, \beta, \eta) \\
&= \prod_{c=1}^{C} p(y_m|\eta)^{n_c} \prod_{c=1}^{C} p(\theta_c|\alpha) \prod_{k=1}^{K} p(\phi_k|\beta) \prod_{m=1}^{M} \prod_{n=1}^{N_m} p(z_{m,n}|\theta_{y_m}) \prod_{m=1}^{M} \prod_{n=1}^{N_m} p(w_{m,n}|\phi_{z_{m,n}}) \\
&= \prod_{c=1}^{C} \eta_c^{n_c} (\prod_{c=1}^{C} \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{c,k}^{\alpha_k - 1}) (\prod_{k=1}^{K} \frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\prod_{v=1}^{V} \Gamma(\beta_v)} \prod_{v=1}^{V} \phi_{k,v}^{\beta_v - 1}) (\prod_{k=1}^{K} \prod_{c=1}^{C} \theta_{c,k}^{n_{c,k}} \prod_{v=1}^{V} \phi_{k,v}^{n_{k,v}}) \\
&= \prod_{c=1}^{C} \eta_c^{n_c} (\prod_{c=1}^{C} \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{c,k}^{n_{c,k} + \alpha_k - 1}) (\prod_{k=1}^{K} \frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\prod_{v=1}^{V} \Gamma(\beta_v)} \prod_{v=1}^{V} \phi_{k,v}^{n_{k,v} + \beta_v - 1})
\end{aligned} \tag{4.8}
$$

Gibbs sampling is used to estimate the latent variables in this model. In Gibbs sampling, each item of a vector of a variable is sampled whole all other elements of the vector remain constant and using those elements to sample the unknown element. This can be seen in Equation 4.9

$$
p(z_i | z_{\neg i}, x) \tag{4.9}
$$

In subtopicLDA, there are three latent variables to sample. To simplify the calculation, collapsed Gibbs sampling is used. In collapsed Gibbs sampling, some variables are integrated out to simplify sampling. In this case, θ and ϕ are integrated out, leaving only z to be

sampled.

$$
\begin{aligned}
& p(w, y, z|\alpha, \beta, \eta) \\
&= \int\int p(\Theta|\alpha)p(\Phi|\beta)p(z|y,\Theta)p(w|\Phi,z)p(y|\eta)d\Theta d\Phi \\
&= p(y|\eta)\int p(\Theta|\alpha)p(z|y,\Theta)d\Theta \int p(\Phi|\beta)p(w|\Phi,z)d\Phi \\
&= \prod_{c=1}^{C}\eta_c^{n_c}\int(\prod_{c=1}^{C}\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}\theta_{c,k}^{n_{c,k}+\alpha_k-1})d\theta_c\int(\prod_{k=1}^{K}\frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)}\prod_{v=1}^{V}\phi_{k,v}^{n_{k,v}+\beta_v-1})d\phi_k \\
&= \prod_{c=1}^{C}\eta_c^{n_c}(\prod_{c=1}^{C}\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\int\prod_{k=1}^{K}\theta_{c,k}^{n_{c,k}+\alpha_k-1}d\theta_c)(\prod_{k=1}^{K}\frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)}\int\prod_{v=1}^{V}\phi_{k,v}^{n_{k,v}+\beta_v-1}d\phi_k) \\
&= \prod_{c=1}^{C}(\eta_c^{n_c}\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\frac{\prod_{k=1}^{K}\Gamma(n_{c,k}+\alpha_k)}{\Gamma(\sum_{k=1}^{K}n_{c,k}+\alpha_k)})\times\prod_{k=1}^{K}(\frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)}\frac{\prod_{v=1}^{V}\Gamma(n_{k,v}+\beta_v)}{\Gamma(\sum_{v=1}^{V}n_{k,v}+\beta_v)})
\end{aligned}
\tag{4.10}
$$

The next step is to convert the probability $p(w, y, z|\alpha, \beta, \eta)$ into Gibbs sampling form, i.e. $p(z_i|z_{\neg i}, w, y, \alpha, \beta, \eta)$.

$$
\begin{aligned}
p(z_i|z_{\neg i}, w, y, \alpha, \beta, \eta) &= \frac{p(w, y, z_i, z_{\neg i}|\alpha, \beta, \eta)}{p(w, y, z_{\neg i}|\alpha, \beta, \eta)} \\
&\propto p(w, y, z_i, z_{\neg i}|\alpha, \beta, \eta)
\end{aligned}
\tag{4.11}
$$

Because the denominator does not depend on z_i,

$$
= \prod_{c=1}^{C}(\eta_c^{n_c}\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\frac{\prod_{k=1}^{K}\Gamma(n_{c,k}+\alpha_k)}{\Gamma(\sum_{k=1}^{K}n_{c,k}+\alpha_k)})\times\prod_{k=1}^{K}(\frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)}\frac{\prod_{v=1}^{V}\Gamma(n_{k,v}+\beta_v)}{\Gamma(\sum_{v=1}^{V}n_{k,v}+\beta_v)}).
\tag{4.12}
$$

Algorithm 3 Collapsed Gibbs Sampling Algorithm

Initializaton ▷ zero all count variables $n_{c,k}$, $n_{k,v}$, n_c, n_k
for all documents $m \in [1, M]$ **do**
  for all words $w \in [1, N_m]$ **do**
    sample topic index $z_{m,n} = k \sim Mult(1/K)$
    increment count variables $(n_{c,k}, n_{k,v}, n_c, n_k)$ by 1, for the resulting sampled k
  end for
end for ▷ Gibbs Sampling ▷ Burn-in Period
for Iteration iter$\in [1, EndBurnIteration]$ **do**
  for all documents $m \in [1, M]$ **do**
    for all words $w \in [1, N_m]$ **do**
      find the current assignment k for $z_{m,w}$
      decrement counts $(n_{c,k}, n_{k,v}, n_c, n_k)$ by 1, for the current assignment k
      sample new $k \sim p(z_{m,w}|z_{\neg(m,w)}, w, y, \alpha, \beta, \eta)$ from equation 4.21
      increment count variables $(n_{c,k}, n_{k,v}, n_c, n_k)$ by 1, for the new k
    end for
  end for
end for ▷ Sampling Until Convergence
while Not Converge **do**
  for all documents $m \in [1, M]$ **do**
    for all words $w \in [1, N_m]$ **do**
      find the current assignment k for $z_{m,w}$
      decrement counts $(n_{c,k}, n_{k,v}, n_c, n_k)$ by 1, for the current assignment k
      sample new $k \sim p(z_{m,w}|z_{\neg(m,w)}, w, y, \alpha, \beta, \eta)$ from equation 4.21
      increment count variables $(n_{c,k}, n_{k,v}, n_c, n_k)$ by 1, for the new k
    end for
  end for
  calculate expected value of Φ
  calculate expected value of Θ
  check convergence of Φ and Θ compared to previous iteration's Φ and Θ
end while

Since $\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}$ and $\frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)}$ are constants, the joint probability can be rewritten as

$$p(w,y,z|\alpha,\beta,\eta) \propto \prod_{c=1}^{C}(\eta_c^{n_c}\frac{\prod_{k=1}^{K}\Gamma(n_{c,k}+\alpha_k)}{\Gamma(\sum_{k=1}^{K}n_{c,k}+\alpha_k)}) \times \prod_{k=1}^{K}(\frac{\prod_{v=1}^{V}\Gamma(n_{k,v}+\beta_v)}{\Gamma(\sum_{v=1}^{V}n_{k,v}+\beta_v)}) \tag{4.13}$$

The sample is one cell in the Z matrix, so z_i can be represented as $z_{a,b}$, sampling for document a, word b. Rewriting the above equation in terms of a and b:

$$\begin{aligned} = (\prod_{c\neq y_a}^{C}\eta_c^{n_c})\eta_{y_a}^{n_{y_a}}(\prod_{c\neq y_a}^{C}\frac{\prod_{k=1}^{K}\Gamma(n_{c,k}+\alpha_k)}{\Gamma(\sum_{k=1}^{K}n_{c,k}+\alpha_k)})\frac{\prod_{k=1}^{K}\Gamma(n_{y_a,k}+\alpha_k)}{\Gamma(\sum_{k=1}^{K}n_{y_a,k}+\alpha_k)}\times \\ \prod_{k=1}^{K}\frac{\prod_{v\neq v_{a,b}}^{V}\Gamma(n_{k,v}+\beta_v)\Gamma(n_{k,v_{a,b}}+\beta_{v_{a,b}})}{\Gamma(\sum_{v=1}^{V}n_{k,v}+\beta_v)} \end{aligned} \tag{4.14}$$

Where $v_{a,b}$ is the term of the bth word from document a.

