Social Media Analytics to Support COVID-19 Pandemic Response

Submitted April 26, 2021, in partial fulfillment of
the conditions for the award of the degree **BSc Computer Science.**

Xinyi OUYANG
20030865

Supervised by Dr.Heng YU

I hereby declare that this dissertation is all my own work, except as indicated in the text.

School of Computer Science University of Nottingham Ningbo China

Abstract

COVID-19 has been regarded as a serious disaster because of the enormous influence on human society, and during the COVID-19 pandemic, the use of social media has been increased. In this situation, the social media information can help the researchers to grasp the situational information and public reactions to this pandemic. Therefore, social media, such as Twitter, can be regarded as a valuable tool to predict the outbreak of epidemic diseases. Based on some research, AI-based NLP topic models can be used to detect the latent topics, and LDA (Latent Dirichlet Allocation), BTM (Biterm Topic Model) are said to be two potent topic models for topic extraction. In this project, we intend to propose a framework to preprocess text data then extract the potential topics by implementing topic models and apply the algorithm on the Twitter dataset related to the epidemic for further analysis.

Contents

List of Figures

List of Tables

List of Abbreviations

BOW	Bag of Word
BTM	Biterm Topic Model
LDA	Latent Dirichlet Allocation
LSI	Latent Semantic Index
NLP	Natural Language Processing
NPMI	Normalized Pointwise Mutual Information
PLSA	Probabilistic Latent Semantic Analysis
TTM	Triterm Topic Model

Chapter 1

Introduction

1.1 Background

Epidemic diseases have an enormous influence on almost all aspects of society, and billions of people suffer from diverse epidemics every year [3]. Therefore, it is crucial for the public to be able to precisely analyze and predict the outbreaks of infectious diseases, and various traditional models have been exploited and performed in diverse aspects [4]. The Centers for Disease Control and Prevention (CDC) uses a surveillant system, which resorts to the outpatient reporting and virological test, and could notice the outbreak with a two-week delay after the disease occurred [5]. However, according to [5], the disease information from social media could respond more promptly. Since social media platforms provided the public with a communicating platform when emergency events happened, researchers could rely on those messages present in social media to capture the latest information of related events and keep abreast of the situational developments [6]. Based on research studies [7], social media, including Twitter and Facebook, is regarded as useful tools to detect disease outbreaks with a faster reaction than traditional methods in disease surveillance.

As one of the most popular social media, Twitter can give users a platform not only to communicate, but also to share information. It is estimated that millions of users publishing health information on Twitter on a daily basis [8]. In a research study about Zika virus (ZIKV), an emerging arbovirus which brought about an epidemic around the world, prediction of the outbreak based on Twitter data shows high accuracy. According to [9], Twitter data combined with Natural Language Processing (NLP)- based machine learning techniques have been used in medical areas such as medicinal drugs analysis through obtaining the topics and extracting the topics' trending from tweets. The above investigations show the significant worth of social media to monitor and analyze in the health area. Both cases demonstrate the potential of the social media data that can be exploited in the public health surveillance and prediction domains [10].

Topic models have been regarded as useful tools to analyze document collections and other discrete data [11]. As said in [12], probabilistic topic models have been proved useful to extract latent topics in documents and widespread used for dimension-reduction of sparse count data. Those models can abstract the words in a document to a lower-dimensional latent variable representation, which can catch the general meaning of the document beyond the specific words in it [12].

1.2 Motivation

During the COVID-19 pandemic, with the same as other mass convergence events occurred, the use of social media soared [13], and in such a conjuncture, the situational information is useful for public to react to the disease epidemic. Consequently, it is vital for researchers to grasp the phasic information and capture the widespread topics from social media during the pandemic. To fill this gap, this research will analyze the potential topics about COVID-19 on Twitter using AI-based Natural Language Processing methods.

Latent Dirichlet Allocation (LDA) topic model is said to be potent for semantic mining and topic extraction [14], and for short texts like tweets, another novel method proposal, referred to as Biterm Topic Model (BTM), is well recognized given its superior performance [15]. In this project, we will have an in-depth study on those two models and propose a new topic model based on BTM, then applied those topic models to social media data about COVID-19 to extract the latent topics of the documents.

1.3 Description of Work

Topic modeling is said to be as a method related to multi-component semantic and computational linguistics and has been applied in many research areas. The aim of this project is to implement and improve the topic model to discover and analyze the potential topics about COVID-19. Specifically, we want to find out:

1. The prevalent topics about COVID-19 in Twitter
2. The interrelation among the topics
3. The evolution of the topics during the pandemic

The key objectives are:

1. Collecting data from Social media
2. Data preprocessing
3. Designing and training a new topic model and evaluating with traditional topic model
4. Using the topic model to capture the topics of the documents
5. Result visualization

To sum up, the work of this project can be divided into two parts, one is having a depth learning about the traditional topics models and design our model, the other is applying our model to analyze COVID-19 related tweets.

Chapter 2

Related Work

2.1 Social Media

Nowadays, there is a large amount of data has been produced by online social media which covered diverse areas such as medicine, history, arts, and it leads to the creation of knowledge by analyzing and clustering those data [16]. As illustrated before, in this project, our work is to do text mining on social media and discover the potential topics about COVID-19 to analyze the public response during the pandemic. Before this project, the idea of analyzing events by Twitter data cluster has been attempted in many fields such as social interactions, sentiment analysis, and link prediction since there are over 300 million active users and it is open for researchers to access the information on Twitter [17].

As a popular social media outlet, Twitter has become a huge source of linguistic data consist of discussion, opinion and sentiment, and it is considered to have the potential to exert social influence [18]. During times of emergency, Twitter users send tweets with some detailed information which can help the government or researchers to analyze the events and make key decisions[19]. In [20], it proposed a system to discover the public responses to the pandemic and the evolution of the responses in different time by using NLP and text mining through the tweets corpus that related to the COVID-19 pandemic. Based on research about Twitter mention of 6162 COVID-19-related scientific publications [21], it has been found that the Twitter platform and the users are significant for spreading the research outputs on OVID-19, and the amounts of users that mention COVID-19 are on increase. Therefore, it is visible that social media plays an important role during the COVID-19 pandemic, and Twitter can be chosen as a suitable platform to set up the analysis of the epidemic.

2.2 Topic Model

Topic model is a kind of statistical model to discover the abstract topics in a group of documents, which is commonly used as a text mining tool to extract the latent semantic structures in a text [22]. Latent Dirichlet Allocation(LDA) is a classical topic model which has been described as a probabilistic model to process the collections of discrete text data [23]. LDA is a three-level hierarchical Bayesian model, and its basic idea is that the topics are the probabilistic distribution over words and the documents are generated by random potential topics. In the existing research [24], LDA has been proved as a useful tool to analyze collections of documents, especially on long text.

According to the statistical analysis, most of the texts from social media contain less than 140 characters [18]. With the plenty of use of the social network in people's daily life, it seems more important for topic models to semantic modeling the short texts on social media. However, short texts generally contain less effective information which makes the features of the sample sparsity, and with the high dimensionality of the feature set, it is hard to extract correct and key features to cluster [25]. Therefore, using traditional topic models such as LDA modeling on the short texts to inference the topic distribution, many values of the distribution would be zero, which will lead to the sparsity problem [26].

Aggregating the short texts to long pseudo-documents is a simple solution to reduce the sparsity problem and has been proved to work better than original LDA [27]. Nevertheless, the effect of the heuristic way to a large extent depends on the quality of data [15]. Adding strong assumptions on the short texts such as assuming each word in the same sentence has the same topic [28] is also helpful to alleviate the sparsity problem since it can simplify the model. But it might cause the loss of the possibility of capturing multiple topics in a document and might cause overfitting of the model [23]. Under this circumstance, Biterm Topic Mode (BTM) is proposed to extract the latent topics in a corpus by modeling the co-occurrence patterns (biterms) in the corpus rather than modeling the single word [15].

To get a more satisfied result from topic models, some technological improvements have been put on BTM. Li et al. [29] combined the K-means clustering algorithm with BTM, which uses BTM to attain the topics and uses K-means to cluster those topics better. A novel model named Relation BTM (R-BTM) using word embeddings to link short texts with similar words and enhance the variety of the biterm list [30]. Moreover, He et al. [31] proffered a fast BTM to accelerate the sampling process to suit large datasets. As for our model, we extend the biterms (word pairs) to triterms (three-word groups), and modeling the triterms in the corpus, wanted to extract topics efficiently.

2.3 Applying to COVID-19

According to [32], there is a hybrid AI model to implement COVID-19 prediction. Firstly, to analyze the variation of the infection rates and detect the spread and the development trend of the disease, an improved susceptible-infected (ISI) model was proffered. This model has been used for predicting some other diseases like SARS and Ebola and has shown strong capabilities in this field. However, those traditional models predict disease by analyzing the dynamic change of the number of diseased individuals and assume that each individual has the same infection rate, which has many limitations and only can generate a general prediction result. Additionally, because of the serious situation during the COVID-19 pandemic and based on some prevented measures and policies from the government and the self-consciousness of citizens, the traditional model cannot meet the prediction requirement of COVID-19. Some news information features and social media information should be considered in this case, so the Natural Language Processing (NLP) is proposed in this epidemic model to obtain a higher accuracy prediction [32].

With the proposal of putting NLP into epidemic models, our project would concentrate on the NLP part of the social media information and not focus on the infection. According to [33], there exist a software system to detect the outbreak of the disease based on machine

learning and data processing from social media data, whose general framework and goal suit our project. Based on the research [34, 35, 36], the data preprocessing framework has been proposed, which is also helpful to complete this project.

Thus, our project will learn from the above frameworks of the existing system and different topic models to implement the aims and objectives. The design and implementation details will be discussed in chapter 4&5. Moreover, Song et al. [37] applied a topic model on tracking and detecting the disinformation that occurred in the COVID-19 pandemic. And many other systems combined the topic model with other text information such as post time and location to implement a dynamic system to predict the broke out of the epidemic [38]. The future work of our project can be learned from those existing system, so that the topic model can be further and efficiently applied to the COVID-19 epidemic.

Chapter 3

Preliminaries

In this project, some unsupervised algorithms can be used as the clustering model to extract the potential topics in the input documents. According to [39], clustering, regarding as popular data mining algorithms, is extensively used in classification [40, 41] and document organization. However, some traditional clustering algorithms are not suited for text clustering because of some unique properties of text such as large dimensionality and the word correlation [39]. In this section, the two common types of text clustering algorithms, embedding-based clustering algorithms and probabilistic-based clustering algorithms, are demonstrated as the preliminaries of this project.

3.1 Word-Embedding-Based Clustering Algorithm

3.1.1 Feature Extraction Techniques

A simple way to implement text clustering is to extract the key features of the text and group the documents by traditional clustering algorithms such as K-means cluster. One-hot encoding, TF-IDF and word2vec are the most common strategies for text feature extraction. Using one-hot encoding on feature extraction is a kind of BOW (bag of word). The basic idea is to ignore the order of words and represent a document by the frequency of each word in the document. If the size of the dictionary is n and we want to represent a certain word is on k position, we can create an n-dimensionality vector, whose the k dimensionality would be set to 1 while the other would be set to 0. And the vector of the sample document can be expressed by directly added vectors of each word in the document.

