Statistical Data Mining for Sina Weibo, a Chinese Micro-blog: Sentiment Modelling and Randomness Reduction for Topic Modelling

London School of Economics and Political Sciences

Wenqian Cheng

A thesis submitted to the Department of Statistics of the London School of Economics for the degree of Doctor of Philosophy,

London, March 2017

Declaration

I certify that the thesis I have presented for examination for the MPhil/PhD degree of the London School of Economics and Political Science is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it). The copyright of this thesis rests with the author. Quotation from it is permitted, provided that full acknowledgement is made. This thesis may not be reproduced without my prior written consent.

I warrant that this authorisation does not, to the best of my belief, infringe the rights of any third party.

I declare that my thesis consists of approximately 50,000 words.

Abstract

Before the arrival of modern information and communication technology, it was not easy to capture people's thoughts and sentiments; however, the development of statistical data mining techniques and the prevalence of mass social media provide opportunities to capture those trends. Among all types of social media, micro-blogs make use of the word limit of 140 characters to force users to get straight to the point, thus making the posts brief but content-rich resources for investigation. The data mining object of this thesis is Weibo, the most popular Chinese micro-blog.

In the first part of the thesis, we attempt to perform various exploratory data mining on Weibo. After the literature review of micro-blogs, the initial steps of data collection and data pre-processing are introduced. This is followed by analysis of the time of the posts, analysis between intensity of the post and share price, term frequency and cluster analysis.

Secondly, we conduct time series modelling on the sentiment of Weibo posts. Considering the properties of Weibo sentiment, we mainly adopt the framework of ARMA mean with GARCH type conditional variance to fit the patterns. Other distinct models are also considered for negative sentiment for its complexity. Model selection and validation are introduced to verify the fitted models.

Thirdly, Latent Dirichlet Allocation (LDA) is explained in depth as a way to discover topics from large sets of textual data. The major contribution is creating a Randomness Reduction Algorithm applied to post-process the output of topic

models, filtering out the insignificant topics and utilising topic distributions to find out the most persistent topics. At the end of this chapter, evidence of the effectiveness of the Randomness Reduction is presented from empirical studies. The topic classification and evolution is also unveiled.

Acknowledgements

First, I would like to express my sincere gratitude to Prof. Piotr Fryzlewicz for the continuous support of my PhD study, for his patience, motivation, and immense knowledge. I am grateful to him for his constructive criticism during my thesis writing and for his encouragement at all stages of my research. This work would not have been done without his continued guidance and generous support. My thanks also goes to Dr. Clifford Lam and Dr. Wicher Bergsma for their comments and suggestions on my work.

I would like to thank the Centre of Analysis of Time Series (CATS) and the Time Series Group of the Statistics Department at LSE for providing a perfect environment in which to pursue research. I am thankful to Ian Marshall (the Research Administrator of the Department of Statistics), and Lyn Grove and Jill Beattie (the Administrators at CATS) for their kind support.

There are many people whose suggestions, discussions and comments have greatly contributed to my PhD work. I very much appreciate Christopher Sciberras's efforts and kind assistance in proofreading this thesis and correcting my grammatical errors. I am grateful to Dr. Alan Pryor, my master project supervisor, to keep giving me valuable advice and inspiration in my research. I am also grateful to Ivan Sanchez, Dr. Sebastian Riedel and Dr. Nikos Aletras from the Computer Science Department at University College London and Dr. Georg Hahn from the Statistics Department at Imperial College for their comments and suggestions about my research. Hearty thanks go to my colleagues over the years for their stimulating

discussions and after-work chats. They include Rafal Baranowski, Anna Louise Schroeder, Na Huang, Majeed Simaan, Ewelina Sienkiewicz, Ali Habibnia, Cheng Qian, Yajing Zhu, and Hyeyoung Maeng.

I would also like to extend thanks to the following for their constant and unfailing support, and I could not have managed it without them: Chen Lu, Si Qiao, Ying Chen, Ruoxi Li, Anran Chen, Kun Wang, Jiang Shu, Nawal Mustafa, Michelle Warbis and Cynthia Endezoumou.

Finally, I am very grateful to my family who offered me both spiritual encouragement and financial support throughout all the stages of my research. Thanks for your unconditional love and belief in me.

Thank you all!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Social Media is a type of Internet-based application which allows the creation and exchange of user-generated content. It was built on the ideological and technological foundations of Web 2.0 (Oreilly, 2005), which allows users to update status and interact with each other, rather than just retrieve information. Due to the easy accessibility and multimedia nature of microblogging platforms, recently some Internet users tend to migrate from traditional communication applications, such as blogs or mailing lists, to microblogging services.

Authors of those posts broadcast their current status to the public or a selected circle of contacts, discuss current issues and share opinions on a variety of topics. Individuals can also embed shortened URLs, insert hash tags, and comment on others' posts. The 140 characters word limit of microblogs forces users to get straight to the point, which makes the posts brief but content-rich resources for investigation.

"Weibo" is the Chinese word for "microblog". Sina Weibo is a Chinese microblogging website which is used by well over 30% of Internet users in China, with a market penetration similar to Twitter in the United States (Rapoza, 2011). Forbidding the usage of Twitter in China resulted in the birth and popularity

of indigenous Weibo. It was launched by Sina Corporation on 14 August 2009, and attracted 503 million registered users by 2012 (Ong, 2013). Statistics showed there were 100 million daily users on Sina Weibo by the third quarter of 2015 (China-Resonance, 2011).

On average 600 million tweets were generated per day (500 million for twitter (TwitterInc., 2013), 100 million for Weibo (SinaCorp., 2013)), which create extensive resources for statistical data mining. The emotions and attitudes exhibited by these massive user-generated contents would be good indicators for people's thinking and future behaviours. Researchers, marketeers and political activists see these data as a good source for opinion mining, topic extraction and sentiment analysis for marketing or social studies. How to recognise the valuable parts of those enormous data and carry out statistical studies has become a significant research issue. The information from microblog posts is stored in unstructured formats (text) and not organised using any automated system, resulting in the complexity of data preprocessing and the difficulties of applying statistical methods.

In the literature, Weibo-related research mainly focuses on numerous qualitative analyses for popularity, social effect or verification mechanism, but only a few quantitative analyses could be found, which will be introduced in Section 2.1. My thesis will focus on the Sina Weibo, which is an emerging, attractive and still partially untapped research field. In addition, the research of Weibo sentiment time series modelling and topic modelling is relatively new and there remains much to be explored.

The introductory Chapter 2 provides initial results from the exploratory data

mining. It starts with a literature review for microblogs and describes how to generate data from Weibo for seven specific companies in Chart 2.1. Before capturing textual features, we conduct quantitative analysis on the pattern of the time of the posts for those specific companies and their relationship with share prices. In the second half of Chapter 2, after introducing Chinese word segmentation, we generated several results from initial text analysis. The term frequency chart can be a good show case for the most important terms and helps to build a basic understanding of the posting content. Clustering methods, such as k-means and k-mediods, are applied to group the posts.

In Chapter 3, we describe some fundamental methods for sentiment analysis and present time series for both positive and negative sentiments. A general framework for univariate time series model fitting is provided in detail, and this framework is then employed for our sentiment time series fitting. Results show that the proportional positive time series fits well using the general framework, while we need alternative approaches to fit the proportional negative time series. Model comparison and validation are presented for proportional negative sentiment after attempting multiple approaches for model fitting. Multivariate time series models are adopted as we intend to model and explain the interactions and comovements between positive and negative sentiment time series. Interesting results are obtained by exploiting the vector autoregressive (VAR) and BEKK models for the Box-Cox transformed time series data.

In Chapter 4, we first review the development and fundamentals of topic models. Latent Dirichlet Allocation (LDA), one of the most wide-spread topic models, was first presented and published as a graphical model for topic discovery by Blei et al.

(2003). The thorough generative process and model learning using Gibbs sampling are introduced in detail. After applying LDA on our microblog's data, it can be found that the generated topics contain some randomness. We intend to reduce the randomness and retain stable and well-explained topics for topic evolution analysis. Thus, the Randomness Reduction algorithm for filtering out the insignificant topics and utilising topic distributions to find out the most persistent topics is proposed. Case studies are presented to show the significance of the performance of the algorithm. The filtered topics are then classified and the evolution of these topics is detailed.

Finally, Chapter 5 concludes with a summary of contributions and includes a few ideas for future research.

Chapter 2

Exploratory Data Mining

2.1 Literature Review of Data Analysis of Microblogs

Before performing initial data analysis and further data mining on Weibo's data, a requisite is to conduct a literature review to explore previous researches on microblogs. Due to the wider spread and earlier establishment of Twitter, a more considerable body of research can be found on Twitter than on Weibo. The first step is to investigate the relevant literature on Twitter.