The label of the document is constant with respect to the topic chosen for $z_{a,b}$ so the probability of the document label can be dropped. The topic assignments for documents from different classes are also constant with respect to documents from the current class, so the product over other classes can be dropped. The topic assignment of other terms from the vocabulary are constant with respect to the current term of the vocabulary, so the probability over the other terms can be dropped.

$$p(w_{a,b},y_a,z_{a,b}|\alpha,\beta,\eta) \propto \frac{\prod_{k=1}^{K}\Gamma(n_{y_a,k}+\alpha_k)}{\Gamma(\sum_{k=1}^{K}n_{y_a,k}+\alpha_k)}\prod_{k=1}^{K}\frac{\Gamma(n_{k,v_{a,b}}+\beta_{v_{a,b}})}{\Gamma(\sum_{v=1}^{V}n_{k,v}+\beta_v)} \tag{4.15}$$

This probability still depends on other documents from the same class. In fact, the

topic assignments of documents from the same class are constant with respect to the topic assignments of the current document. The above probability becomes:

$$\begin{aligned} &= \frac{\prod_{k=1}^{K}\Gamma(n_{y_a,k}^{m\neq a}+n_{y_a,k}^{m=a}+\alpha_k)}{\Gamma(\sum_{k=1}^{K}n_{y_a,k}+\alpha_k)}\prod_{k=1}^{K}\frac{\Gamma(n_{k,v_{a,b}}+\beta_{v_{a,b}})}{\Gamma(\sum_{v=1}^{V}n_{k,v}+\beta_v)} \\ &\propto \frac{\prod_{k=1}^{K}\Gamma(n_{y_a,k}^{m=a}+\alpha_k)}{\Gamma(\sum_{k=1}^{K}n_{y_a,k}+\alpha_k)}\prod_{k=1}^{K}\frac{\Gamma(n_{k,v_{a,b}}+\beta_{v_{a,b}})}{\Gamma(\sum_{v=1}^{V}n_{k,v}+\beta_v)} \end{aligned} \tag{4.16}$$

Where $n_{y_a,k}^{m\neq a}$ is the number of times that topic k is assigned to a word from any document of the same class that is not the current document.

Let $n^{\neg a,b}$ be the count for all positions in a document except the current one. For the topic $z_{a,b}$ that is assigned to position (a,b), the count for that topic $n = n^{\neg(a,b)} + 1$. For all other topics, $n = n^{\neg(a,b)}$. We can split the product over K to be in terms of $z_{a,b}$ and $z_{\neg(a,b)}$

$$\begin{aligned} &\frac{\prod_{k\neq z_{a,b}}^{K}\Gamma(n_{y_a,k}^{\neg(a,b)}+\alpha_k)*\Gamma(n_{y_a,z_{a,b}}^{\neg(a,b)}+\alpha_{z_{a,b}}+1)}{\Gamma(\sum_{k=1}^{K}n_{y_a,k}+\alpha_k)}\times \\ &\prod_{k\neq z_{a,b}}^{K}\frac{\Gamma(n_{k,v_{a,b}}^{\neg(a,b)}+\beta_{v_{a,b}})}{\Gamma(\sum_{v=1}^{V}n_{k,v}^{\neg(a,b)}+\beta_v)}\times\frac{\Gamma(n_{z_{a,b},v_{a,b}}^{\neg(a,b)}+\beta_{v_{a,b}}+1)}{\Gamma(\sum_{v=1}^{V}n_{z_{a,b},v}^{\neg(a,b)}+\beta_v+1)} \end{aligned} \tag{4.17}$$

A property of the Γ function is that $\Gamma(\alpha+1)=\alpha\Gamma(\alpha)$. Using this property:

$$\begin{aligned} &= \frac{\prod_{k\neq z_{a,b}}^{K}\Gamma(n_{y_a,k}^{\neg(a,b)}+\alpha_k)*\Gamma(n_{y_a,z_{a,b}}^{\neg(a,b)}+\alpha_{z_{a,b}})(n_{y_a,z_{a,b}}^{\neg(a,b)}+\alpha_{z_{a,b}})}{\Gamma(\sum_{k=1}^{K}n_{y_a,k}+\alpha_k)}\times \\ &\prod_{k\neq z_{a,b}}^{K}\frac{\Gamma(n_{k,v_{a,b}}^{\neg(a,b)}+\beta_{v_{a,b}})}{\Gamma(\sum_{v=1}^{V}n_{k,v}^{\neg(a,b)}+\beta_v)}\times\frac{\Gamma(n_{z_{a,b},v_{a,b}}^{\neg(a,b)}+\beta_{v_{a,b}})}{\Gamma(\sum_{v=1}^{V}n_{z_{a,b},v}^{\neg(a,b)}+\beta_v)}\frac{(n_{z_{a,b},v_{a,b}}^{\neg(a,b)}+\beta_{v_{a,b}})}{(\sum_{v=1}^{V}n_{z_{a,b},v}^{\neg(a,b)}+\beta_v)} \end{aligned} \tag{4.18}$$

Recombine the products over Gamma to products over K.

$$= \frac{\prod_{k=1}^{K} \Gamma(n_{y_a,k}^{\neg(a,b)} + \alpha_k)(n_{y_a,z_{a,b}}^{\neg(a,b)} + \alpha_{z_{a,b}})}{\Gamma(\sum_{k=1}^{K} n_{y_a,k} + \alpha_k)} \times$$
$$\prod_{k=1}^{K} \frac{\Gamma(n_{k,v_{a,b}}^{\neg(a,b)} + \beta_{v_{a,b}})}{\Gamma(\sum_{v=1}^{V} n_{k,v}^{\neg(a,b)} + \beta_v)} \frac{(n_{z_{a,b},v_{a,b}}^{\neg(a,b)} + \beta_{v_{a,b}})}{(\sum_{v=1}^{V} n_{z_{a,b},v}^{\neg(a,b)} + \beta_v)} \tag{4.19}$$

The products over K and $\Gamma(\sum_{k=1}^{K} n_{y_a,k} + \alpha_k)$ become constants with respect to the $z_{a,b}$ assignment.

$$\propto \quad (n_{y_a,z_{a,b}}^{\neg(a,b)} + \alpha_{z_{a,b}}) \frac{(n_{z_{a,b},v_{a,b}}^{\neg(a,b)} + \beta_{v_{a,b}})}{(\sum_{v=1}^{V} n_{z_{a,b},v}^{\neg(a,b)} + \beta_v)} \tag{4.20}$$

To form a probability, a normalizing factor is needed.