For example, there are two sentence: 1) This is an example. 2) This is another example. The dictionary of the above sentence is: 'this', 'is', 'a', 'another', 'example'. The feature vector of sentence 1 is [1, 1, 1, 0, 1] and that of sentence 2 is [1, 1, 0, 1, 1].

TF-IDF is a common weighing technology for data mining and information retrieval [42]. TF means the term frequency, and the more frequently the term appears in the document, the larger TF value would be. As the following formula shows, $n_{i,j}$ is the count of word i in document j and the denominator is the sum of the number of occurrence of all words in the document d_j.

$$TF_{i,\ j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{3.1}$$

IDF means the inverse document frequency, and the less frequently the term appears in the corpus, the larger IDF value would be. The following is the calculation of IDF, where $|D|$ is the number of documents in the corpus and $\{j \ : \ t_i \ \in d_j\}$ means the number of documents which contain term i. The addition of 1 is to prevent the denominator being zero when the term i is not in the corpus.

$$IDF_i = \log \frac{|D|}{\{j \ : \ t_i \ \in d_j\} \ + \ 1} \tag{3.2}$$

The value of TF-IDF can represent the importance of a term for the corpus, which can be calculate as the product of $TF_{i,j}$ and IDF_i

$$TF - IDF_{i,j} = TF_{i, \ j} \ \times IDF_i \tag{3.3}$$

Both one-hot encoding and TF-IDF are simple and fast ways to extract the text features, but they all focus on the frequency of the words and ignore the sequence of the words. It assumes that the word are independent of each other and neglect the mutual relation between the words. Additionally, the obtained features are discrete and sparse.

Since the shortage of the above techniques, word-embedding technique has been proposed. Word-embedding is a representation of word, and is technique that project words to real-number vectors, where words with similar meaning s have similar representations. Word2vec algorithm[43, 44], as one of the word-embedding techniques, can learn the word associations by neural network from a large corpus [45]. It use Besides semantic similar words should be encoded into neighboring regions, expected word vectors should also support simple vector operations, which project semantic operations to vector operations. For example, $vec(king)+vec(woman)-vec(man) = vec(queen)$ [46]. Moreover, to add the influence of context on the word rather than only analysis the text on word-dimensionality, doc2vec method was proposed by Le and T [47]. Doc2vec is based on word2vec method but adds a paragraph vector for training, which means that when training a sentence or a document, each time predicting the probability of the word, it uses the semantics of the entire sentence.

3.1.2 K-means Clustering

After extracting features of text, clustering algorithms such as K-means cluster can be applied on them. K-means is a partitioning based algorithm proposed by McQUeen [48]. The basic idea of K-means is that for a given sample set, it is expected to be divided into K clusters according to the distance between the samples and the centers. The points in each cluster are expected to be as closed as possible with each other and the distance among the clusters is expected to be large enough.

The main steps of K-means algorithm are as follows [49]:

Algorithm 1: K-means algorithm

Input: Sample set
Output: Clustered result
Random choose the initial centers of K clusters;
Initialize k samples as the initial cluster centers $a = a_1, a_2, ...a_k$;
while *cluster membership does not stabilize* **do**
    for *each sample* x_i **do**
        Calculate the distance to each cluster center, and assign it to its closets center;
    end
    for *each cluster* a_j **do**
        Re-calculate and update its cluster center $a_j = \frac{1}{|c_i|} \sum_{x \in c_i} x$;
    end
end

Additionally, in K-Means, Euclidean Metric or cosine distance is often used to calculate the distance between each sample.

3.2 Probabilistic-Based Clustering Algorithm

Topic modeling is a probability-based clustering tool for data mining and discovering the relationships in text documents [39, 50]. Traditional topic models such as Latent Semantic Index (LSI) and Probabilistic Latent Semantic Analysis (PLSA) have been applied in information retrieval, NLP, machine learning for text and related areas [51]. Those algorithms assume each document is generated by repeating the steps that choosing a topic based on a certain probabilistic then choosing the word from the topic with a certain probabilistic. Combined with Bayesian probability framework, Latent Dirichlet Allocation (LDA) was proposed [23], and a more advanced model Biterm Topic Model (BTM) is also produced based on LDA. The followings are the derivations of LDA and BTM [52].

3.2.1 Mathematical Basis

To understand the principle of LDA and BTM, there are some mathematical basis should be known in advance.

LDA proposed based on Bayesian model, and the basic process of Bayesian parameter estimation is

$$prior\ distribution\ +\ data(likelihood)\ =\ posterior\ distribution$$

Using coin toss as an example, assume the probability of the coin being heads up is p,

repeat n times trials, the probability of k times being heads up can be represented as a binomial distribution.

$$Binom(k|n,p) \;=\; \binom{n}{k} p^k (1-p)^{n-k} \tag{3.4}$$

To estimate the parameters, besides the likelihood, the prior and the posterior should also be known. Prior probability is a estimate probability before the event happen while the posterior probability is obtained by correcting the prior probability based on a certain evidence. In Bayesian probability theory, if the posterior distribution is the same as the prior probability distribution, the prior and posterior are called conjugate distribution [53]. Regard the above binomial distribution as the likelihood, its conjugate distribution is Beta distribution. The general formula for Beta distribution is as follows:

$$Beta(p|\alpha,\beta) \;=\; \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \tag{3.5}$$

And the posterior probability distribution is :

$$\begin{aligned} P(p|n,k,\alpha,\beta) &\propto P(k|n,p)P(p|\alpha,\beta) \\ &= P(k|n,p)P(p|\alpha,\beta) \\ &= Binom(k|n,p)Beta(p|\alpha,\beta) \\ &= \binom{n}{k} p^k (1-p)^{n-k} \times \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \\ &\propto p^{k+\alpha-1}(1-p)^{n-k+\beta-1} \end{aligned} \tag{3.6}$$

Normalization the above equations, the posterior probability is as follows, and the posterior probability distribution is exactly the Beta distribution.

$$P(p|n,k,\alpha,\beta) \;=\; \frac{\Gamma(\alpha+\beta+n)}{\Gamma(\alpha+k)\Gamma(\beta+n-k)} p^{k+\alpha-1}(1-p)^{n-k+\beta-1} \tag{3.7}$$

Therefore, the intuitive expression of the above Bayesian analysis process is:

$$Beta(p|\alpha,\beta) \;+\; BinomCount(k,n-k) \;=\; Beta(p|\alpha+k,\beta+n-k) \tag{3.8}$$

Extend the above two-dimensional process to multi-dimensional. The multinomial distribution is the extension of the binomial distribution, and Dirichlet distribution is the conjugate prior of multinomial distribution, which is also the multivariate generalization of the Beta distribution [54].

Assume still do n times trials, each time has m results, the multinomial distribution is $multi(\vec{m}|n,\vec{p})$. $\vec{\alpha} \;=\; (\alpha_1,\alpha_2,...,\alpha_k)$ is the parameter of the Dirichlet distribution, the

Dirichlet distribution can be represented as $Dirichlet(\vec{p}|\vec{\alpha})$. The same as Beta-Binomial conjugate, Dirichlet-Multinomial conjugate can be described as:

$$Dirichlet(\vec{p}|\vec{\alpha}) \ + \ MultiCount(\vec{m}) \ = \ Dirichlet(\vec{p}|\vec{\alpha}+\vec{m}) \tag{3.9}$$

Additionally, for a nature of Dirichlet distribution, the mathematical expectation of the distribution $\vec{p} \ \sim \ Dir(\vec{t}|\vec{\alpha})$ is:

$$\mathbb{E}(\vec{p}) \ = \ (\frac{\alpha_1}{\sum_{i=1}^{K}\alpha_i}, \frac{\alpha_2}{\sum_{i=1}^{K}\alpha_i}, ..., \frac{\alpha_K}{\sum_{i=1}^{K}\alpha_i},) \tag{3.10}$$

3.2.2 LDA

As claimed in [23], LDA is proposed as a generative probabilistic model for collecting discrete data. The physical process of LDA text modeling can be described as a dice game.

Algorithm 2: LDA Topic Model

1 There are two jars contains doc-topic dices and topic-word dices respectively;
2 Randomly pick K topic-word dices and number them from 1 to K;
3 Before generating a new document, randomly pick a doc-topic dice, and repeat the following steps to generate the words in the document;
4 **repeat**
5 | Throw this doc-topic dice and get a topic z;
6 | Pick the number z dice from K topic-word dices, then throw it to get a word;
7 **until** *finish the document*;

Assume there are M documents in the corpus, all the word and their topics in the document can be denoted as vectors, in which $\vec{w}_m$ denote the words in the m^{th} document and $\vec{z}_m$ denote the corresponding topics of the words.

$$\vec{w} \ = \ (\vec{w}_1, ..., \vec{w}_M)$$
$$\vec{z} \ = \ (\vec{z}_1, ..., \vec{z}_M)$$

The figured process of LDA model is as figure 3.1 shows. $\vec{\theta}_m$ in the doc-topic dice and $\vec{\varphi}_k$ in the topic-word dice are the parameters in the model, which correspond to multinomial distribution.

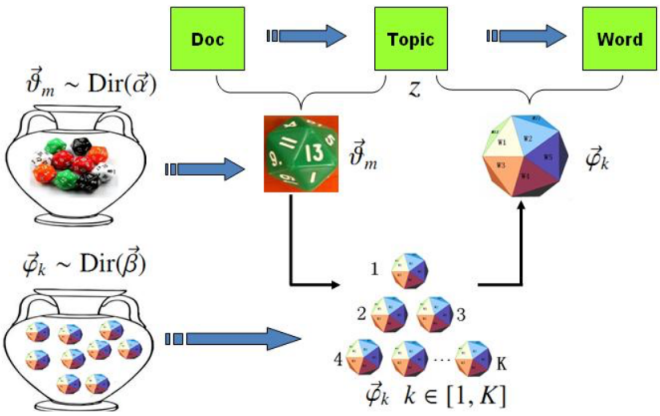


Figure 3.1: LDA Model [1]

Figure 3.2 is the probabilistic graphic model of LDA. It can be broken down to two physical processes:

1. $\vec{\alpha} \rightarrow \vec{\theta}_m \rightarrow z_{m,n}$, denotes the process that when generating the m^{th} document, pick a doc-topic dice $\vec{\theta}_m$, then use this dice to generate the topics $z_{m,n}$ of the n^{th} word;
2. $\vec{\beta} \rightarrow \vec{\varphi}_k \rightarrow w_{m,n}|k = z_{m,n}$, denotes the process using the dice numbered $k = z_{m,n}$ from K topic-word dices $\vec{\varphi}_k$ to generate the word $w_{m,n}$, which means generate the n^{th} word in the m^th document.