2.1.1 Review of Relevant Twitter Data Analysis

There have been several papers on exploring hotspots and examining the predictive power of Twitter. Using Twitter's data as predictors, researchers illustrated that it is possible to improve the prediction outcomes of disease outbreaks (Louis and Zorlu, 2012), including seasonal influenza (Achrekar et al., 2012). Singh et al. (2012) pointed out that tweets can be used to visualise immediate traffic conditions in London. Improving voting behavior prediction by adding Twitter information

(Chrzanowski and Levick, 2012), such as elections prediction (Gayo-Avello, 2012a; Chung and Mustafaraj, 2011), is another interesting research topic. However, some counter-views (Gayo-Avello, 2012b) have been put forward against using tweets as a predictive indicator for election. Other studies such as movie performance prediction (Asur and Huberman, 2010) and Oscar prediction (Thelwall et al., 2011) may also be of interest.

There are a few analyses about stock price prediction (Logunov, 2011; Mittal and Goel; Bollen et al., 2011). How the posts from microblogs relate to the fluctuation of stock price is an appealing topic for researchers, and there are many papers on this theme. Most stock prediction practice is based on a large amount of tweets and aims at predicting benchmark stock market indices, rather than focusing on specific companies. One of the most influential papers was by Bollen et al. (2011), investigating whether the measurement of collective mood states generated from large-scale Twitter posts are correlated to the value of the Dow Jones Industrial Average (DJIA), and the results indicate that the accuracy of DJIA predictions was improved significantly by including specific public mood dimensions. In addition, most articles with keywords "microblog" ("Twitter") and "stock price" also include "sentiment" in their title. This fact implies that sentiment time series modelling is a crucial element for predicting future stock price using microblog posts.

In the literature, there are two possible ways for sentiment analysis: one is lexicon-based approaches, and the other is applying various machine learning classification techniques. Most widely-used English lexicons include Profile of Mood States (POMS) (McNair et al., 1981) and WordNet (Fellbaum, 1998). OpinionFinder (2016) is a system that performs subjectivity analysis and sentiment

identifications for English texts, and it was applied by Bollen et al. (2011) and Ramon (2012) for sentiment analysis in their research. We adopted several Chinese lexicons in our research and they will be introduced in Chapter 3. Machine learning algorithms, such as naive Bayes classifier, Support Vector Machine (SVM) and Neutral Networks (NN), were employed as supervised learning methods for sentiment classification in many studies (Xie et al., 2012; Ramon, 2012; Ding et al., 2011; Tian and Zheng, 2012). Supervised learning algorithms examine the training data and develop an inferred function, which can be used for mapping new data. The relevance and quality of training data is crucial when high accuracy is required. However, due to the lack of relevant training data and the difficulties of manually tagging, we will narrow down our sentiment classification to lexicon-based approaches in this thesis.

Among those studies making use of Twitter's data, Ding et al. (2011) presented a stock market prediction based on the combination of share price time series and market sentiment. They obtained a well-structured tweet dataset from a private company, Infochimps, and employed Python's Natural Language Toolkit for the sentiment analysis. The training set for sentiment containing 295 labeled tweets was prepared manually, and different supervised learning methods, e.g. support vector machine and logistic regression, were employed and compared. Results showed that SVM appeared to be the most accurate learning model for predicting market movement, and 5-days of prior data achieved the highest accuracy rate. Ramon (2012) also illustrated several approaches to extract information from Twitter to improve time series prediction, including stock price movement and box office performance. Techniques such as text categorization, sentiment analysis and

regression models were applied for selecting and analysing the content of tweets. To assess the model adequacy, non-linearity and causality tests on the time series were conducted. Results showed that adding the additional time series from Twitter results in an increase in prediction accuracy of 5 percent or more. These studies are closely connected with the research detecting the potential relationship between Weibo's post amount and share price, which can be a starting point for our Weibo data analysis.

2.1.2 Review of Weibo Data Analysis

According to iResearch's report (China-Resonance, 2011), Sina Weibo had 56.5% of China's microblogging market based on active users, and 86.6% based on browsing time over competitors such as Tencent Weibo and Baidu's services. It successfully went public on Nasdaq under the symbol "WB" in 2014.

The differences between Twitter and Weibo are not negligible. Compared to Twitter, there are plenty of distinctive features of Weibo. First, a post in Weibo can contain a graph or a video directly (not just a hyperlink). Second, users can comment on any posts without reposting them. That means commenting in Weibo is more casual than in Twitter, allowing users an option to put these replies/comments on their own timelines (personal page) or not. Third, the innovative verification mechanism attracts more users because there are numerous verified celebrities, professionals, and prominent business people on Weibo. In addition, media censorship control exists for Weibo, and the system is supervised by the Central Propaganda Department of the Chinese government.

There is little literature in English on Weibo, therefore, we searched for Weibo literature using an official Chinese Academic Platform named China Knowledge Resource Integrated Database (CNKI, 2016). Most papers about Weibo are qualitative analyses, for example about Weibo's propagation characteristics, posters' motive and behaviour, and the relationship between Weibo and marketing. Only a few studies have included quantitative analyses on Weibo and these studies will be introduced in the following paragraphs. Reviews will include three main aspects: stock market prediction, hotspot detection, and topic or social affair trend forecasting.

A group of researchers at Shanghai University (Zhou et al., 2013) found that investors' emotions could be detected from the heat of some keywords on microblogs and could be used to predict stock market fluctuations, which is similar to Twitter analysis for stock price prediction. They defined six keywords which could be translated as "Bull Market", "Positive News", "Stock Index rise", "Bear Market", "Negative News", and "Stock Index drop". The heat of the keywords was defined as the daily number of posts containing those words. The input data were the sequences of the heat and the output data were the fluctuation labels (+1 and -1) of the closing price. A significant Granger causality relationship between "rise" or "drop" and closing prices was found. Results showed that the heat of keywords' indicated the trend of some events indirectly, and the accuracy was about 59% for "stock index rise" and 78% for "stock index drop". By setting a lag of one week's time, the change of heat of these two words could correctly predict the change of Shanghai Composite index in most cases.

In another paper, Sheng (2012) illustrated a method to find, track and analyse

possible hotspots of a certain subject using Weibo data. A case study to find the potential hotspots of the specific field of "data mining" was conducted and compared with other traditional hotspots detecting methods. Sheng (2012) performed word frequency counts, co-word analysis, and social media analysis to find out the hot keywords and academic leaders for this subject, and track them continuously. Comparing those automatically detected hotspot results from Weibo with the topics on Web of Science for "data mining" subject, they found around 30 percent of specific topics were identical. Results showed that making use of Weibo data would improve the hotspot detection further as it could include several more latest hotspots than traditional methods.

Zheng et al. (2012) have proposed an approach for news topics detection using Weibo data. By finding the emerging keywords from a large amount of posts and then clustering them, news topics could be recognised and recorded. To identify the keywords, researchers introduced the compound weight to combine the word frequency and the growth trend. A measurement of the likelihood of a word to be a news keyword was taken, and a contextual relevance model was used to support incremental clustering and construct the topic. The results proved the effectiveness of the approach to detect news topics out of massive messages (3 million posts for 10 days).

Tian (2012) conducted another analysis for event trend prediction on the Weibo platform. Tian (2012) employed the Moving Average Convergence and Divergence (MACD) algorithm and the Latent Dirichlet Allocation (LDA) algorithm to detect the contents of breaking events and extend the related text of the known events. The Latent Dirichlet Allocation algorithm for topic modelling will be employed

and illustrated fully later in Chapter 4. By calculating aggregation and separation between the short-period and long-period moving average line, breaking events can be recognised. The LDA algorithm is applied to figure out the event-related "word bag" and corresponding weight for the event. The data of the first seven days were used as the training materials and the eighth day's posts were used to check the prediction. The results of the prediction showed some significance, but the limitation was that only a single case study with small data size and very short time span was conducted.

After reviewing the relevant literature for both Twitter and Weibo, we start our research with some exploratory data mining, such as the time of the posts, the relationship between post amount and share price, term frequency and cluster analysis. Time series modelling on sentiment and topic modelling for hotspots detection will be carried out and clarified in the following chapters.

2.2 Data Collection

Most social media platforms provide Application Programming Interface (API), the programmatic access to read and write data, for public developers. The initial choice to capture Weibo data is through its official API. However, after Sina Weibo updated its authority mechanism from OAuth1.0 to OAuth2.0 on the 15th of October 2012, it closed the search interface by which one can download posts with specific keywords. This most effective and frequently-used method adopted by previous Weibo data mining no longer exists. It means that it is impossible to get a large amount of Weibo posts containing specific keywords via Sina API,

but it is still possible to download posts by individual users or geolocation. The technical details of obtaining data via API can be found in Appendix A.1.