$$p(z_{a,b}|z_{\neg(a,b)}, x, y, \alpha, \beta, \eta) \quad = \quad \frac{(n_{y_a,z_{a,b}}^{\neg(a,b)} + \alpha_{z_{a,b}}) \frac{(n_{z_{a,b},v_{a,b}}^{\neg(a,b)} + \beta_{v_{a,b}})}{(\sum_{v=1}^{V} n_{z_{a,b},v}^{\neg(a,b)} + \beta_v)}}{\sum_{k=1}^{K} n_{y_a,k}^{\neg(a,b)} + \alpha_k) \frac{(n_{k,v_{a,b}}^{\neg(a,b)} + \beta_{v_{a,b}})}{(\sum_{v=1}^{V} n_{k,v}^{\neg(a,b)} + \beta_v)}} \tag{4.21}$$

The next step is to estimate the expected values of Φ and Θ given the learned model,

$$\begin{aligned}
p(\theta_c|\alpha, M) &= p(\theta_c|\alpha)p(z|y,\theta_c) \\
&= \prod_{k=1}^{K} p(\theta_c|\alpha) \prod_{m=1}^{M} \prod_{n=1}^{N_m} p(z_{m,n}|\theta_{y_m=c}) \\
&= \prod_{k=1}^{K} p(\vec{\theta}_c|\alpha) \prod_{y_m=c}^{M} \prod_{k=1}^{K} p(\vec{z}_m|\vec{\theta}_{y_m=c}) \\
&= \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{c,k}^{\alpha_k-1} \prod_{k=1}^{K} \theta_{c,k}^{n_{c,k}} \\
&= \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{c,k}^{n_{c,k}+\alpha_k-1} \\
&\sim Dir(\vec{\theta}_c|\vec{n}_c + \vec{\alpha}) \\
\theta_{c,k} &= \frac{n_{c,k}+\alpha_k}{\prod_{k=1}^{K} n_{c,k}+\alpha_k}
\end{aligned} \tag{4.22}$$

The expected value of a Dirichlet distribution $Dir(a) = \frac{a_i}{\sum_i a_i}$

$$\begin{aligned}
p(\Phi|\beta, M) &= p(\Phi|\beta)p(w|\Phi, z) \\
&= \prod_{k=1}^{K} p(\phi_k|\beta) \prod_{k=1}^{K} \prod_{v=1}^{V} p(w_v|\phi_{z_v=k}) \\
&= \prod_{k=1}^{K} \frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)} \prod_{v=1}^{V} \phi_{k,v}^{\beta_v-1} \prod_{k=1}^{K} \prod_{v=1}^{V} \phi_{k,v}^{n_{k,v}} \\
&= \prod_{k=1}^{K} \frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)} \prod_{v=1}^{V} \phi_{k,v}^{n_{k,v}+\beta_v-1} \\
p(\phi_k|\beta, M) &= \frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)} \prod_{v=1}^{V} \phi_{k,v}^{n_{k,v}+\beta_v-1} \\
p(\phi_k|\beta, M) &\sim Dir(\vec{\phi}_k|\vec{n}_k + \vec{\beta}) \\
\phi_{k,v} &= \frac{n_{k,v}+\beta_v}{\sum_{v=1}^{V} n_{k,v}+\beta_v}
\end{aligned} \tag{4.23}$$

4.3 Prediction

Once the latent variables are known from the inference step, the next step is to classify unseen documents into one of the classes using the topic model.

$$
\begin{aligned}
p(\hat{y_c}|\hat{w}, z, \Phi, \Theta, \alpha, \beta, \eta) &= \frac{p(\hat{y}, \hat{w}, z, \Phi, \Theta|\alpha, \beta, \eta)}{\sum_{c=1}^{C} p(\hat{y_c}, \hat{w}, z, \Phi, \Theta|\alpha, \beta, \eta)} \\
&\propto p(\hat{y_c}, \hat{w}, z, \Phi, \Theta|\alpha, \beta, \eta) \\
&= p(\Theta_{y_c}|\alpha)p(\Phi_z|\beta)p(z|\Theta_{y_c})p(w|\Phi_z)p(y_c|\eta) \\
&= \int\int \sum_{k=1}^{K} p(\Theta_{y_c}|\alpha)p(\Phi_z|\beta)p(z|\Theta_{y_c})p(w|\Phi_z)p(y_c|\eta)d\Theta d\Phi \\
&= \prod_{n=1}^{N_m} p(y_c|\eta) \int\int p(\Theta_{y_c}|\alpha)p(\Phi_z|\beta)p(w_n|\hat{\theta_{y_c}}, \hat{\Phi})d\theta d\Phi \qquad (4.24)
\end{aligned}
$$

Where $p(w_n|\hat{\Theta}_{y_c}, \hat{\Phi}) = \sum_{k=1}^{K} p(z = k|\Theta_{y_c})p(w|\Phi_z) \simeq (\hat{\Theta}\hat{\Phi})$, the product of the Θ and Φ matrices.

$$
\begin{aligned}
y &= \arg\max_c \eta_c \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{c,k}^{n_{c,k}+\alpha_k-1} (\prod_{k=1}^{K} \frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\prod_{v=1}^{V} \Gamma(\beta_v)} \prod_{v=1}^{V} \phi_{n,k}^{n_{k,v}+\beta_v-1}) \\
&\propto \eta_c \prod_{k=1}^{K} \theta_{c,k} \prod_{v=1}^{V} \phi_{n,k} \\
&\simeq \prod_{n=1}^{N_m} \eta_c \hat{\Theta}_c \hat{\Phi}_{k,w_n} \qquad (4.25)
\end{aligned}
$$

The goal is to find the class label, y, which is the most likely candidate class, c. To find this class, the exact probability is not needed, only which class has the highest probability. Since the denominator is a normalizing constant, it can be dropped. When predicting the class

label for a new document, $\hat{w}$, Θ and Φ are constant, so they can also be dropped. Since the gamma functions are constants with respect to the class of a document, they can be dropped as well.

$$y = \arg\max_{c} \eta_c \prod_{n=1}^{N_m} \hat{\Theta}_c \hat{\Phi}_{k,w_n} \tag{4.26}$$

This chapter proposed the framework for subtopicLDA. Using the collapsed Gibbs sampling algorithm, subtopicLDA can be programmed to build the model from the training data. From the prediction step in Equation 4.26, unseen documents can be classified. In the next chapter, subtopicLDA is tested with real data for event detection using Twitter to determine which tweets are relevant for a foodborne illness outbreak.

Chapter 5

Event Detection from Twitter for Foodborne Illness

In the last chapter, subtopicLDA was proposed. This chapter demonstrates how subtopicLDA performs on real data. Foodborne illness outbreak is an event that has a large social impact on society. As mentioned in the Chapter 1, foodborne illness causing bacteria are accountable for millions of illnesses and thousands of deaths in the United States every year [12]. In Chapter 3, previous work demonstrated that Twitter can be used for event detection because users on Twitter are posting real-time information about the events around them. This chapter presents the experimental process and results for applying subtopicLDA to Twitter data for determining which tweets are relevant to a foodborne illness outbreak.

This chapter is divided into four sections. In the first section, data collection from Twitter is described. Section 2 discusses how the data was labeled. The preprocessing steps are presented in Section 3. Finally results of how various models performed on classifying tweets as related to foodborne illness or not is described in Section 4. In section 4, subtopicLDA is compare to generic classifers on standard datasets used in document classification.