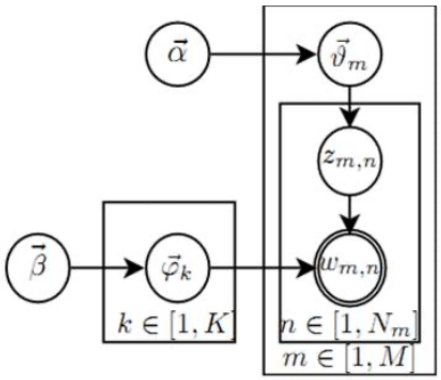


Figure 3.2: Probabilistic Graphic Model of LDA [1]

For the first process, $\vec{\alpha} \rightarrow \vec{\theta}_m$ corresponds to Dirichlet distribution and $\vec{\theta}_m \rightarrow \vec{z}_m$ corresponds to Multinomial distribution. Thus, this process is a conjugate structure:

$$\vec{\alpha} \underbrace{\longrightarrow}_{Dirichlet} \vec{\theta} \underbrace{\longrightarrow}_{Multinomial} \vec{z}_m$$

Based on little computing, we can get:

$$p(\vec{z}_m|\vec{\alpha}) \;=\; \frac{\Delta(\vec{n}_m+\vec{\alpha})}{\Delta(\vec{\alpha})}$$

in which $\vec{n}_m \;=\; (n_m^{(1)},...,n_m^{(k)}))$, $n_m^{(k)}$ is the number of words that generated from k^{th} topic. Moreover, utilizing the Dirichlet-Multinomial conjugate, the posterior distribution of parameter $\vec{\theta}_m$ is:

$$Dir(\vec{\theta}_m|\vec{n}_m+\vec{\alpha})$$

The generations of topics of M documents are independent, thus there are M independent Dirichlet-Multinomial structures. And the probability of the topic generation is:

$$\begin{aligned} p(\vec{z}|\vec{\alpha}) \;&=\; \prod_{m=1}^{M} p(\vec{z}_m|\vec{\alpha}) \\ &=\; \prod_{m=1}^{M} \frac{\Delta(\vec{n}_m+\vec{\alpha})}{\Delta(\vec{\alpha})} \end{aligned} \tag{3.11}$$

Similarly, for the second process $\vec{\beta} \;\rightarrow\; \vec{\varphi}_k \;\rightarrow\; \vec{w}_{(k)}$, $\vec{\beta} \;\rightarrow\; \vec{\varphi}_k$ corresponds to Dirichlet distribution and $\vec{\varphi}_k \;\rightarrow\; \vec{w}_{(k)}$ corresponds to Multinomial distribution:

$$\vec{\beta} \underbrace{\longrightarrow}_{Dirichlet} \vec{\varphi}_k \underbrace{\longrightarrow}_{Multinomial} \vec{w}_{(k)}$$

Based on the computing and this conjugate structure, the posterior distribution of parameter φ_k is also a Dirichlet distribution $Dir(\vec{\varphi}_k|\vec{n}_k+\vec{\beta})$. The generations of words from K topics in the corpus are independent from each other, thus there are K independent Dirichlet-Multinomial conjugate structures. And the probability of the generation of the words in the whole corpus is [52]:

$$\begin{aligned} p(\vec{w}|\vec{z},\vec{\beta}) \;&=\; p(\vec{w}'|\vec{z}',\vec{\beta}) \\ &=\; \prod_{k=1}^{K} p(\vec{w}_{(k)}|\vec{z}_{(k)}\vec{\beta}) \\ &=\; \prod_{k=1}^{K} \frac{\Delta(\vec{n}_k+\vec{\beta})}{\Delta(\vec{\beta})} \end{aligned} \tag{3.12}$$

Combined equation 3.11 with 3.12, the probability of the generation of the whole corpus

is [52]:

$$\begin{aligned} p(\vec{w}, \vec{z}|\vec{\alpha}, \vec{\beta}) &= p(\vec{w}|\vec{z}, \vec{\beta})p(\vec{z}|\vec{\alpha}) \\ &= \prod_{k=1}^{K} \frac{\Delta(\vec{n}_k + \vec{\beta})}{\Delta(\vec{\beta})} \prod_{m=1}^{M} \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{\alpha})} \end{aligned} \tag{3.13}$$

With the joint distribution $p(\vec{w}, \vec{z})$, Gibbs sampling can be used to estimate the posterior distribution by collecting the samples [55]. Gibbs sampling is a Markov chain Monte Carlo (MCMC) algorithm used when direct sampling is difficult[56], which is regarded as a simulation tool to acquire samples from non-normalized joint density function[57].

The $\vec{w}$ is the observed known data, while $\vec{z}$ is the implicit variable. Thus, the distribution that should be sampled is $p(\vec{z}|\vec{w})$. Mark the topic of the i^{th} word in corpus $\vec{z}$ as z_i, where $i = (m, n)$ and corresponds to the n^{th} word in the m^{th} document and $\neg i$ means get rid of word i. According to the requirement of Gibbs sampling algorithm, we should get the corresponding condition distribution $p(z_i = k|\vec{z}_{\neg i}, \vec{w})$ of each arbitrary coordinate axis i. After getting rid of the i^{th} word, the Dirichlet-Multinomial conjugate will not be changed. Thus, the posterior distribution of both $\vec{\theta}_m$ and $\vec{\varphi}_k$ still are Dirichlet distribution:

$$\begin{aligned} p(\vec{\theta}_m|\vec{z}_{\neg i}, \vec{w}_{\neg i}) &= Dir(\vec{\theta}_m|\vec{n}_{m,\neg i} + \vec{\alpha}) \\ p(\vec{\varphi}_k|\vec{z}_{\neg i}, \vec{w}_{\neg i}) &= Dir(\vec{\varphi}_k|\vec{n}_{k,\neg i} + \vec{\beta}) \end{aligned} \tag{3.14}$$

Assume the observed word $w_i = t$, based on the Bayesian rule(the conditional probability is proportional to joint probability) and combined with the above equation 3.14 [52]:

$$\begin{aligned} p(z_i = k|\vec{z}_{\neg i}, \vec{w}) &\propto p(z_i = k, w_i = t|\vec{z}_{\neg i}, \vec{w}_{\neg i}) \\ &= \mathbb{E}(\theta_{mk}) \cdot \mathbb{E}(\varphi_{kt}) \\ &= \hat{\theta}_{mk} \cdot \hat{\varphi}_{kt} \end{aligned} \tag{3.15}$$

Additionally, knowing the posterior distribution, the way to estimate the parameter is to use the mean value of this parameter in the posterior distribution. The expectation of Dirichlet distribution, as equation 3.10, can estimate each parameter $\hat{\theta}_{mk}$ and $\hat{\varphi}_{kt}$ [52]:

$$\begin{aligned} \hat{\theta}_{mk} &= \frac{n_{m,\neg i}^{(k)} + \alpha_k}{\sum_{k=1}^{K}(n_{m,\neg i}^{(k)} + \alpha_k)} \\ \hat{\varphi}_{kt} &= \frac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^{V}(n_{k,\neg i}^{(t)} + \beta_t)} \end{aligned} \tag{3.16}$$

The final Gibbs sampling formula for LDA model is as follows, in which the right side is

$p(topic|doc) \cdot p(word|topic)$. This Gibbs sampling can help to sample the topics of all the words in the corpus. And the Gibbs sampling algorithm can be used to train or inference the LDA model.

$$p(z_i = k|\vec{z}_{\neg i}, \vec{w}) \propto \frac{n^{(k)}_{m,\neg i} + \alpha_k}{\sum_{k=1}^{K}(n^{(k)}_{m,\neg i} + \alpha_k)} \cdot \frac{n^{(t)}_{k,\neg i} + \beta_t}{\sum_{t=1}^{V}(n^{(t)}_{k,\neg i} + \beta_t)} \tag{3.17}$$

3.2.3 BTM

The main difference between BTM and LDA is that LDA modeling each word in the corpus while BTM modeling each biterms. 'biterm' denotes the word pair co-occuring in the text[15]. For example, document (w_1, w_2, w_3) will generate three biterms $\{(w_1, w_2), (w_1, w_3), (w_2, w_3)\}$. The formula derivation of BTM is basically the same as LDA. The difference of LDA and BTM is showed as the graphical representation figure 3.2.3

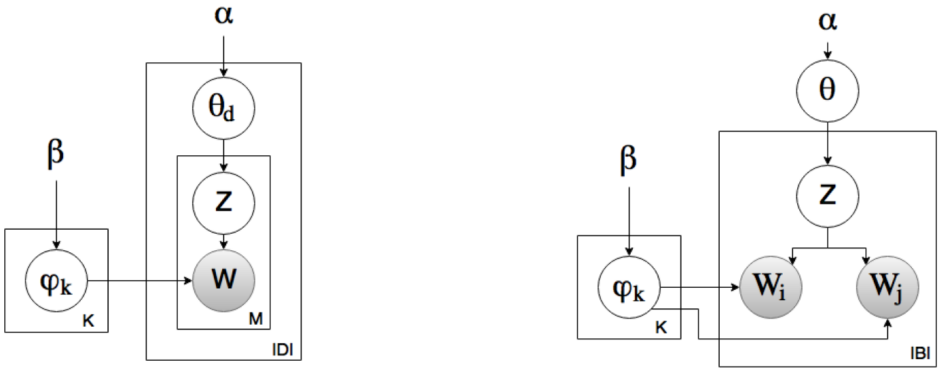


Figure 3.3: Graphical representation of LDA(left) & BTM(right) [2]

The generation of the corpus in BTM is described as follows[15]:

1. Draw $\theta \sim Dirichlet(\alpha)$
2. For each topic $k \in [1, K]$
   (a) draw $\phi_k \sim Dirichlet(\beta)$
3. For each biterm $b_i \in B$
   (a) draw $z_i \sim Multinomial(\theta)$
   (b) draw $w_i, w_j \sim Multinomial(\phi_{z_i})$

Assume the i^{th} biterm in the corpus as $b_i = \{w_i, w_j\}$, in which the words $w_i = t_1$, $w_j = t_2$. $z_{\neg i}$ denotes the topics for all biterms except b_i, $n^{(k)}_{\neg i}$ is the number of biterms assigned to

topic k except b_i, and $n_{\neg i,k}^{(w)}$ denotes the number of times that word w assigned to topic k except words in b_i [15]. The extend of the Gibbs sampling formula for BTM is:

$$\begin{aligned} p(z_i = k|\vec{z}_{\neg i}, \vec{b}) &\propto \theta_k \cdot \varphi_{k,w_i} \cdot \varphi_{k,w_j} \\ &= \frac{n_{\neg i}^{(k)} + \alpha_k}{\sum_{k=1}^{K}(n_{\neg i}^{(k)} + \alpha_k)} \cdot \frac{n_{\neg i,k}^{(t_1)} + \beta_{t_1}}{\sum_{t_1=1}^{V}(n_{\neg i,k}^{(t_1)} + \beta_{t_1})} \cdot \frac{n_{\neg i,k}^{(t_2)} + \beta_{t_2}}{\sum_{t_2=1}^{V}(n_{\neg i,k}^{(t_2)} + \beta_{t_2})} \end{aligned} \tag{3.18}$$

Chapter 4

Design

This chapter discusses the whole design of the system, containing the overall design and the detailed description of each part of the system.