Unlike Web Crawling, specified in Appendix A.2, which downloads all the information from each page, Web Parsing is generally targeted at capturing specific information and transforming unstructured data into structured data that can be stored and analysed in a database. Extensible Markup Language (XML) is adopted for Web Parsing as a markup language that defines a set of rules for encoding data in a format that is both human-readable and machine-readable. The advantage of this method is that there is no need to consider API limits. The disadvantage is that it is still an unofficial method to capture data, so the web page format or the download limit may change. This method had been adopted and kept stable for more than a half year until Sina changed the version of the web interface December in 2013 and included the Captcha to avoid data acquisition from external applications.

Web Page Parsing can be achieved using Rweibo (Li and Chen, 2012), an R package developed as an unofficial Software Development Kit for Weibo. Software Development Kit (SDK) is a wrapper around APIs that makes the downloading of data and designing of applications easy for developers. Rweibo provides a search interface to parse Weibo posts containing specific keywords. The limitation is that it can only capture approximately 800 posts (40 pages with 20 posts each page) per search, and it is suggested by the R package author that limit the search to every half hour and 40 pages for each search to guarantee the stability. In this way, the hotness of the keyword might determine the completeness of the data. If a keyword is very popular, there will be a large number of posts within a certain

period, and if we intend to capture all of them, the time interval of running our code needs to be relatively short. Thus, due to the limit of the search frequency, we may only choose moderately hot keywords in order to capture continuous data. For instance, if we chose "Baidu" (the most popular Chinese search engine) as our keyword, the number of posts containing "Baidu" will far exceed the limit of 1600 per hour. Thus, we cannot capture a complete dataset for a very popular keyword. In contrast, if a brand is not popular at all, only a few posts can be found within an hour. It will be difficult to conduct text mining or time series analysis.

The latest version of Rweibo enables a function to include automatically incremental posts after each download, and thus we can capture the posts completely by continuously running a loop. We chose seven companies as our specific keywords and performed the data collection by ourselves. The seven companies, Suning, Biguiyuan, Maotai, Guomei, Vanke, Biyadi and Donghang meet requirements: well-known Chinese company, listed on China's stock market, and with moderate popularity to guarantee a certain number of posts every day, but also to avoid hitting the maximum limit. More descriptions about the companies can be found in Table 2.1.

Table 2.1: Descriptions about the companies.

Name	Introduction	Stock Symbol	Website Link	Market Value
Vanke	One of largest residential real estate developers	000002.SZ	www.vanke.com	250.88b CNY
Biguiyuan	One of largest residential real estate developers	2007.HK	www.countrygarden.com.cn	92.20b HKD
Suning	One of the largest privately owned electrical appliance retailers	002024.SZ	www.suning.com	99.62b CNY
Guomei	One of the largest privately owned electrical appliance retailers	0493.HK	www.gome.com.cn	21.75b HKD
Maotai	A state-owned enterprise in China, specializing in the production and sales of Maotai liquor	600519.SS	www.moutaichina.com	38.27b CNY
Biyadi	The sixth largest Chinese automobile manufacturer	1211.HK	www.bydauto.com.cn	145.96b HKD
Donghang	A major Chinese airline operating international, domestic and regional routes	600115.SS	www.ceair.com	63.17b CNY

Our data set covers approximately six months and the post amount of each company are listed in Chart 2.2. In summary, the data collection for four companies, Biyadi, Guomei, Donghang and Vanke, started on 3rd May 2013, and for three other companies, Biguiyuan, Maotai and Suning, started on 8th May 2013. Our data collection ended on 9th December 2013 when Sina added Captcha to its login mechanism. More technical details can be found in Appendix A.3.

Table 2.2: Post amount for each company.

Company	Length	Post Amount
Vanke	221 days	247373
Biguiyuan	215 days	94809
Suning	215 days	732684
Guomei	221 days	327298
Maotai	215 days	183539
Biyadi	221 days	134530
Donghang	221 days	38963

Due to interrupted internet connections or Windows blue screens (system crashes), we have several small gaps in our data set. These data are missing at random, as the probabilities that they are missing are independent of the variable itself and due to external factors. Single imputation is employed here to deal with the missing data. It provides the dataset with a specific value in place of the missing data with straightforward computation (Rubin, 1976). While there is more than one type of single imputation, in general the process involves analysing the other data points, looking for or calculating the most likely value, and placing it in the dataset. When only few data are missing, single imputation provides a useful enough tool, as the variance of the dataset is unlikely to be altered significantly by single imputation. But when one is dealing with a considerable amount of missing data, multiple imputation might be a better choice (Rubin, 1978), because single

imputation treats the imputed data points as an equal to the original ones, which may cause misleading results (Rubin, 1988). There are on average three gaps in our dataset for each company and the gaps are mostly shorter than one day, so we employed simple imputation: the average of the day before and the day after is calculated as the modified daily post amount. For instance, in Vanke's data, the post amount on 26th June was only 56, which is unusual, and the average of the 25th and 27th's amount $\frac{1}{2}(1079 + 1526) = 1303$ is used instead. Other methods such as exponential smoothing are employed for missing data and extreme values in Chapter 3 for further sentiment time series modelling.

Apart from some analyses conducted on all the companies, the majority of the analyses in this thesis are mainly based on Vanke's data. This is due to the fact that, compare to other companies, it has higher quality and a moderate quantity of data. Suning and Guomei include a large number of promoted posts and thus have many bursts and a very large quantity of data. The other companies are less popular than Vanke and have a smaller quantity of data.

2.3 Analysis of the Time of the Posts

In this section, we will investigate the posting patterns by analysing the time of the posts. A series of charts and graphs are created to show the different characteristics of posting time. Most of the analyses in this section are based on the data collected for company Vanke from the start of May till the start of December. The results for the other companies can be found in Appendix A.4.

Daily post amount, coloured by time, from 3rd May to 9th Dec 2013 can be seen in

Figure 2.3.1 for Vanke and Figure A.4.1 for all the other companies. It seems that the daily number of posts for Vanke during these 7 months are relatively stable at around 1300 posts per day with a slight downward trend, except for three extreme peaks and two gaps of missing data.

By observing the posts over the periods of peaks, we can find possible causes of those extreme peaks. Two peaks around 4th July and 25th November seem to result from breaking news of Vanke. The former appears to be from a news story announcing Vanke ranked as the top 1 in sales value in the "Top 50 ranking in China's real estate business sales in the first half of 2013". The latter appears to be due to a news story stating that Vanke evaded land value added tax amounting to 3.8 trillion Yuan. Many Weibo users forwarded and commented on these news, and there were many discussions and follow-ups. The reason for the peak around 29th July is hard to determine. It might result from many promotions and advertisements held by Vanke's official account during that week. It involved users by drawing prizes from all the forwarders, i.e. the official account will make posts for promotion purpose, and many followers will forward the posts to take the opportunities for random lucky draws, which boosted the post amount and attracted many followers.

Figure 2.3.2 shows the hourly post amount of Vanke. Unsurprisingly, people usually post during the period from 8am to 11pm. The morning seems to be the favourite posting time period for all the companies, and the first posting peak appears around 10am. However, the second peak appears in the afternoon for Vanke, which is similar to Biyadi and Biguiyuan, but the other three companies seem to have a second peak around 9pm in the evening (see Figure A.4.2). Suning has three