Query terms		
nausea	nauseate	stomach hurt
vomit	puke	stomach cramp
diarrhea	fever	abdominal cramp
listeria	salmonella	stomach ache
e. coli	tummy ache	tummy cramp
food poisoning		

Table 5.1: Keywords for Collecting Tweets

5.1 Data Collection

Data collected was from the Twitter Streaming API using Twitter4j[1], a Java library for accessing the Twitter API. The Streaming API broadcasts tweets that match the query terms as soon as the tweet is posted. A program using the Twitter4j library listens to the Streaming API and saves any tweets it receives. The data is all tweets containing one of the words or phrases from Table 5.1, from December 5, 2011 to July 2, 2012. The list of query terms is a biased list of terms. There may be some terms people are using to express that they have foodborne illness that are not among these query terms. Some data during this time period is missing, due to internet connectivity issues. For missing data, there is a sharp drop in the charts in the Results section. For the days that the Twitter listener did not collect tweets, the number of tweets used for those days is the actual number collected from the API instead of averaging the tweet frequency over these time periods from tweet counts of earlier and later days. Total 5,106,710 tweets were collected. Some of these tweets are labeled. These tweets were posted between December 5 and December 8; there were 49,348 labeled tweets. The next section describes how these tweets were labeled.

[1]http://twitter4j.org/en/index.html

5.2 Labeling

An important step to detecting a foodborne illness outbreak is labeling the data. Tweets, that contain either the symptoms of FBI or the bacteria names that commonly cause FBI, tend to fall into one of five themes:

1. Tweet author is ill with a FBI
2. News article about FBI outbreak
3. Retweets or Responses to a tweet from Theme 1.
4. Tweet author has different disease like seasonal influenza
5. Tweet is completely unrelated to any illness

The theme that is associated with positive class is Theme 1. Tweets that fall under any of the other themes are considered to belong to the negative class. Labelers were given examples from each theme as well as the most common symptoms of foodborne illness to determine if the user was ill with FBI or not. If a child of a user was ill with FBI, the labelers could choose to label it as either positive class or negative class, since it was not the author of the tweet who was ill.

Tweets were labeled through one of three applications. For most of the labeling, labelers were given a spreadsheet with the tweet id and content of several tweets. They were asked to label each tweet for foodborne illness, given the symptoms the tweet mentions and the examples for each theme given. This approach tended to produce several tweets that were false negatives. In spreadsheets, the labeler could copy a label to multiple tweets without necessarily looking at each tweet. Since most of the tweets were in the negative class, some

tweets that could be part of the positive class were given the negative label. Tweets that contained the phrase "food poisoning" and given the negative class label were reviewed and updated if the tweet clearly claimed the user had foodborne illness. The rest of the tweets maintained the original label from the labeler. The second approach to label tweets was Amazon's Mechanical Turk[2]. After preliminary testing, Mechanical Turk appeared to be a more expensive approach without significant improvement in the accuracy of the labels, so this approach was discarded. The third approach was a php webpage that gave labelers only 1 tweet at a time. This webpage forced labelers to look at each tweet individually. It also updated a database automatically. The spreadsheet approach needed to be converted into SQL to update the database manually.

The number of labels each tweet varies depending on the method of labeling. All labeled tweets received their first label through the spreadsheet method. Labels from Amazon Turk were discarded, because they were less reliable and pertained to future work. Of the labeled tweets, some contain a second label and a third label. The second label was acquired through the website approach. The third label was set either automatically, if the first and second label were the same, or a third label was acquired through another website to break ties.

5.3 Preprocessing

Once the collection of tweets were labeled, the next step was to preprocess the tweets into feature vectors. Tweets were removed if they were not in English or they are retweets. Retweets occur when one user posts the tweet that another user posted. If the original post is from someone who is ill, the user posting the retweet is not likely ill. Latent Dirichlet

[2]https://www.mturk.com/mturk/welcome

Allocation makes the bag of words assumption but can be expanded to include bigrams and trigrams [4]. From the content of the tweets, all non-stopword unigrams were extracted. Bigrams and Trigrams were extracted after stopwords removal. Hashtags, a user added label to the tweet, were included as additional words in the content (with the # symbol removed). In some cases, the only mention of the symptom term is in the hashtag. Mentions of another Twitter user (starting with @ followed by the username) were removed since they were likely to appear in only 1 tweet per username mention. To increase the number of words that appear in common among tweets, Porter stemmer was used to stem all words that remained in the dataset. Singletons, terms that only appear in one tweet, were removed.

5.4 Results

In this section, subtopicLDA is compared with previous approaches to demonstrate the advantage of using topic modeling for text classification. The focus of this research is to classify tweets from Twitter. However, subtopicLDA can be applied to any document dataset. In Twitter, the goal is to predict if the user is ill with foodborne illness or not, based on the content of their tweet. Since the ground truth, whether the user was actually ill, is not available, the tweets are labeled and these labels are considered the ground truth.

The performance of subtopicLDA is compared to generic classifiers that are able to handle high dimensional data. The perfomance measure used is dependent on the number of classes in the dataset. For binary data, such as Twitter data, precision, recall, and F-measure are evaluated. Twitter data is strongly skewed to the negative class. The class of interest is the positive class, so precision, recall, and F-measure are reported for the positive class only. Overall accuracy is not used on Twitter data because a classifier can achieve high accuracy by

classifying all datapoints as belonging to the negative class. For multiclass datasets, accuracy is used. In multiclass data, all classes are equally important, so the overall accuracy can be used. The dataset used is articles from LA Times, which has 6 classes. In LA Times, the proportion of data in each class varies, but no single class contains a majority of the data.

In the first subsection, previous approaches are tested on foodborne illness Twitter data. Previous approaches can be divided into two categories: keyword filtering and supervised classification with generic classifiers. Classifiers include K-Nearest Neighbor (KNN) and Support Vector Machines (SVM), which perform well on high dimensional data. The second subsection demonstrates how topic models perform on determining relevant tweets for foodborne illness from irrelevant tweets. The results from subtopicLDA are presented in this subsection. Subsection 3 compares subtopicLDA with generic classifiers on LA Times datasets.

5.4.1 Baseline

Previous work used two general approaches, keyword filtering and generic classification, to determine which tweets are relevant to the health-related event of interest. Most related work focused on influenza monitoring. Early work in influenza monitoring used simple keyword filtering. They counted the total number of tweets per day containing one of their query terms. For influenza, some search terms provided useful results with simple counting. "Flu" for example is primarily used in only one context. In this foodborne illness dataset, the phrase "food poisoning" has a similar property. If a tweet mentions the phrase "food poisoning", it most often pertained to the user claiming they had food poisoning. This phrase is only used to mean one of a few things, so a spike in the raw count, as seen in Figure 5.1, may correlate to an outbreak. According to the FDA, on December 5 there was an *e. coli* outbreak in 5