4.1 Overall Design

To meet the requirements of the project, the general design of the system can be divided into three parts: input data, clustering model and output result. COVID-19 related documents from Twitter would be regarded as the input dataset of the system. The system can be treated as a black box and the NLP topic model and additional analysis of the generated topics can be the representation for this black box. Additionally, the output is the analyzing result from the potential topics of the input data, which is ultimately what we want in our system.

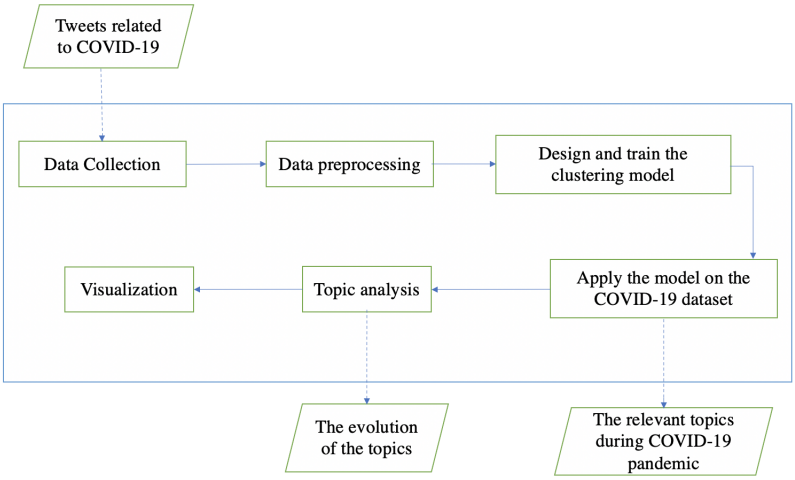


Figure 4.1: Overall Design

The above figure 4.1 shows the general graphical structure of the system design, where the rectangles represent the main steps, and the parallelograms with the dashed arrow represent the input or products of those steps. And the main process for the implementation is: 1) data collection, 2) data preprocessing, 3) select the appropriate model to cluster the texts documents, 4) applying the model to the dataset and generate the result, 5) topic analysis, and 6) show the visualization result.

4.2 Basic Components Design

The design of the basic components in the system is discussed in this section and it will include the functional design of each part and why it suit the project.

4.2.1 Data Collection

Twitter data is chosen as the input social media data of the whole system. Crawling Twitter data can be implemented by an official Twitter API or directly download from an open source website. When designing the data collector, four requirements should be considered:

1. The collected Tweets should be written in English since this project focus on English Tweets analysis.
2. Besides the main body of the Tweet, the username, Tweet created time and location should also be collected to complete the dataset.
3. The created time of the collected Tweet should be during the COVID-19 pandemic.
4. The contents of the collected Tweets should be related to COVID-19.

4.2.2 Data Preprocessing

The collected data can not be directly used in the clustering model since the structures of social media data mainly from unstructured user grammar so that structures are complex and irregular [58]. Therefore, after collecting data at stage 1, the original dataset should be preprocessed to make it suits for the clustering model.

The following shows the basic steps of preprocessing:

1. Structure unification: the media social data may be from different sources with different data structures. If those data should be regarded as the input of the later model training, there is a uniform structure requiring. In addition to the necessary information mentioned before, the dataset can also contain information that would not be used at the current stage, in case need it later.
2. Data regularization: the collected Twitter data might contain some noise information such as emoji, URL, pictures, and videos, but we primarily focus on the pure text. So, in this step, that noise information would be ignored, and the emoji would be replaced by the corresponding regular text.
3. Stemming: since the algorithm will be modeling on the words in the document, the unification of a word is significant. Stemming is unifying the various forms of a word into one common representation [59]. For example, words: 'interesting', 'interested', 'interests' should all be represented as 'interest'.

4. Removing: because of the particularity of text data, there are some features might influence the accuracy of the clustering model. For example, stop words such as 'are', 'and' are not useful for the analysis of the documents and should be removed. Also, social network users prefer to use slang in their texts, such as '4u' means 'for you', '2day' means 'today'. These slangs should also be filtered and translated to their basic form. Moreover, punctuations should also be deleted.

5. Tokenization: the task of tokenization is to split the text data into units (tokens), which is one of the early processing steps of NLP [60]. There are three types of separation of texts: split texts into words, into single characters and split texts into sub-word (n-gram). Since our algorithm will be modeling on words, we adopt the word-tokens strategy.

6. Data filter: after the above steps, if the document remains less than three words, it could be treated as useless data, which should be filtered.

4.2.3 Clustering Model

Some NLP topic models can be used as the clustering model in our system to excavate the main topics of the input documents. The followings are the functional requirements of the model:

1. The filtered data should be generated from the clustering model and should be assigned into clusters.

2. Each cluster should have an explainable topic.

4.2.4 Visualization

The visualization result mainly focuses on using interactive graphics to make user make sense about the science data on some potential features such as the relationship and the trend [61]. For different purpose or different data type, there are various techniques for visualization including basic 2D visualization such as bar chart, 3D visualization and dynamic visualization. In this project, there are two types of visualization should be shown:

1. The visualization result should show the details of the topics, for example, display the top words in each topic.

2. The visualization result should show the topic distribution and variation during the pandemic.

Word cloud has become a visually attractive and straightforward method of text visualization, which display the highest frequency words in a text to represent the overview of the text [62]. As figure 4.2 shows, it is satisfied for topic visualization. Traditional bar

chart can distinct demonstrate the distribution of the topics, and the example is as figure 4.3.

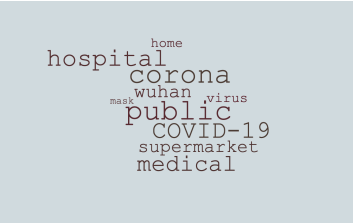


Figure 4.2: Word Cloud

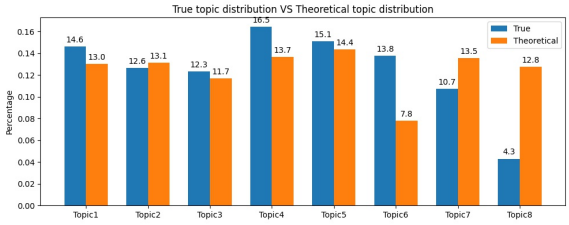


Figure 4.3: Bar Chart

4.3 Algorithm Design Specification

In Chapter 2, there are three different algorithms, K-means, LDA and BTM, are introduced that can be used in this project for text clustering. In the early stage of the project, the exploration of suitable clustering model started from K-means, then tried LDA and BTM respectively, and finally designed our own model based on BTM. This section focuses on the specific design of our model Triterm Topic Model(TTM). Since TTM is a modified BTM, the design details of BTM are also discussed.

4.3.1 BTM

The purpose is to train the parameters $\vec{\varphi}_1, ..., \vec{\varphi}_K$ and $\vec{\theta}_1, ..., \vec{\theta}_M$, and using the model to compute a new topic-word distribution $\vec{\theta}_new$ for a new document. A plentiful enough corpus will be used as the training dataset to inference the Tweets related to COVID-19.

The training process for BTM model is using Gibbs sampling obtained the (z, b) samples in the corpus, and estimate all the parameters based on the samples. The training algorithm is as follows [15], where n_k denotes to the number of biterms assigned to topic k, $n_{w|k}$ denotes to the number of times that word w assigned to topic k, and the topic-word matrix $\vec{\varphi}$ is the model:

Algorithm 3: Gibbs sampling for BTM training

Input: topic number K, α and β, biterm set B
Output: $\vec{\theta}$, $\vec{\varphi}$

```
Random assign a topic z to all the biterms;
for iter = 1 to N_iter do
    foreach biterm b_i = (w_i, w_j) ∈ B do
        Draw topic k from P(z_i | z_¬i, B);
        Update n_k, n_{w_i|k} and n_{w_j|k};
    end
end
compute the parameters θ and φ;
```

The inference process is similar to the training process. Keep the $\hat{\varphi}_{kt}$ in the Gibbs sampling formula unchanged from training process, then estimate the topic distribution for the new document.

Algorithm 4: Gibbs sampling for BTM inference

Input: topic number K, new document d_new, α and β, new biterm set B_{new}, phi
Output: $\vec{\theta}_new$

```
Randomly assign a topic z to all the biterms in d_new;
for iter = 1 to N_iter do
    foreach biterm b_i = (w_i, w_j) ∈ B_new do
        re-sampling the topic of b_i by Gibbs sampling formula
    end
end
compute the topic distribution of the new document, which is θ⃗_new;
```

4.3.2 TTM

As illustrated in Section 2.2, our model Triterm Topic Model(TTM) extends modeling the biterms to triterms in the short text. 'triterm' denotes the three-word group co-occuring in the short text. For example, document(w_1, w_2, w_3, w_4) will generate four triterms$\{(w_1, w_2, w_3), (w_1, w_3, w_4), (w_1, w_2, w_4), (w_2, w_3, w_4)\}$. The graphical representation of TTM is as the following figure 4.4 shows.

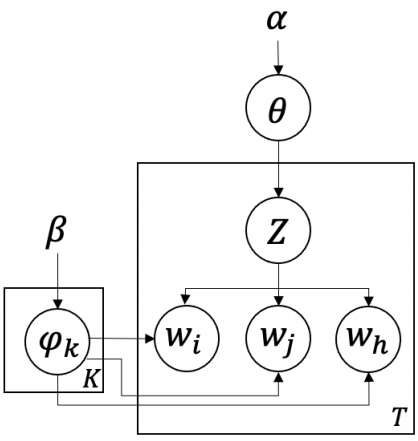


Figure 4.4: Graphical representation of TTM

The generative process of TTM is:

1. Draw $\theta \sim Dirichlet(\alpha)$
2. For each topic $k \in [1, K]$
   (a) draw $\phi_k \sim Dirichlet(\beta)$
3. For each biterm $b_i \in B$

(a) draw $z_i \sim Multinomial(\theta)$

(b) draw $w_i, w_j, w_h \sim Multinomial(\phi_{z_i})$

The training and inference processes are very similar to BTM. For training a TTM, randomly assign a topic to each triterm. In each iteration, update the topic for each triterm by the Gibbs sampling formula and when the Gibbs sampling converges, compute the parameter $\vec{\theta}$ and $\vec{\varphi}$ then get the model. And the Gibbs sampling formula for TTM is:

$$p(z_i = k|\vec{z}_{\neg i}, \vec{t}) \ \propto \ \theta_k \ \cdot \ \varphi_{k,w_i} \ \cdot \ \varphi_{k,w_j} \ \cdot \ \varphi_{k,w_h} \tag{4.1}$$

Algorithm 5: Gibbs sampling for TTM training

Input: topic number K, α and β, triterm set T
Output: $\vec{\theta}$, $\vec{\varphi}$

```
Random assign a topic z to all the triterms;
for iter = 1 to N_iter do
    foreach triterm t_i = (w_i, w_j, w_h) ∈ T do
        Draw topic k from P(z_i | z_¬i, T);
        Update n_k, n_{w_i|k}, n_{w_j|k} and n_{w_h|k};
    end
end
compute the parameters θ⃗ and φ⃗;
```

Also the formula for parameter updating is as follows, where $n_{w|k}$ denotes the number of times that word w assigned to topic k, n_k denotes to the number of triterms in topic k and N_T is the number of all triterms.

$$\begin{aligned} \varphi_{k,w} &= \frac{n_{w|k} + \beta}{\sum_w n_{w|k} + W\beta} \\ \theta_k &= \frac{n_k + \alpha}{N_T + K\alpha} \end{aligned} \tag{4.2}$$

The process of TTM inference is designed as follows:

Algorithm 6: Gibbs sampling for TTM inference

Input: topic number K, new document d_{new}, new titerm set T_{new}, α and β, $\vec{\phi}$
Output: $\vec{\theta}_{new}$

```
Random assign a topic z to all the triterms in d_new;
for iter = 1 to N_iter do
    foreach titerm t_i = (w_i, w_j, w_h) ∈ T_new do
        re-sampling the topic of t_i by Gibbs sampling formula
    end
end
compute the topic distribution of the new document, which is θ⃗_new;
```

Chapter 5

Implementation

This chapter discusses the implementation details and the process of data collection, preprocessing, and algorithms. And we use python to implement our algorithms.