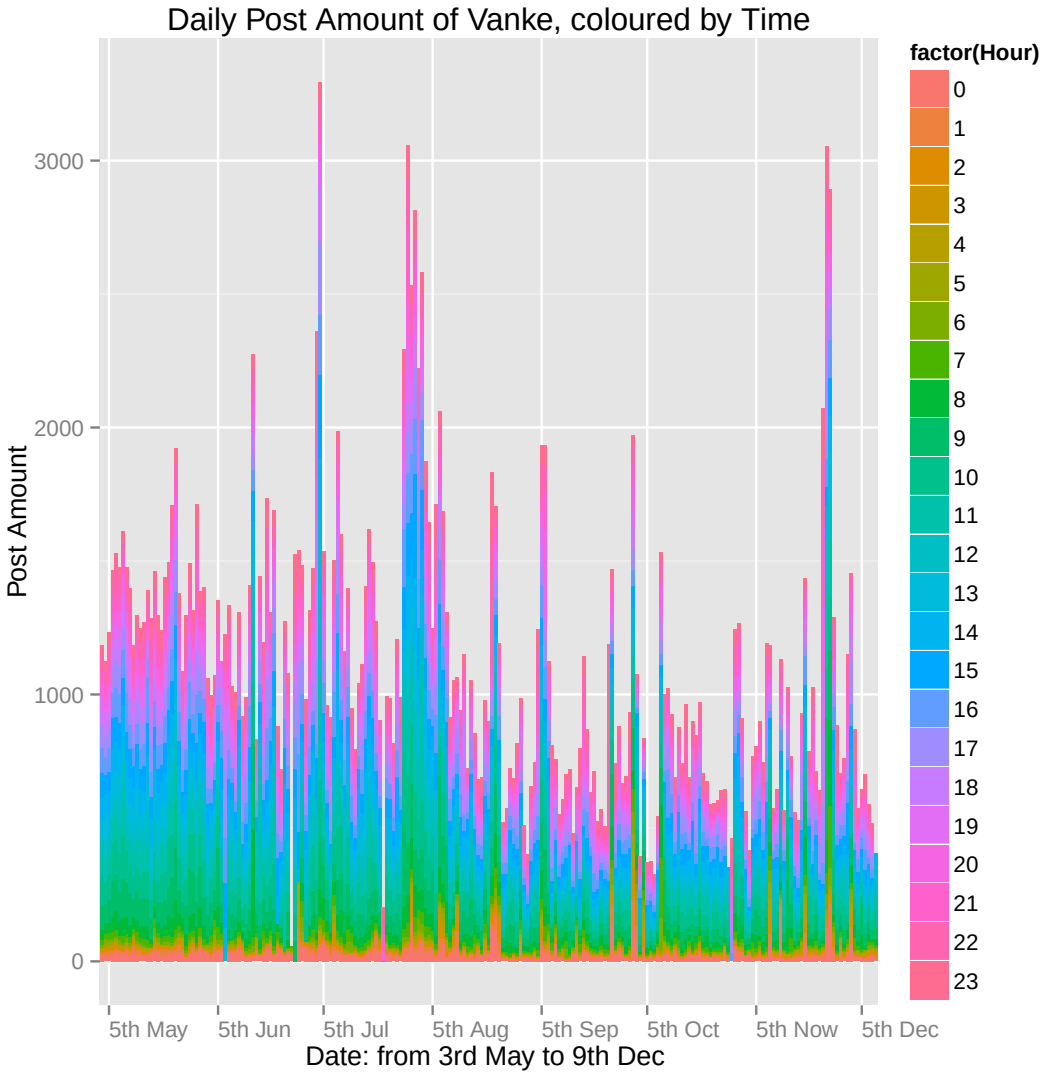

Figure 2.3.1: Daily post amount of Vanke. Different colors represent different posting time, i.e. 6am to 12 noon is in different shades of green, 12 noon to 7pm is in different shades of blue, 7pm to 12 midnight is in different shades of pink, and 12 midnight to 6am is in different shades of orange. The bars are the chronological daily post amounts from 3rd May to 9th December 2013.

peaks: morning around 10am, afternoon around 3pm, and evening around 10pm. After observing the original posts, the afternoon peak for Suning can be explained as the promotions by Suning were mostly announced on its official account in the afternoon and they initiated many reposts.

The evening peak could indicate that Weibo users talked about Donghang, Maotai, Suning and Guomei more often at home in the evening as they are brands more related to daily life. It may be because airlines (Donghang), liquor (Maotai), and electrical appliances (Suning and Guomei) are more likely to be topics of interest after work than real estate brands (Vanke, Biguiyuan) and automobile brands (Biyadi). These findings are supported by research which focuses on using tweets as an electronic word of mouth (Jansen et al., 2009). Jansen selected brands and categories under the assumption that these brands would be most likely mentioned in and affected by microblogging and after exploring several lists including American Customer Satisfaction Index, Business Weeks Top Brand 100, etc. The categories were chosen to be closely related with items in daily life, which included transportation, food and consumer electronics. Jansen's findings provide support to the empirical findings described above: Donghang (transportation), Maotai (food), Suning and Guomei (consumer electronics) are in Jansen's chosen categories and closely linked to daily life.

From the hourly time series plot in Figure 2.3.3, we can see that the post amount of Vanke fluctuates on an hourly basis averaging around 60 per hour. There are several extremely high values around 400 posts per hour, and they can be a good indicator for the occurrence of sudden events, e.g. major news or big promotions. The hourly posts amount varies between different companies, and the figures of the

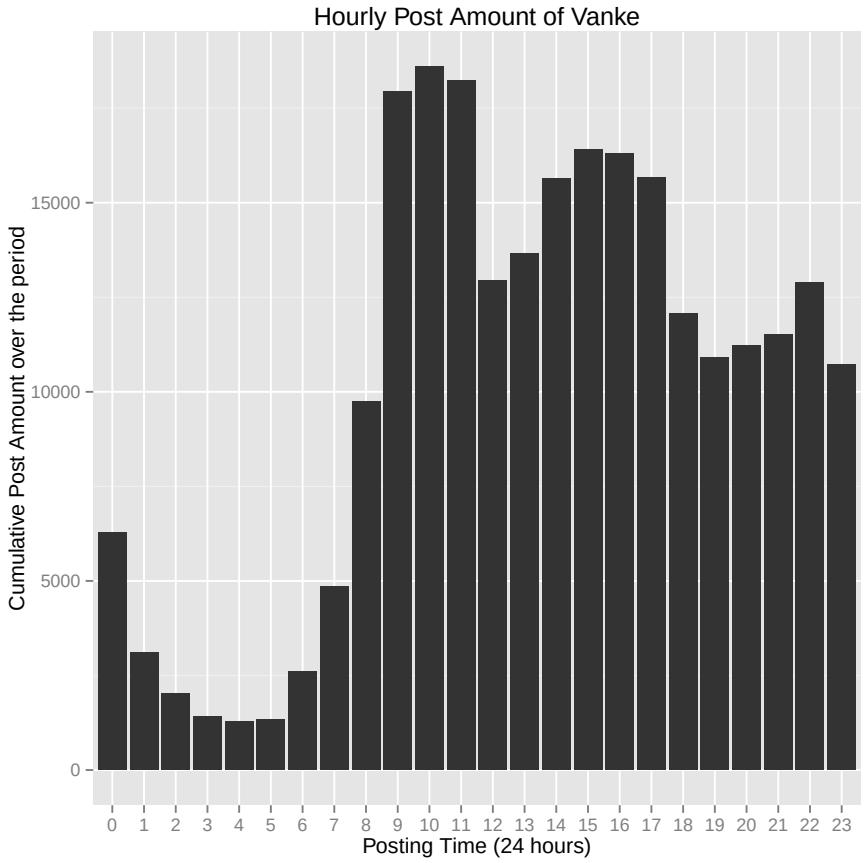

Figure 2.3.2: Hourly post amount of Vanke.

other six companies can be found in Figure 2.3.4. They have diverse patterns with different peaks, which can be observed further separately for sudden events. One feature of both Guomei and Suning is that they have many relatively long-lasting bursts rather than the momentary peaks for other companies. After observing the original posts, we can see that the bursts are mainly due to big promotions from their official account.

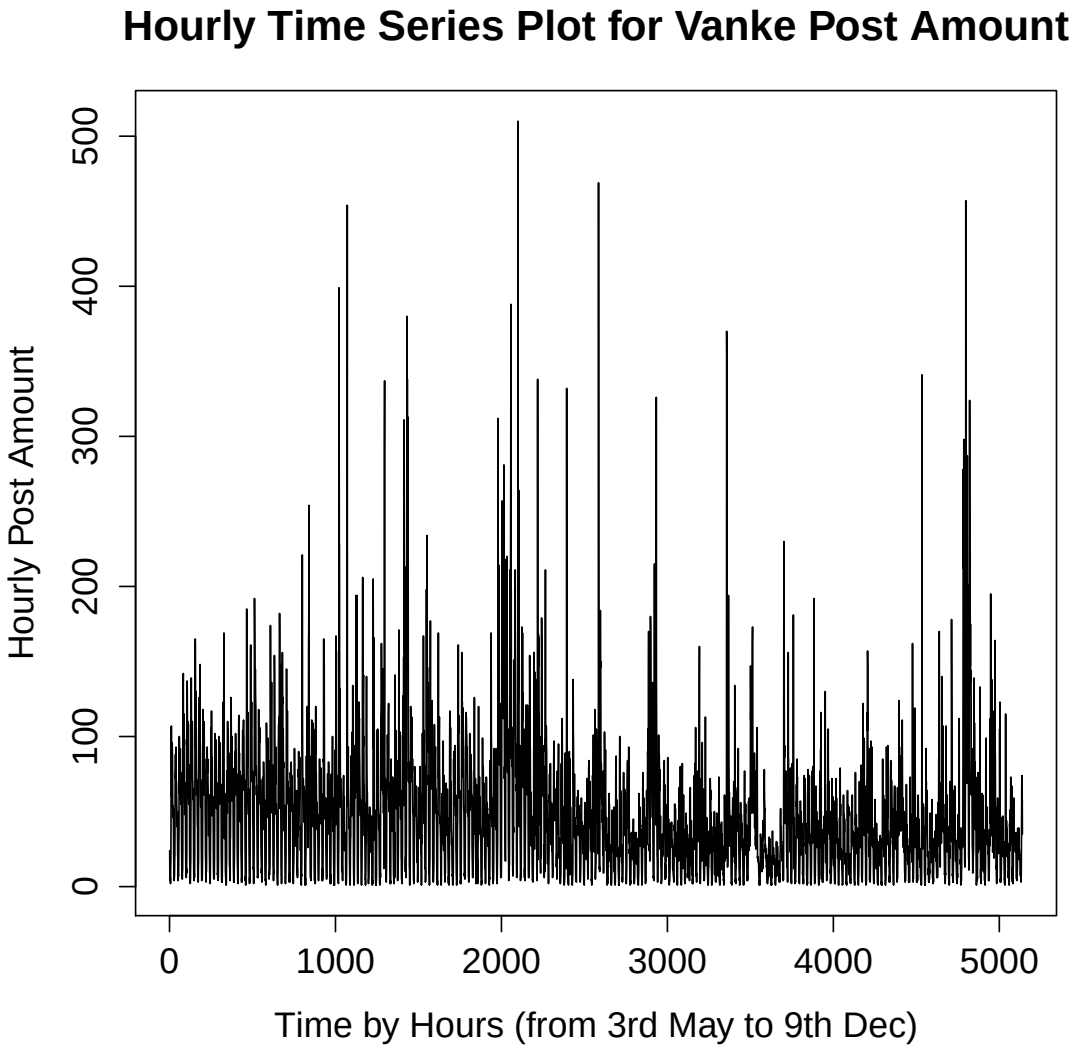

Figure 2.3.3: Hourly time series plot for Vanke post amount from 3rd May to 9th December 2013.