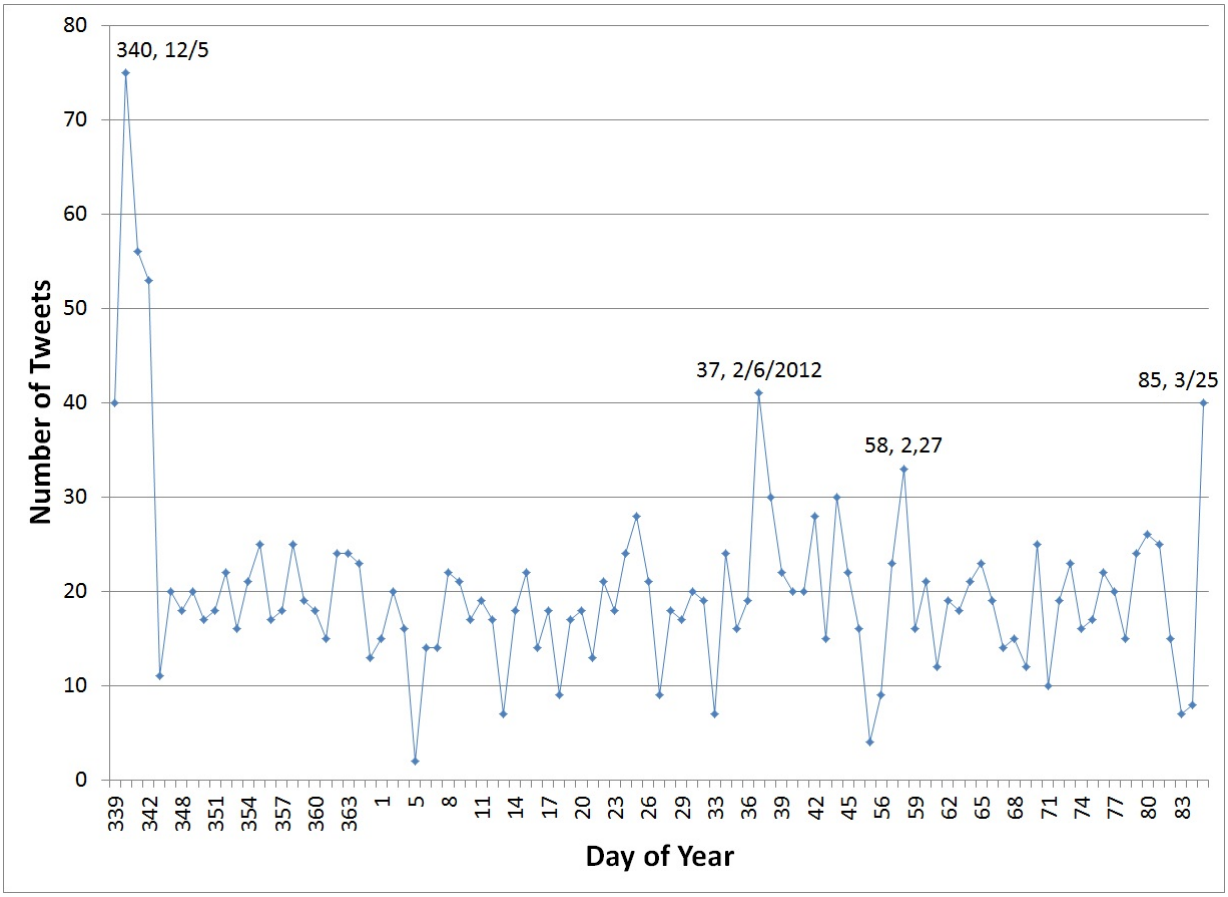

Figure 5.1: Number of Tweets per Day Containing "Food Poisoning"

states linked to food from Taco Bell[3].

The state a user is from is based on the user's location from their profile. The location field is an open textbox, so a user can express their location in anyway they like. This includes using vernaculars, mentioning multiple locations, or even mentioning fictional locations. The raw counts shown in Figure 5.1 includes all users, regardless of what they place in their profile location field. In Figure 5.2, a user's profile location is compared to city names, a set of vernaculars, state names, and state abbreviations to determine their location. If their state can be determined from their profile location, then their tweet is added to that state's tweets.

[3]http://www.fda.gov/NewsEvents/Newsroom/PressAnnouncements/2006/ucm108801.htm

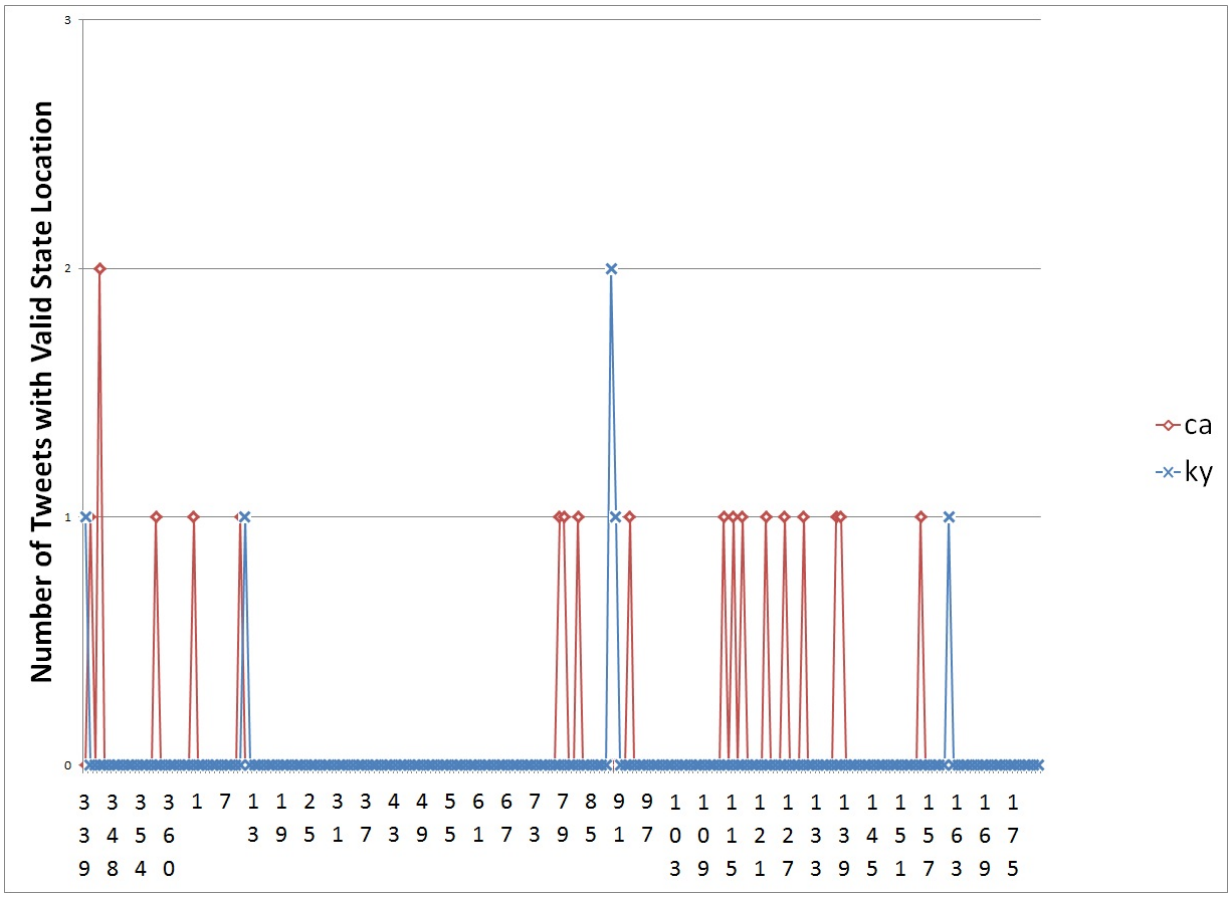

Figure 5.2: Number of Tweets per Day Containing "Food Poisoning" For California and Kentucky