5.1 Data Collection

As mentioned in Section 4.2.1, COVID-19 related Twitter information should be collected. Twitter provides an official API for a developers to gather data from Twitter. After successfully applying for Twitter developer account, 'consumer key', 'consumer secret', 'access token' and 'access secret' will be generated to be used to connect to Twitter API and access Twitter data. A python library named Tweepy can help the implementation of crawler technology on Twitter. After authentication and biding the key, we can acquire the Twitter information that is needed in this project, and can procure Tweets by searching data, keyword, location, or their combination.

The codes for collecting Twitter data including Tweet, created time, location, username, and ID of Tweet by searching keyword 'COVID-19' are provided in the source code. But still encountered some problems.

1. The standard Twitter API only can be used to search Tweets in the last seven days, so there is no way to get earlier data by using it.

2. To manage thousands of request to Twitter API, the number of requests limits are set, and the most common request limit interval is 15 minutes [63]. The maximum number of requests is 150 times every 15 minutes, which means if the API was repeatedly called beyond the limit, the program would sleep for a while. It significantly reduces the efficiency of data collection. During the implementation, because of the limit, the program ran for over an hour even can only get the Tweets of 20 minutes (for example, get the Tweets from 11:40 to 12:00).

3. When searching data by date using Tweepy, developers can only set the date instead of the specific time. Therefore, if the program was interrupted, the subsequent data cannot get back by setting the specific time.

4. Due to the international restriction, the domestic connection to Twitter API is unstable. Connection exception would be thrown after connecting for a while and the program would stop. Combined with the second limit, when recalling the program, the previous record cannot be saved and only can get the data from scratch.

These restrictions cannot be avoided. In order to ensure the smooth progress of the project, an open-sourced dataset was found to suit this project. The dataset can be accessed in Github. This dataset was collected by keywords tracking and the following figure 5.1 is a sample of the keywords list [64].

Tracked since	Keyword
1/21/2020	Coronavirus; Corona; CDC; Ncov; Wuhan; Outbreak; China
1/22/2020	Koronavirus; Wuhancoronavirus; Wuhanlockdown; N95; Kungflu; Epidemic; Sinophobia
2/16/2020	Covid-19
3/2/2020	Corona virus
3/6/2020	Covid19; Sars-cov-2
3/8/2020	COVID—19
3/12/2020	COVD; Pandemic
3/13/2020	Coronapocalypse; CancelEverything; Coronials; SocialDistancing
3/14/2020	Panic buying; DuringMy14DayQuarantine; Panic shopping; InMyQuarantineSurvivalKit
3/16/2020	chinese virus; stayhomechallenge; DontBeASpreader; lockdown
3/18/2020	shelteringinplace; staysafestayhome; trumppandemic; flatten the curve
3/19/2020	PPEshortage; saferathome; stayathome
3/21/2020	GetMePPE
3/26/2020	covidiot
3/28/2020	epitwitter
3/31/2020	Pandemie

Figure 5.1: Keywords Sample

Additionally, many Twitter datasets only contained the Tweet ID instead of the body text, so we also provided the code to transfer the ID to the text in Twitter.

5.2 Preprocessing

Since there are many defects for raw data to be analyzed such as inconsistencies, noise and missing values [34], data preprocessing is a pivotal stage in text clustering framework, and the same as feature extraction and feature selection, preprocessing also has a substantial impact on the effectiveness of cluster model [35]. According to the extensive research [35], the accuracy of classification or cluster can be improved by choosing suitable preprocessing tasks. Some preprocessing techniques refer to [65]

5.2.1 Structure Unification

To unify the metadata, for each Tweet, three keys 'Time', 'Location' and 'Text' are set. The unified form for Time is 'yyyy-MM-dd', and for those Tweets without location information, the location key will be set to Null.

5.2.2 Data Regularization

As illustrated in Section 4.2.2, to get the pure text for the subsequent clustering, regularization should be used on the original Tweets. There is no common regularization rule on all text preprocessing to suit all tasks since the variety of the structure for text data. Based on observation of our Twitter data, the followings are the primary regularization techniques we use for preprocessing the Tweets:

1. Remove URL: some tweets might contain URL started with "http" or "www" which would not be needed in the later clustering and should remove them. The regular expression of URLs is "(www\.[^\s]+)|(https?://[^\s]+)".
2. Remove @user: people would like to mention other users in tweets by @ which have no contribution to the analysis and should be removed. The regular expression of it is "@[^\s]+".
3. Remove #: in front of the hashtag, there is a pound sign which should be removed. And its regular expression is "#([^\s]+)".
4. Remove multiple marks: some users prefer using multiple marks to express their feelings and would not have meanings for our project. For example, the regular expression for multiple exclamation marks is "(\!)\1+", and other marks have a similar expression.
5. Remove emoticons: the emoticons have no contribution for NLP. The regular expression example for them is ": $-D| = D| : P$".
6. Remove emojis: there is a library named emoji (version 0.6.0) in Python can import the whole emoji list and can replace emojis into regular text by a function call. Then the emojis can be removed by using the regularization of the replaced text.

5.2.3 Further Preprocessing Techniques

To implement the designed preprocessing requirements, an NLP package in python named NLTK) can be installed. The techniques in NLTK that we can use including removing stopwords, stemming, removing all punctuation, lowercase all characters, and tokenization, which are all mentioned in section 4.2.2. Additionally, to replace slang words and abbreviations with their equivalent, a file named slang.txt which contains the common slang words and their corresponding written words is imported in our project to help to check the slang.

5.3 Algorithm Implementation

As designed in section 4.3, in this project, we will implement several algorithms including both word-embedding-based clustering algorithms and probabilistic-based clustering algorithm from feature extraction technique TF-IDF combined with basic clustering model K-means, to advanced topic model BTM.

Both K-means and LDA can be implemented by python libraries. Library sklearn provides APIs that can help us to achieve TF-IDF and K-means. Load the train texts and prepared the basic requirement such as dictionary and corpus, then K-means text clustering algorithm can be implemented. Library Genism, an open-sourced python package, can be used to implement unsupervised learning of the latent topic expression from unstructured raw text data, which provides the algorithm of LDA topic model. After setting the parameters (the number of topic K) and preparing the corpus and dictionary, LDA can be trained.

Though the above two algorithms are simple to implement, at the middle stage of the project, based on some research and experiments, LDA did not perform well on short text data like our Twitter dataset, thus we should begin to attempt another probabilistic based clustering algorithm BTM.

Algorithm 7: Implementation of BTM Training

Input: documents set $D = d_1, .., d_n$, topic number K, α and β
Output: the topic-word distribution $\vec{\theta}$ and $\vec{\varphi}$

Preprocess each documents in D, build the vocabulary set and encode the documents with the indexes of words;
Initialize biterm set $B =$ **foreach** *document* $d \in D$ **do**
    Generate all biterms b_d in d;
    Append biterms b_d to biterm set B
end
Randomly assign a topic z to all biterms in B;
for *iter* $= 1$ *to* N_{iter} **do**
    foreach *biterm* $b_i \in B$ **do**
        Compute and update $\theta_z = \frac{n_z + \alpha}{|B| + K\alpha}$;
        foreach $w \in b_i$ **do**
            Compute and update $\varphi_{w|z} = \frac{n_{w|z} + \beta}{\sum_w n_{w|z} + W\beta}$;
        end
        Compute $P(z|B) = \theta_z \cdot \varphi_{w_i|z} \cdot \varphi_{w_j|z}$;
        Re-assign topic z to biterm b_i;
        Update n_z, $n_{w_i|z}$ and $n_{w_j|z}$;
    end
    Obtain the final $\vec{\theta}$ and $\vec{\varphi}$;
end

And the algorithm for TTM is very similar to BTM:

Algorithm 8: Implementation of TTM Training

Input: documents set $D = d_1, .., d_n$, topic number K, α and β
Output: the doc-topic distribution $\vec{\theta}$ and topic-word distribution $\vec{\varphi}$

Preprocess each documents in D, build the vocabulary set and encode the documents with the indexes of words;
Initialize triterm set $T =$ **foreach** *document* $d \in D$ **do**
    Generate all biterms b_d in d;
    Append triterms b_d to biterm set T
end
Randomly assign a topic z to all triterms in T;
for $iter = 1$ *to* N_{iter} **do**
    foreach *triterm* $t_i \in T$ **do**
        Compute and update $\theta_z = \frac{n_z + \alpha}{|T| + K\alpha}$;
        foreach $w \in t_i$ **do**
            Compute and update $\varphi_{w|z} = \frac{n_{w|z} + \beta}{\sum_w n_{w|z} + W\beta}$;
        end
        Compute $P(z|T) = \theta_z \cdot \varphi_{w_i|z} \cdot \varphi_{w_j|z} \cdot \varphi_{w_h|z}$;
        Re-assign topic z to triterm t_i;
        Update n_z, $n_{w_i|z}$, $n_{w_j|z}$ and $n_{w_h|z}$;
    end
    Obtain the final $\vec{\theta}$ and $\vec{\varphi}$;
end

With a new document put into the trained TTM, the inference algorithm is as follows:

Algorithm 9: Implementation of TTM Inference

Input: a new document d_{new}, topic number K, α and β, $\vec{\varphi}$
Output: the topic distribution of the new document $\vec{\theta}_{new}$

Encoding the new documents d_{new};
Generate triterm set T_{new} from all triterms in d_{new};
Randomly assign a topic z to all triterms in T_{new};
for $iter = 1$ *to* N_{iter} **do**
    foreach *triterm* $t_i \in T$ **do**
        Compute and update $\theta_z = \frac{n_z + \alpha}{|T| + K\alpha}$;
        Compute $P(z|T) = \theta_z \cdot \varphi_{w_i|z} \cdot \varphi_{w_j|z} \cdot \varphi_{w_h|z}$;
        Re-assign topic z to triterm t_i;
        Update n_z;
    end
    Obtain the final $\vec{\theta}$;
end

Chapter 6

Evaluation

This chapter provides the experimental clustering results of word-embedding based clustering algorithm, traditional probabilistic based clustering algorithm and our model, and evaluates them by calculating the coherence. Moreover, the application of our model TTM on the COVID-19 dataset and the visualization result combined with analysis will also be displayed in this chapter.