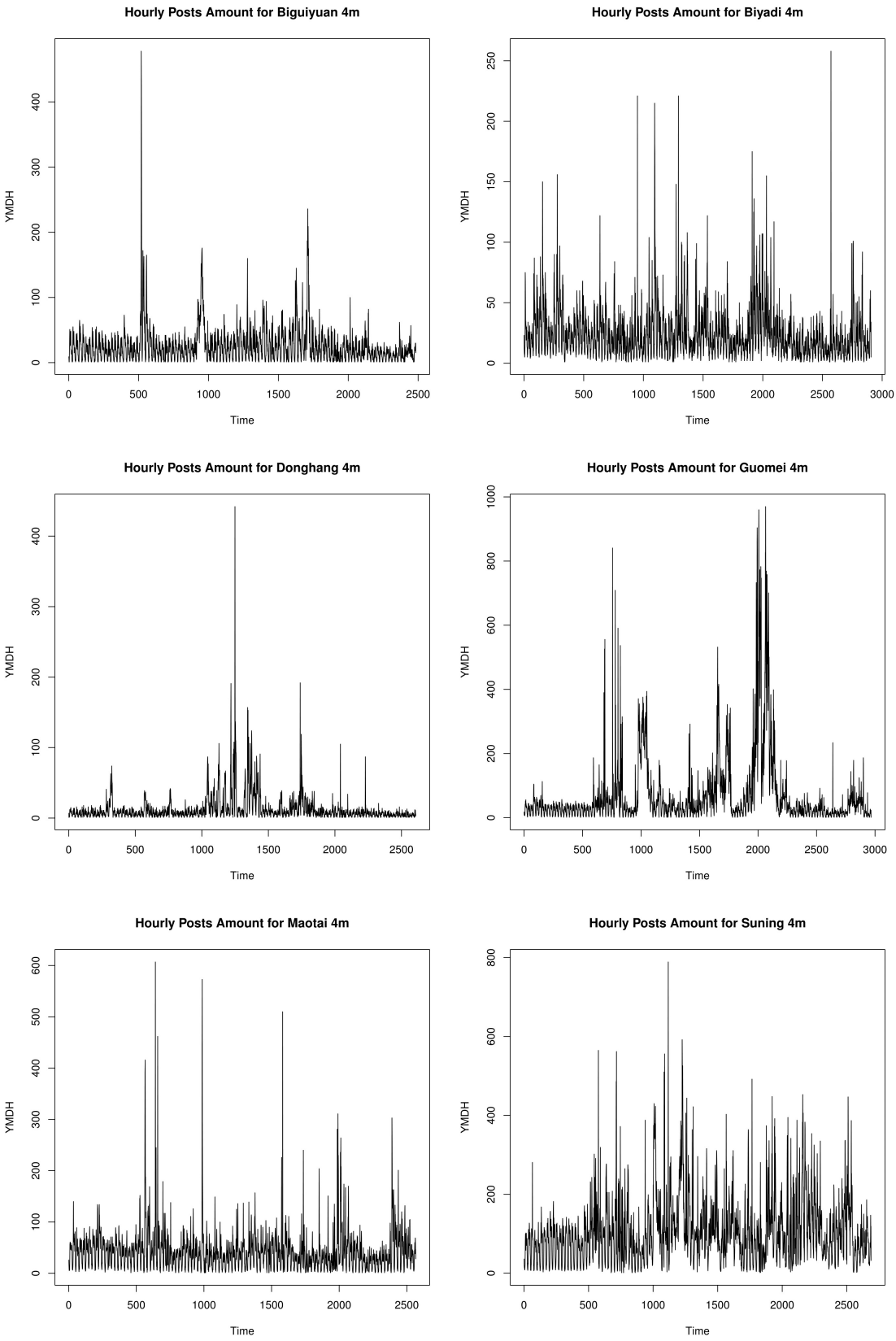

Figure 2.3.4: Hourly time series plot for post amount (multiple companies) from 3rd May to 9th December 2013.

2.4 Intensity of Posts vs Share Price

After analysing only the time of the posts, we are interested in the relationship between the intensity of posts of a company and this company's share price. In this section, we will focus on Vanke to analyse the potential relationship. The historical daily adjusted closing prices (adjusted for dividends and splits) of Vanke (Symbol: 000002.SZ) between 3rd May and 9th December are downloaded from the Yahoo!Finance (2016) website and recorded as original share prices.

We denote the post amount at day t as A_t and the original share price at day t as P_t. Initially, a line chart combining A_t and P_t is drawn in Figure 2.4.1. There seem to be some similar patterns between share price and post amount at the beginning of the period. A dot plot for A_t and P_t for Vanke is produced in Figure 2.4.2, and a weak correlation between the post amount and share price according to the plot can be seen. We conduct several tests in this section to further investigate the relationship.

One of the difficulties of processing the time series is dealing with the weekend gaps in share price. First, we create two modified time series and make the denotations:

P_t: Original Share Price at day t

P'_t: Modified Share Price at day t (create Saturday's share price and Sunday's as Friday's)

A_t: Original Post Amount at day t

A'_t: Modified Post Amount at day t (weekend data deleted)

In this way, we could have two groups of time series:

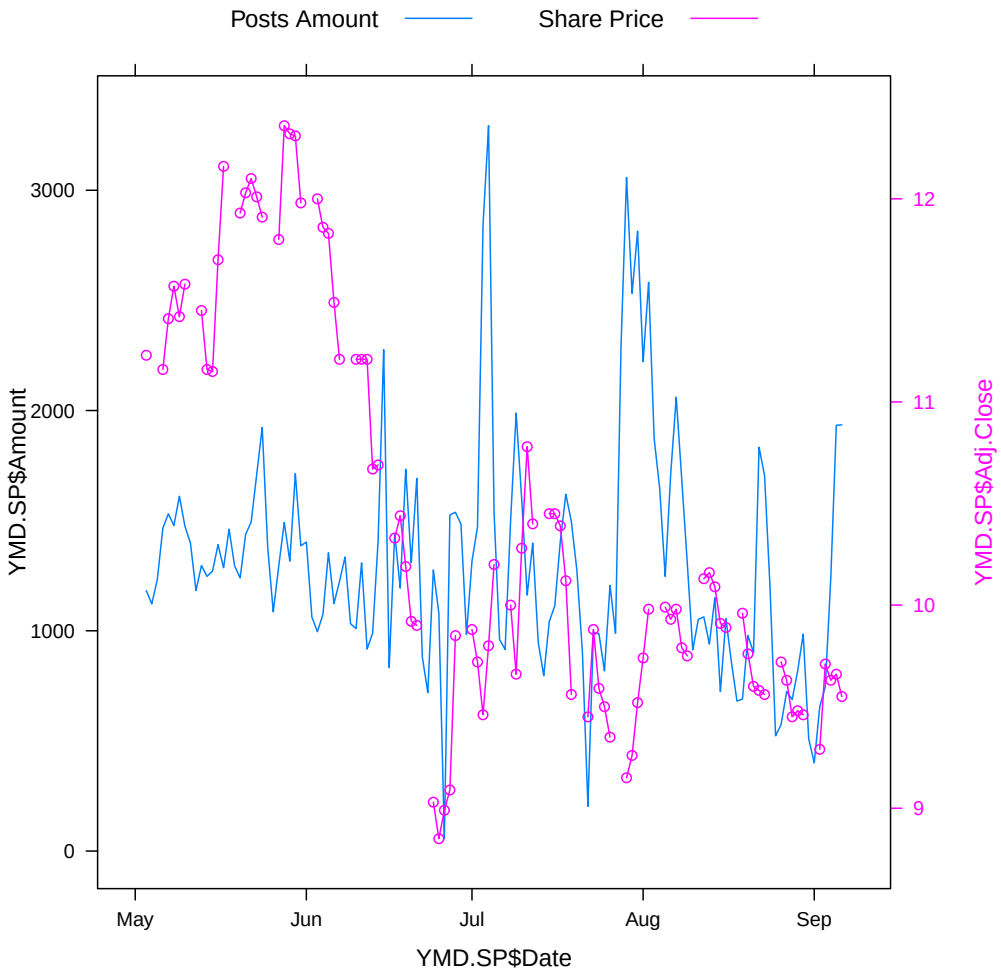

Figure 2.4.1: Post amount vs share price for Vanke. The dotted pink line represents the share price and the blue line represents the post amount. The share price gaps are due to the closure of the stock market during weekends.