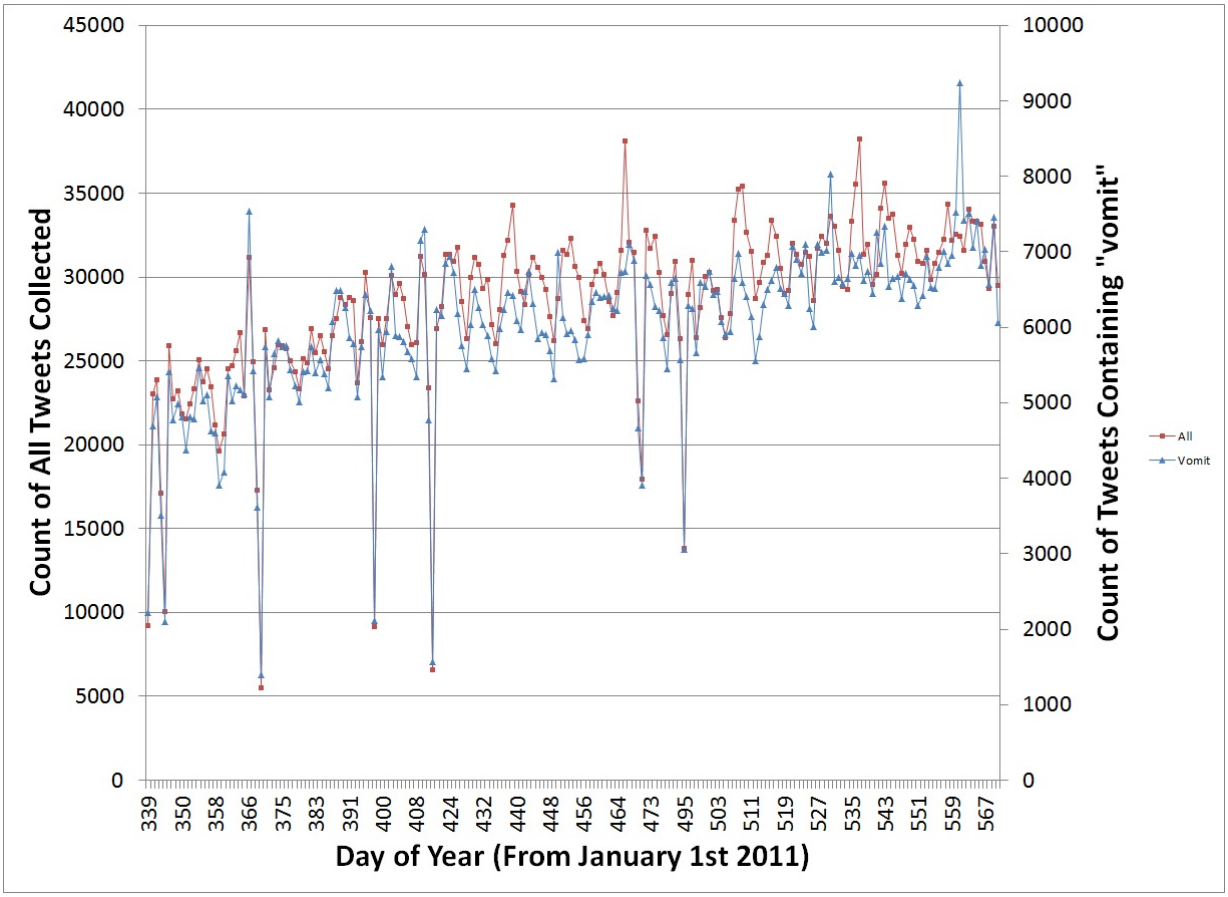

Figure 5.3: Number of Tweets per Day Containing the Term "Vomit"

When looking at the frequency of the phrase "food poisoning" for each state, state frequency lines are similar to those in Figure 5.2. For most states, the maximum count of states from a user in that state is one or two, with most of the days having zero tweets with the phrase. For simplification, the chart in Figure 5.2 only shows 2 states to keep the chart readable.

Not every Twitter user suffering from the symptoms of foodborne illness will tweet the phrase "food poisoning". Users may instead tweet their symptoms. Much of the noise encountered in Twitter relates to determining when a symptom term relates to a real symptom versus when a symptom term is used to describe something unrelated to foodborne illness. Figure 5.3 shows the frequency of tweets with the term "vomit".

Classifier	Precision	Recall	F-Measure
Naïve Bayes	4.20%	3.80%	3.99%
KNN	6.57%	5.81%	6.17%
SVM	5.49%	9.72%	7.00%

Table 5.2: Performance of Generic Classifiers

Clearly, simple term frequency per day is not a good indicator of whether an outbreak occurred. On January 1st, the increase use of "vomit" relates to over consumption of alcohol instead of an outbreak of foodborne illness. Some classifier is needed to determine which tweets are relevant to a likely outbreak of foodborne illness. When simple classification works well, then simple classifiers should be favored over more complex classifiers. Three classifiers that are able to handle multidimensional data are K-Nearest Neighbor(KNN), Naïve Bayes, and Support Vector Machines (SVM). These are the baseline used to compare with subtopicLDA.

Implementation of these classifiers may vary slightly from application to application, and two of these classifiers take parameters that need tuning. For Naïve Bayes and K-Nearest Neighbor, Matlab's implementation of these classifiers was used, with 10-fold cross validation. In 10-fold cross validation, one set is the validation set and the remaining 9 sets are used to train the model. The precision and recall reported is the average precision and recall for the positive class over all 10 validation sets. The positive class are the tweets that indicate that the author is suffering from symptoms of foodborne illness. K-Nearest Neighbor takes a parameter, the number of neighbors k, that is estimated from cross validation. The values for K that were tested were 1, 3, and 5. K=1 returned the highest F-measure. As K increases, precision increases but recall decreases significantly. For example, for K=3 recall is less than 1%.

SVMlite[4] is used to test SVM. The implementation of SVM in SVMlite has a set of default parameters, with the option to change the parameters. One of these parameters determines whether to weigh errors from the positive class equal to, more, or less than errors made on data points on the negative class. This parameter is used to determine the optimal support vector from the training data. The default is to weigh all points equally in determining the support vector. This works well when both the positive class and the negative class have approximately the same number of data points. However in this dataset, the positive class only makes up 5% of the data points. If all data points are weighted equally, the model overfits to the negative class. Precision and recall for the positive class are zero in this case. To balance the errors from each class, an error from misclassification during training of a data point from the positive class is weighted 20 times more heavily the error of misclassifying a data point from the negative class. This become balanced since the negative class has roughly 20 times more data points than the positive class.

Generic classifiers, at least KNN, SVM, and Naïve Bayes, perform very poorly on classifying whether a tweet relates to a foodborne illness outbreak or not. There are two challenges that cause these classifiers to perform poorly. The first challenge generic classifiers face is that the feature set is very large. The features do not provide a good representation of the class because the same terms appear in documents of both classes. A more complex model is needed for this classification problem. The second challenge is this dataset is skewed to a specific domain. For a tweet to belong to this dataset, it must contain either a symptom of foodborne illness or one of the bacteria that cause foodborne illness. These classifiers would improve performance if the sample of tweets was a random sample of all tweets instead of only tweets containing symptom terms.

[4]http://svmlight.joachims.org/

Classifier	Precision	Recall	F-Measure
KNN on All Words	6.57%	5.81%	6.17%
KNN on LDA Topics	5.28%	7.64%	6.25%

Table 5.3: Performance After Feature Reduction using LDA

5.4.2 Topic Modeling

Probabilistic topic models, such as Latent Dirichlet Allocation (LDA), attempt to model the underlying topic distribution that the words are drawn from. LDA is an unsupervised topic model, so a class label cannot be predicted directly from LDA. However, LDA can be used for dimension reduction, transforming a document from its word vector space to its topic vector space, by performing feature reduction. The θ matrix is the expected topic distribution for each document. This topic distribution matrix is passed to a generic classifier. In this experiment, LDA was run on the entire labeled dataset to generate the θ matrix, the topic distribution for each document. The topic distribution per document dataset is split for 10-fold cross validation, and passed to a classifier. In this experiment, the classifier chosen was KNN classifier, with K=1. The results are listed in Table 5.3. In this experiment, the parameters for LDA were set as $\alpha = 0.5, \beta = 0.5, k = 20$.

Using topic modeling for feature reduction does not improve a classifier's accuracy. This implies that the words provide more information to a classifier than the underlying topics do. The reason LDA paired with a generic classifier performs poorly is that the information about the class label is not used to determine which topics best describe the data. In subtopicLDA, the label of a document influences the distribution of topics that a document selects from. SubtopicLDA is a supervised model, so the prediction step is built into the model; there is no dependency on any other model to perform the classification for subtopicLDA.