6.1 Evaluation Method

To evaluate the performance of the unsupervised clustering model, the traditional methods for model evaluation such as cross-validation cannot work. Due to the particularity of the topic model, the obtained result can be displayed as the top words for each topic, which means the performance of the text clustering can be evaluated by human judgments, if most of the clustered topics (with the top words) have interpretable meaning, the performance of this model can be considered good. However, based on some experiments on training the models, most of the generated topics are hard to interpret, which makes evaluating the topics by human judgment difficult and because there is no unified standard for human to score the topics, it is also hard to compare the performance of different models.

Topic coherence measures have been proposed to determine whether the topic is good or not based on the interpretability of the top words [66, 67]. Roder et al. [68] proposed a framework allows the existing coherence measures and new measures that combined the basic components to compute the coherence, and the experiment showed the results have a positive correlation with human ratings.

There are some python libraries provide APIs to compute the coherence, and for convenience, we adopt an online service Palmetto, which can return the coherence value of the word set by simply putting the top words of the topic and choose a coherence measure.

We choose C_P and C_A to evaluate the topics generated from our model. C_P computes the coherence based on a sliding window, a one-preceding segmentation of the words and Fitelson's coherence [69]. It uses a sliding window to derive the word co-occurrence counts of the word set, and for each word, uses the confirmation measures of Fitelson's coherence to calculate the confirmation to its preceding word. And the coherence result is the mean value of the confirmation measure results. While C_A based on a context window, word pairs, normalized pointwise mutual information (NPMI) and the cosine similarity [69]. It uses a context window to retrieve the co-occurrence counts for the word set, which is used

to calculate the NPMI of each word to every other word and generate a vector for each word. Then compute the cosine similarity between all the word pairs and the coherence result is the mean value of those similarities.

6.2 Experimental Result

To evaluate and compare the models, we use the same test dataset as the input of the experiments. Since our model is expected to apply to short texts (Tweets), we tend to choose the short texts datasets as the test data. A SearchSnippets dataset, which contains 12295 documents and 5547 words from the results of web search transactions in eight different domains, is found to suit our experiments. And the number of cluster in the experiments is set to eight that same with the number of topics in the test dataset.

Table 6.1 shows the experimental result of K-means clustering algorithm. Few of the topics are interpretable, but generally the coherence score is low and there are topics obtained negative scores. It demonstrates that a more advanced model should be used to get higher coherence scores.

Topic ID	Top words	C_A	C_P
1	research edu gov information project science papers journal issues topics	0.101	0.169
2	wikipedia encyclopedia wiki system article computer film united culture science	0.235	0.029
3	news com sports information football articles health net reviews statistics	0.093	-0.383
4	information gov home online health business world web music index	0.146	0.17
5	edu university department science computer school theory course theoretical information	0.352	0.376
6	amazon com books music life theory political online computer democracy	0.173	-0.123
7	com sports online definition world information search games index web	0.159	0.036
8	movie movies film com reviews american database video news guide	0.177	0.065
		1.436	0.339

Table 6.1: Experimental Result of K-means

The following table 6.2 shows the experimental result of LDA. Due to the sparsity problem which discussed in section 2.2, the performance of LDA is not good.

Table 6.3 is the experimental result of BTM. The parameters α and β are both set to 0.5, and the iteration is set to 500 times to ensure the Gibbs sampling converged. Combined with LDA, both C_A and C_P scores of BTM are distinct higher than that of LDA, and as for human judgement, there are some topics having interpretable meaning such as topic 5 is about sports, topic 6 is about computer and topic 7 related to health.

Topic ID	Top words	C_A	C_P
1	games game sports play espn scores political com schedule news	0.207	0.103
2	football club bbc wikipedia imdb title cup event encyclopedia diagnosis	0.157	-0.008
3	party union governing diet print community beats elected teacher math	0.069	-0.221
4	tournament sport sports yahoo news directory coverage online com events	0.218	0.213
5	culture calendar school civil texas lessons liberal master phd instruction	0.152	0.124
6	sports physical nuclear rules weapons wins baltimore prize institution statistics	0.221	-0.408
7	republic england theoretical reform london san merchandise research chicago political	0.134	-0.027
8	tickets australian newspaper theatre britannica magazine germany minister conditions physics	0.167	-0.036
		1.325	-0.26

Table 6.2: Experimental Result of LDA

Topic ID	Top words	C_A	C_P
1	business com amazon market theory news trad information wikipedia stock	0.183	0.009
2	research edu science school information university journal computer department home	0.240	0.350
3	music movie com film movies news art video online fashion	0.176	-0.086
4	wikipedia political culture system encyclopedia wiki party government democracy information	0.262	0.249
5	news sports football com games soccer game world tennis match	0.203	0.252
6	computer software web internet memory intel programming com wikipedia device	0.202	0.233
7	health information cancer gov medical news disease healthy nutrition hiv	0.233	0.555
8	amazon theory physics edu theoretical books theorem philosophy mathematical com	0.270	-0.041
		1.593	1.521

Table 6.3: Experimental Result of BTM

The following table 6.4shows the experimental result of TTM, where the parameters α β and the iteration are all same as those in the experimental of BTM. It is found that both C_A and C_P coherence scores are higher than the traditional BTM. In addition to the topics that are also extracted by BTM such as sports (topic 5) and health(topic 7), TTM also explored more latent topics related to political (topic 6) and finance (topic 4). Moreover, the capability to distinguish the topics with semantically similarity becomes little better. For example, topic 1 and topic 8 are both related to university, but it can tell that topic 1 focus on the school while topic 8 more talks about some programs. However, the impact of high-frequency words on the clustering results is aggravated in TTM, so that some high-frequency words such as 'software' might appears on most of the topics.

Topic ID	Top words	C_A	C_P
1	software research computer edu science information web school internet university	0.277	0.435
2	com music movie film news amazon video movies online software	0.217	0.119
3	culture theory wikipedia software science edu amazon com encyclopedia journal	0.184	0.243
4	business market news stock com software trade finance car services	0.169	0.030
5	sports news football com games soccer game tennis world software	0.330	0.254
6	political wikipedia system software party democracy government encyclopedia gov war	0.166	0.225
7	health information cancer medical gov news nutrition disease healthy software	0.229	0.519
8	edu school university software research department graduate science program college	0.525	0.609
		2.097	2.434

Table 6.4: Experimental Result of TTM

6.3 Application and Visualization Result

There are two applications of TTM on COVID-19 dataset. One is directly apply the TTM algorithm on the COVID-19 related Twitter dataset to extract the latent topics people talk about on Twitter. The other is train the model with a rich dataset and obtain the model, then apply the model on the Twitter dataset to inference each Tweet belongs to which topic that generated from the training dataset.

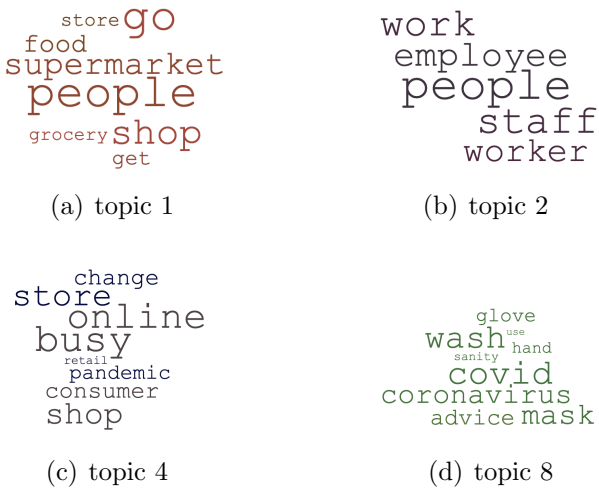
For the first application, we applied TTM on the COVID-19 related Twitter dataset collected from the early stage of the pandemic. Removing some high frequency words, such as covid and corona, the topics and their top words generated from TTM are shown as table 6.5.

The interpretability of the topics is not as expected, but from the generated results we can discover that at the early stage of the pandemic, people on Twitter mainly talk about

Topic ID	Top words
1	supermarket people store go grocery shop food get
2	worker work people employee staff store grocery
3	food demand supply panic buy stock people need
4	consumer online shop retail busy store change pandemic
5	price oil market consumer demand economy advice
6	protect advice scam help
7	toiletpaper toilet paper sanity mask
8	sanity hand mask use advice wash glove

Table 6.5: Hot Topics

food and supermarket (topic 1), work (topic 2) and some protecting measures (topic 8). Moreover, based on the top words of the above topics, we can guess that at that time in the pandemic, people may lack food and toilet paper but cannot go out to shop, and worry about their work, also begin to pay attention to protect themselves safety. The visualization examples (word cloud) of the topics are shown in figure 6.1.



(a) topic 1

(b) topic 2

(c) topic 4

(d) topic 8

Figure 6.1: Examples of Word Cloud

As for the second application, we trained TTM with a relatively various dataset and use the trained model to infer the input COVID-19 related dataset to find the topic distribution in a period of time. A google news dataset contains over ten thousand documents with over 150 topics, but after cursory observation, it is found that the topics generated by the google news dataset are not highly relevant to the content of our Twitter dataset. Therefore, we still used the SearchSnippets dataset but preprocessing by the same techniques of the Twitter dataset. 6.6 shows the trained result with the processed SearchSnippets dataset. Most of the generated topics are interpretable and it can be clearly to distinguish that the topics are related to science and subject (topic 1 & 8), computer (topic 2), polity (topic 4), music and movie (topic 3 & 5), sports (topic 6) and health (topic 7).

We randomly selected 500 Tweets samples each day to estimate the whole topics distribution in that day to discover the variation of topics in the second half of March (early stage of the outbreak). Since the lack of data between Mar. 27^{th} and Apr. 1^{st}, we divided

Topic ID	Top words
1	research edu science busy inform market program com journal
2	software computer web program system internet memory intel com network
3	music com car art engineer electric online buy home
4	polity culture wikipedia system party government encyclopedia inform wiki
5	movie com film video news fashion award art photo
6	sport news game football com soccer ticket team match
7	health inform cancer medical gov disease news research drug
8	amazon book theory wikipedia encyclopedia com physics mathematical theoretical

Table 6.6: Training Result

the data in those days into two parts (three days for one part), and selected 500 Tweets in each part. Additionally, with the purpose to find the topic variation during the pandemic, we use a line chart instead of a bar chart in the design, which is more clear to discover the evolution of topics. The line chart for variation of the topics in the second half of March is shown as figure 6.2, where the x-axis denotes the date, the y-axis denotes the number of Tweets assigned to each topic (in the 500 samples Tweets) and different colors denote different topics.