Group A (without weekends): P_t vs A'_t

Group B (with weekends): P'_t vs A_t

To avoid the potential spurious regression, we examine the stationarities of share price and post amount time series respectively before conducting regression analyses. The Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test (Kwiatkowski et al.,

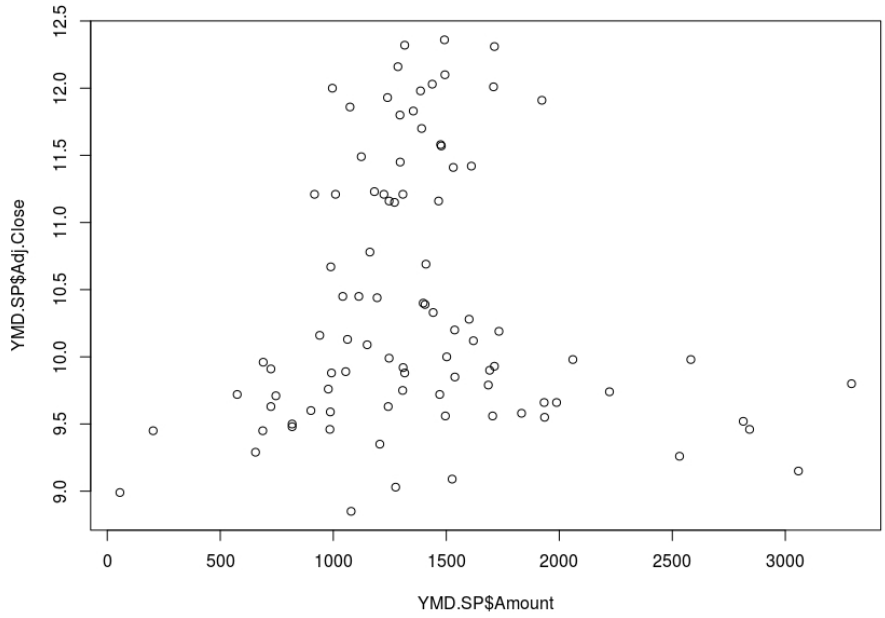

Figure 2.4.2: Dot plot of post amount (x-axis) and share price (y-axis) for Vanke.

1992), the Augmented Dickey-Fuller (ADF) test (Dickey and Fuller, 1979) and the Phillips-Perron (PP) test (Phillips and Perron, 1988) are applied to accomplish this task and validate each other. The KPSS test is a stationarity test with a null hypothesis that an observable time series is stationary around a deterministic trend. It can be used for both level stationarity and trend stationarity. If the result of the p-value is larger than 0.05, it means that H_0 can not be rejected, thus indicating stationarity. However, in the other two tests, the ADF test and the PP test, p-values smaller than 0.05 indicate stationarity.

As the time series for share price is non-stationary and we can assume today's share price is based on yesterday's, it is conventional to take log return transformations to capture the time-varying changes and enhance the stationarity. When returns are relatively small (common for trades with short holding durations, e.g. daily),

the approximation $\log(1+r) \approx r$ ensures they are close in value to raw returns. Another advantage of looking at the log returns of a series is that the relative changes in the variable can be seen and compared directly with other variables whose values may have very different base values. For the post amount, if we assume that some news and promotions last several days and the transmission and popularity of a brand depend on yesterday's, we can expect the hotness of today's keywords based on yesterday's, then we may still try to calculate log returns for the daily differences.

We denote the log returns of the time series:
R_t: Log Return of Original Share Price ($R_t = \log(P_t) - \log(P_{t-1})$)
R'_t: Log Return of Modified Share Price ($R'_t = \log(P'_t) - \log(P'_{t-1})$)
D_t: Log Return of Original Post Amount ($D_t = \log(A_t) - \log(A_{t-1})$)
D'_t: Log Return of Modified Post Amount ($D'_t = \log(A'_t) - \log(A'_{t-1})$)

The time series plots which regard the share price and post amount can be found separately in Figure 2.4.3 and 2.4.4. For the share price figure, the first row shows the three time series P_t, $\log(P_t)$, R_t for the original share price (no weekends) and the second row shows the three time series P'_t, $\log(P'_t)$, R'_t for the modified time series of share price (Friday's share price is applied to Saturday's and Sunday's). The first row shows the three time series A_t, $\log(A_t)$, D_t for the original post amount and the second row shows the three time series A'_t, $\log(A'_t)$, D'_t for the modified post amount (weekend data deleted).

The stationarity chart (Table 2.3) shows the stationary test results (p-value) for the time series mentioned above. For the share price, after taking log returns, results from all three tests indicate stationarity. The original time series for

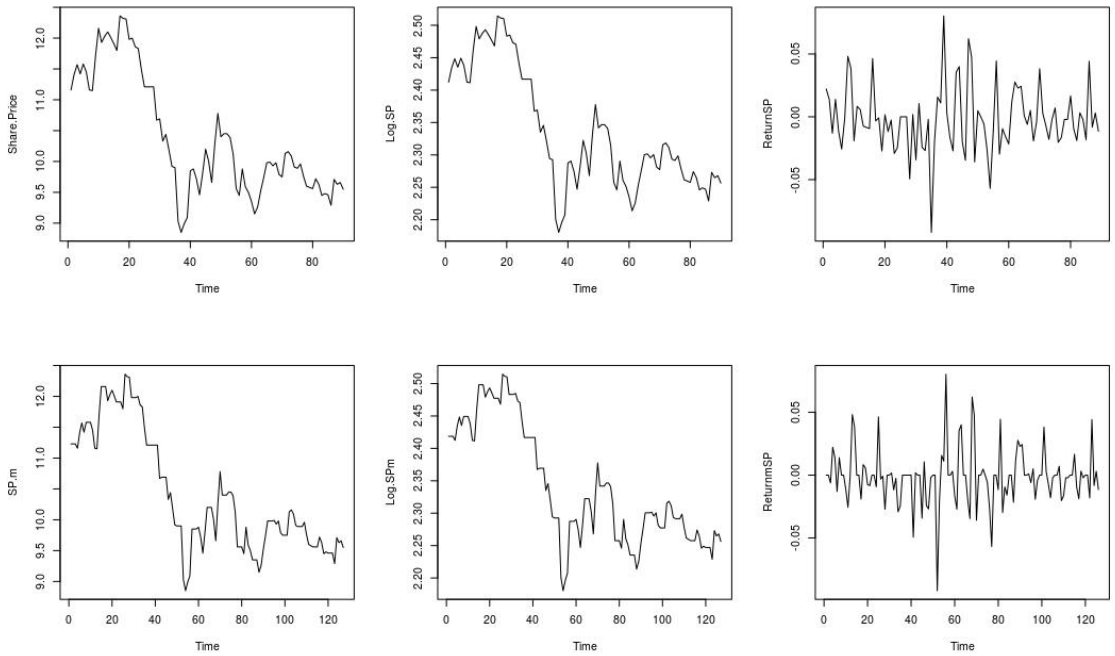

Figure 2.4.3: Time series for share price of Vanke (First row, from left to right: P_t, $\log(P_t)$, R_t. Second row, from left to right: P'_t, $\log(P'_t)$, R'_t).

the post amount seems to be much "more stationary" than the share price, but only after taking log returns did it become completely stationary. The modified post amount (delete weekend data) shows stationarity, even before taking the log transformation. However, the log transformation makes it somewhat nonstationary, but after calculating the difference, the time series becomes stationary again.