In the experiment, the data was split using 10-fold cross validation. For subtopicLDA,

Classifier	Precision	Recall	F-Measure
Naïve Bayes	4.20%	3.80%	3.99%
KNN	6.57%	5.81%	6.17%
SVM	5.49%	9.72%	7.00%
SubtopicLDA	**5.85%**	**53.07%**	**10.32%**

Table 5.4: Comparison SubtopicLDA to Baseline

there are four parameters that need to be tuned, α, β, η, and k. η represents the probabulity that a document will belong to each document. SubtopicLDA generates the best results when η is set to the proportion of the positive class in the training set. For Twitter, η for the negative class is 95% and 5% for the positive class. The remaining three parameters were determined from how the model performed on the validation set. The optimal parameters used in this experiment are $\alpha = 0.00005, \beta = 0.00005, k = 20$. The performance of SubtopicLDA is compared to the three baseline classifiers in Table 5.4.

From these results, subtopicLDA outperforms Naïve Bayes. SubtopicLDA has similar precision to KNN and SVM, but performs significant better in recall. In KNN, precision is higher, indicating that more documents that are classified as positive are true positives, but KNN is unable to predict that most of the documents from the positive class belong to the positive class. SubtopicLDA is able to retrieve more relevant documents during classification than KNN, based on its higher recall. As mentioned above, as K increases, precision increases but recall falls to under one percent with only minor adjustments to K. When comparing KNN to subtopicLDA on precision alone, KNN is the better model. However when comparing these two models on F-measure, subtopicLDA performs better than KNN. The goal of this research is to detect the signal that a foodborne illness outbreak occurs, so high recall is important to retrieve all the relevant tweets. Therefore subtopicLDA outperforms KNN.

Although precision and recall are both important, precision is used to compare subtopi-

cLDA to SVM because both models can achieve perfect recall by classifying all tweets as positive class, depending on how parameters are tuned. Both models were tuned to find the highest precision, with the restriction that the overall accuracy (on both classes) is at least 50%. On precision, subtopicLDA outperforms SVM at 90% confidence. Among the documents that each model predict as belonging to the positive class, subtopicLDA found a higher proportion of those documents were true positives. SubtopicLDA is a better model than SVM for predicting which tweets are relevant to event detection of foodborne illness.

From experimental results, none of the classifiers were able to obtain precision above 10%. One reason for this is that each data point was only labeled by one labeler. A labeler can present bias towards their labels. As mentioned earlier in this chapter, some of these labels may be incorrect due to how the data was labeled. In future work, more labels will be obtained so the label of each document is the consensus vote of multiple labelers. Each data point would have at least 3 labels. The label given to the classification model is the label determined by majority vote. A second reason that none of the classifiers performed extraordinarily is in the data, specifically in the terms each tweet contains. As mentioned in the previous subsection, every tweet contains at least one of the query terms. These models are attempting to predict when each term is used in the context of foodborne illness and when the same term is used in a different context. If the dataset contained tweets without terms related to foodborne illness, the classifiers would have significant improvement.

5.4.3 LA Times

SubtopicLDA is a classifier that can handle multiclass classification. To demonstrate this capability, subtopicLDA is applied to a general datasets, LA Times articles. Similar to Twitter data, subtopicLDA is compared to SVM on this general dataset. SVM-light has

a multiclass version, called SVM-Multiclass[5]. Since SVM-Multiclass only returns accuracy, SVM is compared to subtopicLDA on overall accuracy.

LA Times articles have several differences from Twitter data. Among the differences between these datasets, documents in LA Times are significantly longer than the 140-character tweets and the terms used typically are correctly spelled. Due to their length, LA Times data provides more information about the context of the article in the terms used, while Twitter data hides the underlying context to stay within the 140-character limit. In LA Times, the number of classes is larger than the Twitter data. LA Times has 6 classes. When the documents are written in LA Times, the class is known and the data comes labeled. In Twitter, the class information is not known by the author of the tweet. They are simply stating an opinion or reporting an event, without knowledge of the underlying class information that will be assigned to their tweet. In LA Times, the size of the dataset for each class varies, but the data is not as skewed to one class as the Twitter data is. The smallest class in LA Times has 273 documents, and the largest class has 943 documents. The ratio of smallest class to largest class is 1:4 in LA Times and 1:19 in Twitter data. Twitter data is more skewed.

In experiments with LA Times data, the data is split using 10-fold cross-validation. The first fold is used to tune the parameters. Then the remaining folds are tested using the same parameters. SVM takes one parameter, c, which is the trade-off between training error and margin of the support hyperplanes. Setting c to 3 provided the best accuracy on the first fold, so all folds were tested using this parameter. For SubtopicLDA, the optimal parameters were $\alpha = 0.5, \beta = 0.5, k = 60$, and $\eta_c = \frac{1}{6}$ for each class c. Despite each class having a different number of documents in the training set, setting the probability for each

[5]http://svmlight.joachims.org/svm_multiclass.html

SubtopicLDA	Multi-SVM
66.32%	37.46%

Table 5.5: Accuracy of SubtopicLDA and SVM on LA Times

class, $p(y|\eta)$, to be equal for each class produced good results. Increasing or decreasing the number of topics, k, caused worse performance in accuracy.

The results appear in Table 5.5. SubtopicLDA performs better than SVM with 99% confidence. Longer documents provide more information to both models. LA Times is still a noisy dataset because words may be used in different contexts within the same document or in different documents from the same class. SubtopicLDA is able to adjust its probabilities to handle this noise by assigning terms to different topics based on the context each term is from. SVM does not take into account the context, or underlying topics, each term is drawn from when determining the optimal hyperplane to separate each class.

In this chapter, subtopicLDA was compared to generic classifiers on several datasets. On Twitter data and LA Times, subtopicLDA outperformed generic classifiers. In the next chapter, improvements to subtopicLDA and future work are discussed.

Chapter 6

Future Work

The focus of this research is event detection using social media, specifically detecting health-related events. There are three challenges to detecting or monitoring health-related events: identify which tweets are relevant to the event, detect when the discussion of an event starts, and determine where the event occurred. The focus of this thesis is to propose an approach to solve the first problem, ascertaining which tweets are relevant. The proposed solution is a model called subtopicLDA. Compared to generic classifiers used by previous approaches, subtopicLDA performs better than most classifiers.

Currently subtopicLDA makes the assumption that all tweets are independent of each other. This is not the case because tweets from the same user are related to each other. A user posts tweets related to what is important to them. For example if a user attends a concert, they are likely to post before they go that they are excited about the concert. During the concert they may post another tweet about the concert. These two tweets are not independent of each other because the same user is posting about the same event. If the model assumes these tweets are independent, it may determine that an event is larger than it really is.

An advantage of topic models, such as subtopicLDA, is they can be extended. SubtopicLDA can draw from author-topic models to include which user posted the tweet to help determine the tweet's context. The models in this thesis were unable to handle the noise

in the data because symptom words can also be used in other contexts. If the user data is included in an extended model, the model can incorporate a word distribution of each user to notice that some users may use terms like "puke" in differrent contexts, so the presense of these terms does not necessarily indicate the user is feeling ill.

There are two additional challenges to solve for event detection of health-related events: identifying when and where the event occurred. Some previous research has focused on predicting a user's location in Twitter. Kinsella et al. [21] used language models to predict the location of a user. They were able to discover some terms that are more likely to occur given the location of the user, such as school names and sports teams. Extensions from one language model can be applied to another language model, so subtopicLDA could be extended to incorporate location information of the user. Other previous research used classifiers to predict users' locations with limited success [7, 15, 26]. Predicting a user's location provides more useful information for event detection than determining the start time of an event, so the next challenge will be to predict the location of the user to identify the location of the event.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Harshavardhan Achrekar, Ross Lazarus, and West Cummings Park. Predicting flu trends using twitter data. *Architecture*, pages 702–707, 2011.

[2] Mark Astley. Us state food safety 'tweets' will help prevent potentially lethal outbreaks -fsis, March 2012.