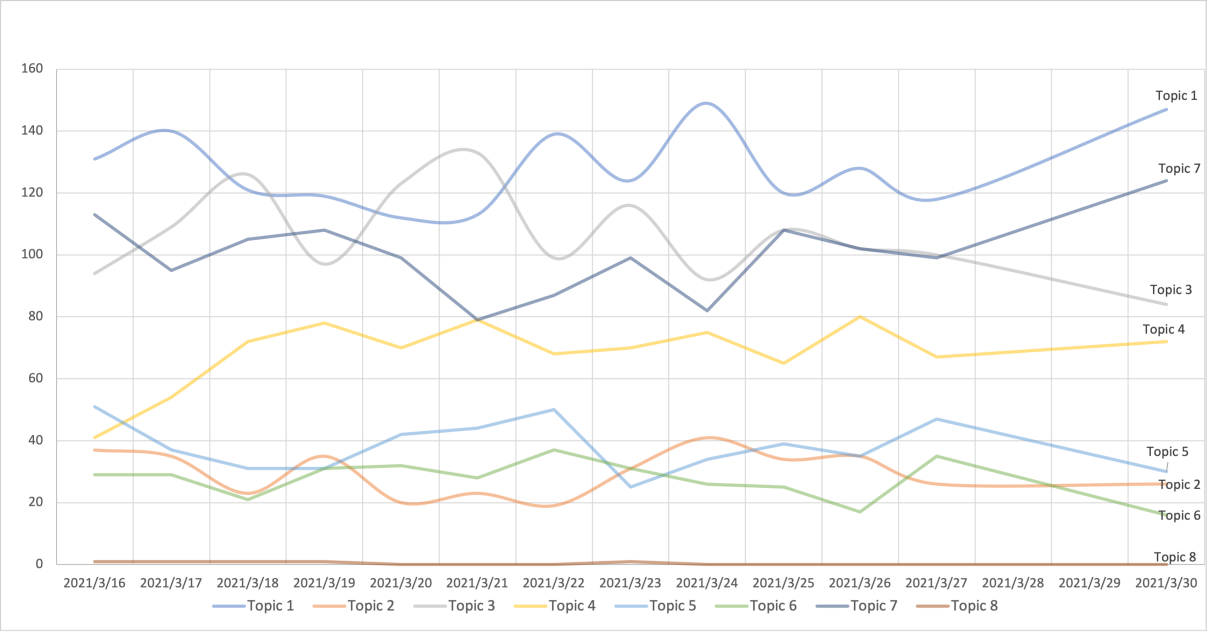


Figure 6.2: Variation of Topic Distribution

The above result shows that the distribution of most topics in the second half of March tends to stabilize. Topic 1 and topic 7 are mentioned frequently and have continuing upward trend. Topic 3, which seems to be related to online shopping and music, is also popular but might reduce in April. The topic about politics is rarely discussed at the beginning, but increases significantly afterwards. While other topics such as the topics related to movies and sports are not mentioned too much during the pandemic.

There is a certain degree of rationality for the above analysis of the change of topics in half month at the early stage of the outbreak. For example, it is clear that the discussions on the topics related to health and government have increased with the severity of the epidemic, which accords with the reality. And we can infer from the curve of topic 7 that after around one week from the outbreak, people start to pay more attention to the epidemic disease and their health. But because of the limited topics in the training dataset, there might be a lot of Tweets cannot be assigned to the right topic (since their topic may not be included in the training set), which will influence the result of the topic distribution. Moreover, the period of a half month is still quite short and cannot get more information for the analysis of public response. Therefore, the next stage of the project is to find a suitable training set with more topics and apply it on discovering the variation of topic distribution in a longer period of time during the pandemic to capture the public response to COVID-19.

Chapter 7

Summary

7.1 Project Management

In the original project plan, the project development should start with Twitter data collection. Creating or finding suitable datasets is an essential and significant task, and those data should be organized and preprocessed. Then the work should focus on researching the topic models and implementing the algorithms that suit the project's requirements. Model test and training should be proceeded synchronized. Whereafter, it comes to generate the results by the model and do the analyzing. The flowing Gannt Chart 7.1 shows the original timeline of the project.

A Write the project proposal

B Complete the ethics form

C Literature review

D Collect Twitter data

E Organize the dataset

F Data preprocessing

G Research on existing topic models

H Model design (for the existing models)

I Model training and test (for the existing models)

J Write the interim report

K Update the dataset

L Applying the topic model on the Twitter dataset

M Generate and organize the results about the prevalent topics in the pandemic

N Find the interrelation of the topics and summarize the topic revolution

O Technical improvement

P summarize the analyzing the results and conclude the research

Q Write the dissertation

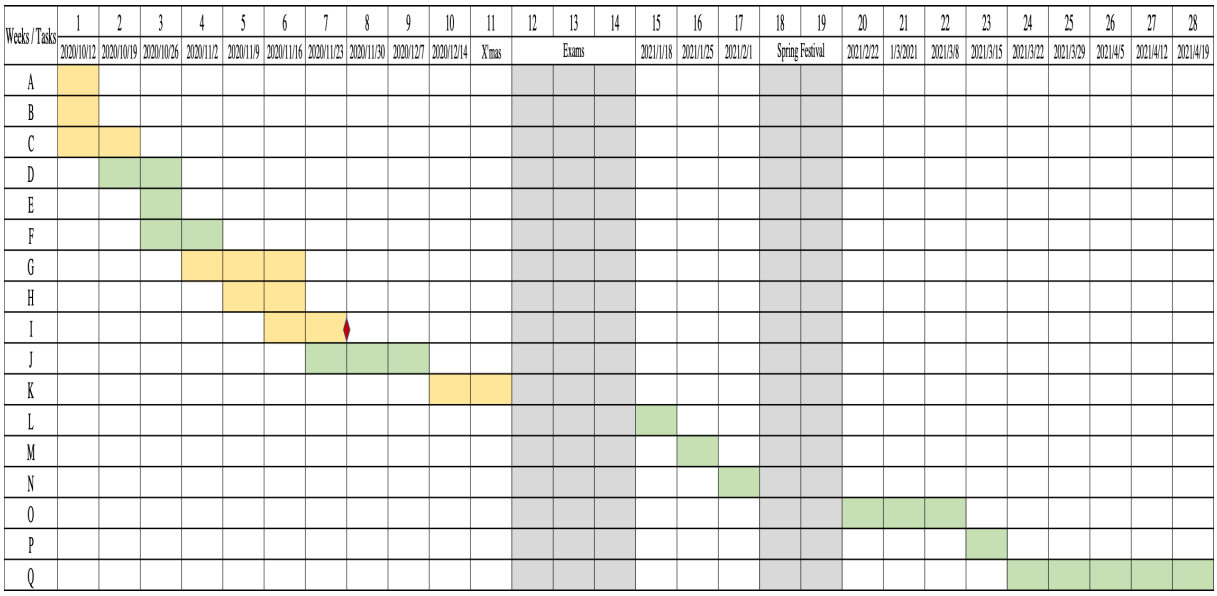


Figure 7.1: Original Work Plan

Weeks / Tasks	1 2020/10/12	2 2020/10/19	3 2020/10/26	4 2020/11/2	5 2020/11/9	6 2020/11/16	7 2020/11/23	8 2020/11/30	9 2020/12/7
A									
B									
C									
D									
E									
F									
G									
H									
I									
J									

(a) Timeline of Autumn Semester

Weeks / Tasks	10 2021/2/22	11 1/3/2021	12 2021/3/8	13 2021/3/15	14 2021/3/22	15 2021/3/29	16 2021/4/5	17 2021/4/12	18 2021/4/19
In-depth study of algorithms									
Technical improvement									
Applying on Twitter dataset									
Result analyzing									
Writing the dissertation									
Summarizing the project									

(b) Timeline of Spring Semester

Figure 7.2: Actual Timeline

Comparing with the original work-plan, the actual project progress is generally similar to it. The meetings with the supervisor were held every two weeks, which means that nearly every two weeks, the project moved to the next stage as schedule. The project progresses can be divided into two parts. The work of the first half of the project was mainly about basic research and data preparation. Though there were some unexpected reasons

such as the limitations of Twitter API, data collecting and preprocessing parts have been still finished in the Autumn semester. As for the Spring semester, the work turned to study about the principles of the topic models and propose the technical improvement, because after communicated with the supervisor, we agreed that the implementation of the algorithm improvement is the basis of the following analysis of result and implementing a more suitable algorithm can make the result better. Therefore, we modified the original work plan to focus on the algorithm instead of directly applying the existing models on our Twitter dataset and analyzing the generated topics, also adjusted the order of the tasks. Figure 7.2 shows the actual project timeline, where the first one is for the Autumn semester and the second one is for the Spring semester (except exam weeks, Christmas and the Spring festival).

7.2 Achievements & Contributions

In this project, we proposed a framework for uncovering COVID-19 events by analyzing social media data. The main contribution of this project can be divided into two parts, technical improvements on traditional topic models and applying the topic model on COVID-19 related dataset.

Firstly, we proposed methods to collect COVID-19 related data from Twitter for creating our own dataset, and implement a framework to preprocess the data. Then, based on the research and experiments about text clustering algorithms, we had an in-depth study about the principles and derivations of different models and compared the efficiency of them. With a depth understanding of the algorithms, we proposed our own model based on the traditional topic model to implement the topic extraction of COVID-19 related Tweets. Though there still are some limitations, such as the impact of high-frequency words, the experimental results showed the performance of our model is better than traditional BTM on the test dataset and some more latent topics can be discovered. Moreover, under this framework, we can discover the events during different periods of the COVID-19 pandemic through extracting the latent topics, which is a new application area for topic models. And the analysis results for the evolution of the topics in different periods can be applied in other research areas such as sociology and anthropology.

7.3 Reflection & Future Work

Overall, the actual project progress generally accords with the work plan and all deadlines were submitted on time. Throughout the first half part of the project, there indeed were some outcomes. The codes for data collection and data preprocessing were finished and the data set were preliminarily built up. But because of the clear objectives of the project, we started the data preparing part and ignored intensively reading related papers, which made the subsequent work for algorithm implementation progressed slowly, and made it harder to understand and improve the topic models in the Spring semester. Fortunately, we did not encounter any unexpected problems during the process of developing in the Spring semester. We reserved enough time to write the dissertation and summarize the whole project. Consequently, though the project is completed on time, it can be more

reasonable for the task management of some parts, for example, it should take more time to read the related papers before coding, which can make the following development more smoothly.

In the future, as mentioned in section 6.3, the first problem we need to focus on is to find more diverse datasets for training to improve the inference ability of our model and apply to a Twitter dataset covered the longer period of time during the pandemic. Subsequently, as mentioned in section 7.2, analyzing the generated topics in different periods of the pandemic can be used in sociology and anthropology research and the social media analysis can help government understand the public opinion to take the timely emergency response and take appropriate management measures during the pandemic [70]. Lee et al. [33] proposed a real-time disease surveillance system, which combines geographical analysis, temporal analysis and text analysis together to automatically track disease activities and has real-time visualized output. It can be considered as the future direction of our project. With the current achievement on text analysis, in the next stage, we hope to propose a dynamic model and achieve a more advanced surveillance system for epidemic disease.

References

[1] Rick Jin. *Mathematical Gossip of LDA*. 1.0 edition, 2013.

[2] Elıas Jónsson and Jake Stolee. An evaluation of topic modelling techniques for twitter. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers)*, pages 489–494, 2015.