After confirming the stationarities, we use the log returns of share price and the log returns of post amount to conduct further analysis. The dot plots of the log returns of share price and log returns of post amount for both Group A and Group B, are shown in Figure A.5.1 in Appendix. From the graphs, we cannot see the relationship very clearly, hence regression models which regard log returns of post amount as an explanatory variable (x) and log returns of share price as

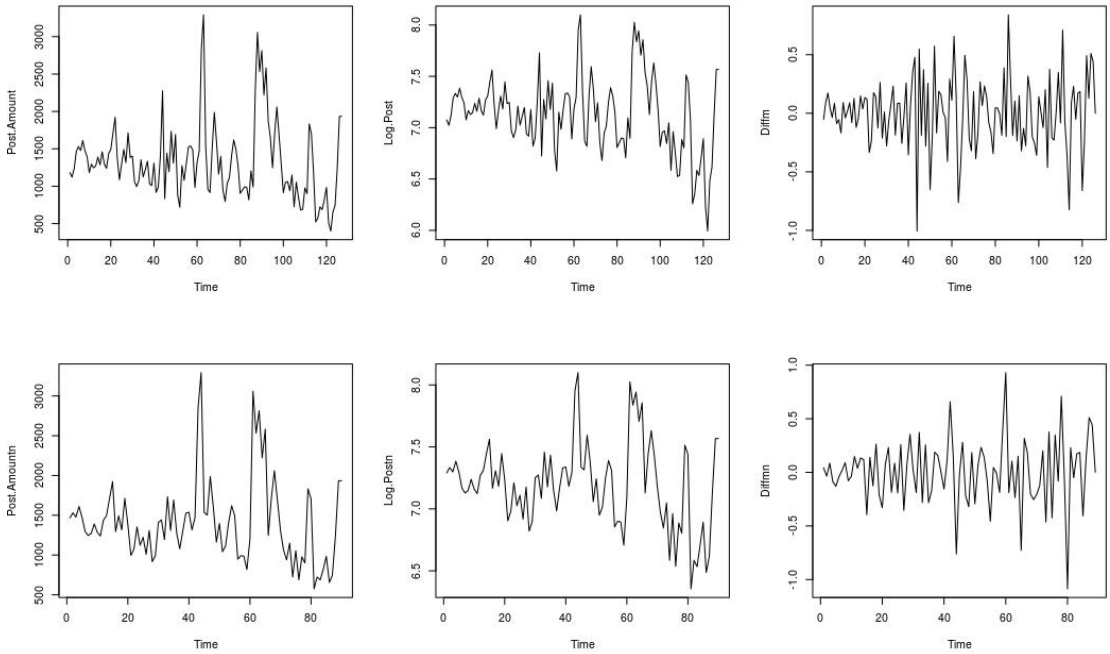

Figure 2.4.4: Time series for post amount of Vanke (First row, from left to right: A_t, $\log(A_t)$, D_t. Second row, from left to right: A'_t, $\log(A'_t)$, D'_t).

the dependent variable (y) are built. The outcomes of linear regression for Group A and Group B can be found in Figure A.5.2 and A.5.3 respectively. From the regression model A.5 and results listed in Appendix, it can be concluded that we can hardly find any evidence for the linear relationship between log returns of post amount and share price.

To analyse further the potential relationship between share price and post amount, we introduce kernel smoothing methods (Wand and Jones, 1994) to find smoother lines for the relationship and determine whether they are significantly different from zero or not.

The Nadaraya-Watson kernel regression (Nadaraya, 1964; Watson, 1964) and local

Table 2.3: Stationarity test results

Case	KPSS test	ADF test	PP test	Stationarity
P_t	0.01	0.6146	0.5847	non-stationary
$\log(P_t)$	0.01	0.5955	0.5448	non-stationary
R_t	0.1	0.03371	0.01	stationary
R'_t	0.1	0.01	0.01	stationary
A_t	0.1	0.09092	0.01	almost stationary
$\log(A_t)$	0.04968	0.207	0.01	almost stationary
D_t	0.1	0.01	0.01	stationary
A'_t	0.1	0.03309	0.01	stationary
$\log(A'_t)$	0.04968	0.08175	0.01	almost stationary
D'_t	0.1	0.01	0.01	stationary

polynomial regression (Fan et al., 1995) are applied as the kernel smoothers. The bandwidth of a kernel density estimate can be customised or selected using direct plug-in methods. We choose the direct plug-in approach, where the unknown functionals that appear in expressions for the asymptotically optimal bandwidths are replaced by kernel estimates (Ruppert et al., 1995). Both of the kernel methods are applied to Group A: log return of original share price R_t and log return of modified post amount D'_t.

A pair of results using different kernel regression estimates for Vanke are listed in Figures 2.4.5 and 2.4.6. By applying SiZer, short for Significance of Zero Crossings of the Derivative (Chaudhuri and Marron, 1999), we examine whether the derivatives of the smoother lines generated in Figures 2.4.5 and 2.4.6 are significantly different from zero or not. It is a method that looks across a range of bandwidths h and classifies the p-th derivative of the smoother into one of three categories: significantly increasing (blue), possibly zero (purple), or significantly negative (red) (Sonderegger, 2011); and grey means unable to detect. Vanke's

SiZer result for the first derivative is shown in Figure 2.4.7. Vanke's SiZer graph indicates that the derivatives of the smoother line is not significantly different from zero (with most parts purple). That is to say, we can not detect a significant relationship between log returns of share price and post amount for Vanke.

The results for other companies can be found in Appendix A.6, including time series plots for original/modified share price and post amount, plots of their log and log returns, regression estimate using local polynomials, and plots of SiZer. All the results show a limited relationship between share price and post amount.

Figure 2.4.5: Nadaraya-Watson kernel regression estimate with Bandwith 0.5 for Vanke (R_t and D'_t).

Figure 2.4.6: Regression estimate using local polynomials with Bandwith 0.5 for Vanke (R_t and D'_t).

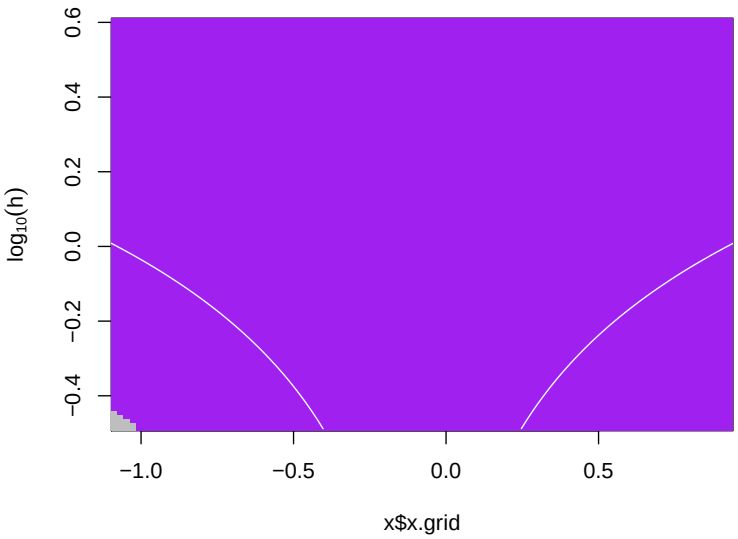

Figure 2.4.7: SiZer plot for Vanke (R_t and D'_t). X-axis represents the log returns of post amount D'_t divided into the grid of 100 units. Y-axis represents the $\log(h)$, and h are the bandwidths which we examine with. For each log return of the post amount, we calculate the first derivatives of the smoother line for different bandwidths h. For instance, for a specific bandwidth, if the first derivative of the smoother line for a specific value of log return is significantly larger than 0, the corresponding small grid in the SiZer plot would be blue, which suggests that this area within this bandwidth of the smoother line increases significantly.

2.5 Chinese Word Segmentation

Initial quantitative and time series analyses provide a brief description of the general patterns for the posts, but they are not sufficient for understanding what people generally posted about these companies. Thus, further textual analysis, as a complement of quantitative methods, would provide insight on the contents that people posted about. This textual analysis can be also referred to as text mining, a process of deriving the patterns and trends from texts through means such as statistical pattern learning. Typical text mining tasks include text categorization, text clustering, concept/entity extraction, sentiment analysis, document summarisation, and entity relation modelling. It is a part of statistical pattern learning, which aims to use artificial intelligence to learn from data.

In this and next two sections, we will first introduce Chinese word segmentation and term frequency analysis as basic text mining, and then explore cluster analysis for grouping posts based on their contents. Next chapters will further discuss sentiment modelling and topic modelling. For quantitative data analyses that only research the amount of posts, there are no other steps required for data pre-processing. Nevertheless, when attempting text mining, some additional steps are requisite.

Unlike English, which contains spaces between adjacent words, all Chinese characters are written together. Therefore, word segmentation is essential for Chinese text as an extra step before text mining. As the segmentation influences further textual analysis, it is crucial to find a way to segment Chinese sentence accurately and efficiently.

There are three R packages that could be used to accomplish Chinese word

segmentation. All of them are originally based on Java. The oldest one built on the MMSEG (Tsai, 2000) for Java Lucene Chinese Analyser, is named $rmmseg4j$. The problem of this method is that the corpus contains many old-fashioned words, making it insufficient to cope with modern microblog texts. The latest version enables the function of inserting a list of words into the corpus, which means one can add new words or new definitions to be used for word segmentation. But it still lacks the ability for detecting and inserting a large amount of new words efficiently.

Another package called rsmartcn, separates words by their semantics effectively and at a faster speed. It forms a port to Imdict Chinese Analyser, which applies an authoritative algorism ICTCLAS (Zhang et al., 2003) based on Hidden Markov Model (HMM) designed by the Chinese Academy of Sciences.