[3] David Blei and Jon McAuliffe. Supervised topic models. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 121–128. MIT Press, Cambridge, MA, 2008.

[4] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[5] FluView: A Weekly Influenza Surveillance Report Prepared by the Influenza Division. Center for disease control and prevention, May 2012.

[6] Bob Carpenter. Integrating out multinomial parameters in latent dirichlet allocation and naive bayes for collapsed gibbs sampling. Technical report, 2010.

[7] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 759–768, New York, NY, USA, 2010. ACM.

[8] Courtney D. Corley, Armin R Mikler, Karan P. Singh, and Diane J. Cook. Monitoring influenza trends through mining social media. In *International Conference on Bioinformatics and Computational Biology (BIOCOMP09)*, Las Vegas, NV, July 2009.

[9] Aron Culotta. Towards detecting influenza epidemics by analyzing twitter messages. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 115–122, New York, NY, USA, 2010. ACM.

[10] Ed De Quincey and Patty Kostkova. *Early warning and outbreak detection using social networking websites: the potential of twitter*, volume 27, pages 21–24. Springer, 2010.

[11] Bernard Dixon. Foodborne disease in the social media age, 2009.

[12] Centers for Disease Control and Prevention. Cdc 2011 estimates: Findings. Website, October 2012.

[13] Evgeniy Gabrilovich, Susan Dumais, and Eric Horvitz. Newsjunkie: providing personalized newsfeeds via analysis of information novelty. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 482–490, New York, NY, USA, 2004. ACM.

[14] Jeremy Ginsberg, Matthew H. Mohebbi, Rajan S. Patel, Lynnette Brammer, Mark S. Smolinski, and Larry Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457:1012–1014, 2009.

[15] Brent Hecht, Lichan Hong, Bongwon Suh, and Ed H. Chi. Tweets from justin bieber's heart: the dynamics of the location field in user profiles. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 237–246, New York, NY, USA, 2011. ACM.

[16] Gregor Heinrich. Parameter estimation for text analysis. Technical report, 2004.

[17] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[18] Liangjie Hong and Brian D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 80–88, New York, NY, USA, 2010. ACM.

[19] Liangjie Hong, Byron Dom, Siva Gurumurthy, and Kostas 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*, KDD '11, pages 832–840, New York, NY, USA, 2011. ACM.

[20] Akshaya Iyengar, Tim Finin, and Anupam Joshi. Content-based prediction of temporal boundaries for events in twitter. In *Proceedings of the Third IEEE International Conference on Social Computing*. IEEE Computer Society, October 2011.

[21] Sheila Kinsella, Vanessa Murdock, and Neil O'Hare. "i'm eating a sandwich in glasgow": modeling locations with tweets. In *Proceedings of the 3rd international workshop on Search and mining user-generated contents*, SMUC '11, pages 61–68, New York, NY, USA, 2011. ACM.

[22] Simon Lacoste-Julien, Fei Sha, and Michael I. Jordan. Disclda: Discriminative learning for dimensionality reduction and classification. In *NIPS*, pages 897–904, 2008.

[23] Balaji Lakshminarayanan and Raviv Raich. Inference in supervised latent dirichlet allocation. In *IEEE International Workshop on Machine Learning for Signal Processing*, September 2011.

[24] Vasileios Lampos and Nello Cristianini. Tracking the flu pandemic by monitoring the social web. In *2nd IAPR Workshop on Cognitive Information Processing (CIP 2010)*, pages 411–416. IEEE Press, June 2010.

[25] Fei-Fei Li and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 524–531, Washington, DC, USA, 2005. IEEE Computer Society.

[26] Jalal Mahmud, Jeffrey Nichols, and Clemmens Drews. Where is this tweet from? inferring home locations of twitter users. In *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media*, 2012.

[27] Adam Marcus, Michael S. Bernstein, Osama Badar, David R. Karger, Samuel Madden, and Robert C. Miller. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 227–236, New York, NY, USA, 2011. ACM.

[28] Marcelo Mendoza, Barbara Poblete, and Carlos Castillo. Twitter under crisis: can we trust what we rt? In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 71–79, New York, NY, USA, 2010. ACM.

[29] David R. H. Miller, Tim Leek, and Richard M. Schwartz. A hidden markov model information retrieval system. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 214–221, New York, NY, USA, 1999. ACM.

[30] Ryan W. Newkirk, Jeff B. Bender, and Craig W. Hedberg. The potential capability of social media as a component of food safety and food terrorism surveillance systems. *Foodborne Pathogens and Disease*, 2012.

[31] Michael J Paul and Mark Dredze. A model for mining public health topics from twitter. *Technical Report. John Hopkins Universtiy*, (May 2009):166, 2011.

[32] Michael J Paul and Mark Dredze. You are what you tweet : Analyzing twitter for public health. *Artificial Intelligence*, 38:265–272, 2011.

[33] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM.

[34] Liza Potts, Joyce Seitzinger, Dave Jones, and Angela Harrison. Tweeting disaster: hashtag constructions and collisions. In *Proceedings of the 29th ACM international conference on Design of communication*, SIGDOC '11, pages 235–240, New York, NY, USA, 2011. ACM.

[35] Alexei Pozdnoukhov and Christian Kaiser. Space-time dynamics of topics in streaming text. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, LBSN '11, pages 1–8, New York, NY, USA, 2011. ACM.

[36] Daniele Quercia, Harry Askham, and Jon Crowcroft. Tweetlda: supervised topic classification and link prediction in twitter. In *Proceedings of the 3rd Annual ACM Web Science Conference*, WebSci '12, pages 247–250, New York, NY, USA, 2012. ACM.

[37] 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*, EMNLP '09, pages 248–256, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[38] Toni M. Rath, Maximo Carreras, and Paola Sebastiani. Automated detection of influenza epidemics with hidden markov models. In *In IDA*, pages 521–532. Springer, 2003.

[39] Joshua Ritterman, Miles Osborne, and Ewan Klein. Using prediction markets and Twitter to predict a swine flu pandemic. In *Proceedings of the 1st International Workshop on Mining Social Media*, 2009.

[40] 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*, UAI '04, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press.

[41] 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*, WWW '10, pages 851–860, New York, NY, USA, 2010. ACM.

[42] Ari Seifter, Alison Schwarzwalder, Kate Geis, and John Aucott. The utility of google trends for epidemiological research: Lyme disease as an example. *Geospatial Health*, 4(2):135–137, 2010.

[43] Spinn3r. Indexing the blogosphere. Website, 2005.

[44] Jeannette N. Sutton, Leysia Palen, Irina Shklovski, and Fifth international ISCRAM conference. Backchannels on the front lines : emergency uses of social media in the 2007 southern california wildfires. In *5th International ISCRAM Conference*. University of Colorado, 2008.

[45] Jaime Teevan, Daniel Ramage, and Merredith Ringel Morris. #twittersearch: a comparison of microblog search and web search. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 35–44, New York, NY, USA, 2011. ACM.

[46] Twitter. Twitter blog: Shutting down spammers. Blog, April 2012.

[47] Sarah Vieweg, Amanda L. Hughes, Kate Starbird, and Leysia Palen. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 1079–1088, New York, NY, USA, 2010. ACM.

[48] Yu Wang, Eugene Agichtein, and Michele Benzi. Tm-lda: efficient online modeling of latent topic transitions in social media. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 123–131, New York, NY, USA, 2012. ACM.

[49] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 261–270, New York, NY, USA, 2010. ACM.

[50] Wayne Xin Zhao, Jing Jiang, Jing He, Yang Song, Palakorn Achananuparp, Ee-Peng Lim, and Xiaoming Li. Topical keyphrase extraction from twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 379–388, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[51] 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 *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, pages 338–349, Berlin, Heidelberg, 2011. Springer-Verlag.