[3] Xinhong Zhang, Daqing Jiang, Tasawar Hayat, and Bashir Ahmad. Dynamics of a stochastic sis model with double epidemic diseases driven by lévy jumps. *Physica A: Statistical Mechanics and its Applications*, 471:767–777, 2017.

[4] Evan L Ray and Nicholas G Reich. Prediction of infectious disease epidemics via weighted density ensembles. *PLoS computational biology*, 14(2):e1005910, 2018.

[5] Charles W Shmidt. Using social media to predict and track deseases outbreaks. *Environ Health Perspect*, 120(1):30–33, 2012.

[6] Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. Processing social media messages in mass emergency: A survey. *ACM Computing Surveys (CSUR)*, 47(4):1–38, 2015.

[7] Lauren E Charles-Smith, Tera L Reynolds, Mark A Cameron, Mike Conway, Eric HY Lau, Jennifer M Olsen, Julie A Pavlin, Mika Shigematsu, Laura C Streichert, Katie J Suda, et al. Using social media for actionable disease surveillance and outbreak management: a systematic literature review. *PloS one*, 10(10):e0139701, 2015.

[8] Luis Fernández-Luque and Teresa Bau. Health and social media: perfect storm of information. *Healthcare informatics research*, 21(2):67, 2015.

[9] Ireneus Kagashe, Zhijun Yan, and Imran Suheryani. Enhancing seasonal influenza surveillance: topic analysis of widely used medicinal drugs using twitter data. *Journal of medical Internet research*, 19(9):e315, 2017.

[10] Shahir Masri, Jianfeng Jia, Chen Li, Guofa Zhou, Ming-Chieh Lee, Guiyun Yan, and Jun Wu. Use of twitter data to improve zika virus surveillance in the united states during the 2016 epidemic. *BMC public health*, 19(1):1–14, 2019.

[11] David M Blei, John D Lafferty, et al. A correlated topic model of science. *The annals of applied statistics*, 1(1):17–35, 2007.

[12] Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. Modeling general and specific aspects of documents with a probabilistic topic model. *Advances in neural information processing systems*, 19:241–248, 2006.

[13] Umair Qazi, Muhammad Imran, and Ferda Ofli. Geocov19: a dataset of hundreds of millions of multilingual covid-19 tweets with location information. *SIGSPATIAL Special*, 12(1):6–15, 2020.

[14] Hamed Jelodar, Yongli Wang, Mahdi Rabbani, and Seyedvalyallah Ayobi. Natural language processing via lda topic model in recommendation systems. *arXiv preprint arXiv:1909.09551*, 2019.

[15] Xueqi Cheng, Xiaohui Yan, Yanyan Lan, and Jiafeng Guo. Btm: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):2928–2941, 2014.

[16] Kia Jahanbin, Vahid Rahmanian, et al. Using twitter and web news mining to predict covid-19 outbreak. *Asian Pacific Journal of Tropical Medicine*, 13(8):378, 2020.

[17] Sanda Martinčić-Ipšić, Edvin Močibob, and Matjaž Perc. Link prediction on twitter. *PloS one*, 12(7):e0181079, 2017.

[18] Eric M Clark, Jake Ryland Williams, Chris A Jones, Richard A Galbraith, Christopher M Danforth, and Peter Sheridan Dodds. Sifting robotic from organic text: a natural language approach for detecting automation on twitter. *Journal of computational science*, 16:1–7, 2016.

[19] William J Corvey, Sarah Vieweg, Travis Rood, and Martha Palmer. Twitter in mass emergency: What nlp can contribute. In *Proceedings of the NAACL HLT 2010 workshop on computational linguistics in a world of social media*, pages 23–24, 2010.

[20] Christian E Lopez, Malolan Vasu, and Caleb Gallemore. Understanding the perception of covid-19 policies by mining a multilanguage twitter dataset. *arXiv preprint arXiv:2003.10359*, 2020.

[21] Zhichao Fang and Rodrigo Costas. Tracking the twitter attention around the research efforts on the covid-19 pandemic. *arXiv preprint arXiv:2006.05783*, 2020.

[22] Wikipedia contributors. Topic model — Wikipedia, the free encyclopedia, 2021. [Online; accessed 10-April-2021].

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

[24] Hanna M Wallach, David M Mimno, and Andrew McCallum. Rethinking lda: Why priors matter. In *Advances in neural information processing systems*, pages 1973–1981, 2009.

[25] Antonela Tommasel and Daniela Godoy. Short-text feature construction and selection in social media data: a survey. *Artificial Intelligence Review*, 49(3):301–338, 2018.

[26] Jian Tang, Zhaoshi Meng, Xuanlong Nguyen, Qiaozhu Mei, and Ming Zhang. Understanding the limiting factors of topic modeling via posterior contraction analysis. In *International Conference on Machine Learning*, pages 190–198. PMLR, 2014.

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

[28] Amit Gruber, Yair Weiss, and Michal Rosen-Zvi. Hidden topic markov models. In *Artificial intelligence and statistics*, pages 163–170. PMLR, 2007.

[29] Weijiang Li, Yanming Feng, Dongjun Li, and Zhengtao Yu. Micro-blog topic detection method based on btm topic model and k-means clustering algorithm. *Automatic Control and Computer Sciences*, 50(4):271–277, 2016.

[30] Ximing Li, Ang Zhang, Changchun Li, Lantian Guo, Wenting Wang, and Jihong Ouyang. Relational biterm topic model: Short-text topic modeling using word embeddings. *The Computer Journal*, 62(3):359–372, 2019.

[31] Xingwei He, Hua Xu, Jia Li, Liu He, and Linlin Yu. Fastbtm: Reducing the sampling time for biterm topic model. *Knowledge-Based Systems*, 132:11–20, 2017.

[32] Nanning Zheng, Shaoyi Du, Jianji Wang, He Zhang, Wenting Cui, Zijian Kang, Tao Yang, Bin Lou, Yuting Chi, Hong Long, et al. Predicting covid-19 in china using hybrid ai model. *IEEE transactions on cybernetics*, 50(7):2891–2904, 2020.

[33] Kathy Lee, Ankit Agrawal, and Alok Choudhary. Real-time disease surveillance using twitter data: demonstration on flu and cancer. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1474–1477, 2013.

[34] Sergio Ramírez-Gallego, Bartosz Krawczyk, Salvador García, Michał Woźniak, and Francisco Herrera. A survey on data preprocessing for data stream mining: Current status and future directions. *Neurocomputing*, 239:39–57, 2017.

[35] Alper Kursat Uysal and Serkan Gunal. The impact of preprocessing on text classification. *Information Processing & Management*, 50(1):104–112, 2014.

[36] Zhao Jianqiang and Gui Xiaolin. Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access*, 5:2870–2879, 2017.

[37] Xingyi Song, Johann Petrak, Ye Jiang, Iknoor Singh, Diana Maynard, and Kalina Bontcheva. Classification aware neural topic model for covid-19 disinformation categorisation. *PloS one*, 16(2):e0247086, 2021.

[38] Yue Li, Pratheeksha Nair, Zhi Wen, Imane Chafi, Anya Okhmatovskaia, Guido Powell, Yannan Shen, and David Buckeridge. Global surveillance of covid-19 by mining news media using a multi-source dynamic embedded topic model. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 1–14, 2020.

[39] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*, 2017.

[40] L Douglas Baker and Andrew Kachites McCallum. Distributional clustering of words for text classification. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 96–103, 1998.

[41] Ron Bekkerman, Ran El-Yaniv, Naftali Tishby, and Yoad Winter. On feature distributional clustering for text categorization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 146–153, 2001.

[42] Apra Mishra and Santosh Vishwakarma. Analysis of tf-idf model and its variant for document retrieval. In *2015 international conference on computational intelligence and communication networks (cicn)*, pages 772–776. IEEE, 2015.

[43] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[44] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*, 2013.

[45] Wikipedia contributors. Word2vec — Wikipedia, the free encyclopedia, 2021. [Online; accessed 13-April-2021].

[46] Kenneth Ward Church. Word2vec. *Natural Language Engineering*, 23(1):155–162, 2017.

[47] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

[48] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[49] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[50] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, 78(11):15169–15211, 2019.

[51] Thomas Hofmann. Probabilistic latent semantic analysis. *arXiv preprint arXiv:1301.6705*, 2013.

[52] Gregor Heinrich. Parameter estimation for text analysis. Technical report, Technical report, 2005.

[53] Wikipedia contributors. Conjugate prior — Wikipedia, the free encyclopedia, 2021. [Online; accessed 15-April-2021].

[54] Jiayu Lin. On the dirichlet distribution. *Department of Mathematics and Statistics, Queens University*, 2016.

[55] Tomas Hrycej. Gibbs sampling in bayesian networks. *Artificial Intelligence*, 46(3):351–363, 1990.

[56] Wikipedia contributors. Gibbs sampling — Wikipedia, the free encyclopedia, 2020. [Online; accessed 19-April-2021].

[57] Alan E Gelfand. Gibbs sampling. *Journal of the American statistical Association*, 95(452):1300–1304, 2000.

[58] Dodo Zaenal Abidin, Siti Nurmaini, Reza Firsandaya Malik, Errissya Rasywir, Yovi Pratama, et al. A model of preprocessing for social media data extraction. In *2019 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, pages 67–72. IEEE, 2019.

[59] S Vijayarani, Ms J Ilamathi, Ms Nithya, et al. Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks*, 5(1):7–16, 2015.

[60] Gregory Grefenstette. Tokenization. In *Syntactic Wordclass Tagging*, pages 117–133. Springer, 1999.

[61] Usama M Fayyad Georges G Grinstein and Andreas Wierse. *Information visualization in data mining and knowledge discovery*. Morgan Kaufmann, 2002.

[62] Weiwei Cui, Yingcai Wu, Shixia Liu, Furu Wei, Michelle X Zhou, and Huamin Qu. Context preserving dynamic word cloud visualization. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pages 121–128. IEEE, 2010.

[63] Rate limits: Standard v1.1. https://developer.twitter.com/en/docs/twitter-api/v1/rate-limits. Accessed April 15,2021.

[64] Emily Chen, Kristina Lerman, and Emilio Ferrara. Tracking social media discourse about the covid-19 pandemic: Development of a public coronavirus twitter data set. *JMIR Public Health and Surveillance*, 6(2):e19273, 2020.

[65] Dimitrios Effrosynidis, Symeon Symeonidis, and Avi Arampatzis. A comparison of pre-processing techniques for twitter sentiment analysis. In Jaap Kamps, Giannis Tsakonas, Yannis Manolopoulos, Lazaros Iliadis, and Ioannis Karydis, editors, *Research and Advanced Technology for Digital Libraries*, pages 394–406, Cham, 2017. Springer International Publishing.

[66] David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 262–272, 2011.

[67] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pages 100–108, 2010.

[68] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408, 2015.

[69] Ben M'sik and Ben Msik Casablanca. Topic modeling coherence: A comparative study between lda and nmf models using covid'19 corpus. *International Journal*, 9(4), 2020.

[70] Xuehua Han, Juanle Wang, Min Zhang, and Xiaojie Wang. Using social media to mine and analyze public opinion related to covid-19 in china. *International Journal of Environmental Research and Public Health*, 17(8):2788, 2020.