The latest Chinese word segmentation tool is Rwordseg (Li, 2012). It uses rJava to call the Java word segmentation tool Ansj (Sun, 2012), which is an improved port based on ICTCLAS. Compared to rsmartcn, it contains some advance functions, such as separating a combination of Chinese and English, adding Part-of-speech tagging, and inserting new external lexicons. We applied this package for Chinese word segmentation in further Weibo text mining.

2.6 Term Frequency

An intuitive first step for text mining after Chinese word segmentation is to calculate and visualise the frequency of occurrence of terms in posts. Before creating this term frequency list, we need to pre-process our dataset for preparation. First, we treat every post as a document and use the tm package (Feinerer, 2014)

to create a corpus. In linguistics, a corpus is a large and structured set of texts and can be used to undertake further statistical analysis. We clean up the data by removing punctuation, numbers, links and English letters. A group of Chinese stop words is removed using a wide-spread Chinese stop words list. Several dictionaries are added from the Sogou Word Database (one of the most widely-used Chinese pinyin input method providers) for popular cyberwords and social media buzz words, and a glossary for those specific companies, e.g. the CEO's name, affiliated brand names, competitor names, etc., is attached. The minimum topic-word length is set as two, in order to ensure the vocabulary for word frequency only contains sensible terms. This is because a single term with a single character in Chinese does not usually imply a well-defined semantic meaning.

A term frequency chart is a data frame containing most frequent terms with their frequencies, generated in a descending order. It is useful to develop a basic understanding of what the public generally mention about the companies. It can also be helpful to check the correctness of word segmentation, attach specific words to the lexicon, and add extra words to the stop words list.

The term frequency charts for our seven companies can be found in Figures 2.6.1, 2.6.2, 2.6.3, 2.6.4, 2.6.5, 2.6.6, and 2.6.7. All the term frequency charts are generated using all the posts that we collected, except Suning, for which we run a random sample of 200,000 posts, due to the large quantity of original dataset (732,684 posts).

We can find out very interesting results from the most frequent terms. Most of the charts include the companies' main business area, main products and most important competitors. Other valuable information such as main focus, main

regions and the name of CEO can also be discovered in several charts. We can also find out many promotion-related terms and understand how they promote themselves. The following sections will list some most frequent terms and discuss the information related to these terms.

2.6.1 Most Frequent Terms for Vanke and Biguiyuan

Vanke and Biguiyuan (see Charts 2.6.1 and 2.6.2) are two well-known real estate companies. The chairman of the board, Shi WANG (rank 2), is the most frequently mentioned term for the company Vanke, which reflects the huge influence of his leadership for Vanke. From the most frequent terms, we can see that Vanke is a real estate brand from Shenzhen (rank 12), and Baoli (rank 23) is an important competitor for Vanke. Many property owners (rank 11) talked about property management (rank 14). For the other real estate company, Biguiyuan , we can find that there were two main property projects during that period (Shilijintan, ranked 9; Phoenixcheng, rank 10). The main regions for their projects are Hainan (rank 14), Nanjing (rank 16), Wuhan (rank 22), and Qingdao (rank 39). It seems that Biguiyuan held a promotion for a free holiday (rank 34 and 35) for the followers (rank 3) of its official account (rank 4).

Rank	Words	(Translation)	Frequency	Rank	Words	(Translation)	Frequency
1	万科	Vanke	238990	21	北京	Beijing	8639
2	王石	Shi WANG (Chairman)	25841	22	广场	Plaza	8253
3	地产	Property	21645	23	保利	Baoli	8176
4	中国	China	19840	24	分享	Share	7691
5	房地产	Real Estate	15801	25	市场	Market	7655
6	企业	Enterprise	15577	26	建筑	Architecture	7618
7	城市	City	14979	27	亿元	Ten millions Yuan	7571
8	项目	Project	14008	28	上海	Shanghai	7333
9	公司	Company	13892	29	销售	Sales	7294
10	生活	Life	12898	30	幸福	Happiness	7277
11	业主	(Property) Owner	11313	31	住宅	Residence	7194
12	深圳	Shenzhen	11213	32	房子	House	7157
13	活动	Promotion	11117	33	国际	International	7137
14	物业	Property Management	10996	34	商业	Business	6698
15	土地	Land	10308	35	品牌	Brand	6384
16	中心	Center	10243	36	关注	Follow	6362
17	地址	Address	9727	37	行业	Industry	6321
18	房产	House property	9427	38	产品	Product	6106
19	集团	Group	9130	39	问题	Problem	6017
20	任志强	Zhiqiang REN	8663	40	社区	Community	5991

Figure 2.6.1: Term frequency for Vanke.

Rank	Words	(Translation)	Frequency	Rank	Words	(Translation)	Frequency
1	碧桂园	Biguiyuan	94559	21	期待	Expectation	4060
2	地址	Address	19275	22	武汉	Wuhan	4004
3	关注	Follow	16326	23	中国	China	3981
4	官方	Official	15302	24	支持	Support	3943
5	微信	Wechat	14775	25	销售	Sales	3695
6	城市	City	13254	26	现场	Scene	3662
7	花园	Garden	12206	27	幸福	Happiness	3620
8	黄金	Golden	10265	28	好运	Good luck	3609
9	十里金滩	Shilijintan	7200	29	中心	Center	3453
10	凤凰城	Phoenixcheng	6602	30	小伙伴	Buddy	3332
11	酒店	Hotel	6223	31	业主	Owner	3195
12	活动	Promotion	5814	32	国际	International	3194
13	生活	Life	5302	33	沙滩	Beach	3141
14	海南	Hainan	5148	34	免费	Free	3116
15	计划	Plan	4619	35	度假	Holiday	3062
16	南京	Nanjing	4612	36	五星级	Five-star	2997
17	希望	Hope	4280	37	别墅	House	2901
18	凤凰	Phoenix	4246	38	开盘	Opening quotation	2798
19	项目	Project	4165	39	青岛	Qingdao	2634
20	引进	Bring in	4160	40	全国	Nationwide	2539

Figure 2.6.2: Term frequency for Biguiyuan.

2.6.2 Most Frequent Terms for Suning and Guomei

For Suning and Guomei, two well-known electric appliance companies, in Charts 2.6.3 and 2.6.4, we can see from the frequency chart that both of them provide online purchase options, which are very popular ("Online" ranked top 3 for

both). The most important competitors for both companies is Jingdong, a Chinese electronic commerce company (rank 17 and 16), but not each other. Individuals mentioning Guomei are likely to mention Suning (rank 15), but there are many posts only containing Suning (we cannot find "Guomei" in Suning's chart). The most popular products for Suning are refrigerator (rank 10), phone (rank 11), and computer (rank 19). It seems that Suning had a promotion on the National day (rank 14) and had a lucky draw for its followers (rank 6, 32 and 34). While for Guomei, the promotion is likely to be for the eighth year anniversary (rank 25), and the membership policy, which is mentioned very frequently (rank 12), might be one of their business strategies.

Rank	Words	(Translation)	Frequency	Rank	Words	(Translation)	Frequency
1	苏宁	Suning	155896	21	大家	Everyone	7192
2	地址	Address	99599	22	上海	Shanghai	7169
3	苏宁易购	Suning Online	46280	23	生活	Life	6932
4	希望	Hope	20588	24	第一届	First session	6886
5	支持	Support	20061	25	中国	China	6755
6	好运	Good luck	18977	26	服务	Service	6666
7	首发	First release	15605	27	诺基亚	Nokia	6423
8	购物	Shopping	15175	28	狂欢	Carnival	6230
9	期待	Expectation	13962	29	爆发	Outbreak	5457
10	冰箱	Refrigerator	11050	30	平板	Tablet	5456
11	手机	Phone	10567	31	独家	Exclusive	5408
12	努力	Endeavor	9678	32	中奖	Win a prize	5386
13	电器	Electric appliance	9537	33	加油	Go for it	5378
14	国庆	National Day	9485	34	幸运	Lucky	5199
15	活动	Promotion	9043	35	快乐	Happy	5168
16	精彩	Wonderful	8812	36	洗衣机	Washing machine	5130
17	京东	Jingdong	8395	37	空调	Air conditioner	5044
18	电商	E-commerce	8230	38	运气	Luck	5015
19	电脑	Computer	8194	39	万人空巷	The whole town turns out	4942
20	广场	Plaza	7691	40	关注	Focus	4903

Figure 2.6.3: Term frequency for sampled Suning.

Rank	Words	(Translation)	Frequency	Rank	Words	(Translation)	Frequency
1	国美	Guomei	292552	21	用心	Attentively	10638
2	地址	Address	182122	22	中奖	Draw a prize	10602
3	在线	Online	154894	23	相伴	Accompanied	10104
4	支持	Support	41071	24	老友	Old friend	10001
5	希望	Hope	38756	25	八周年	The eighth anniversary	9993
6	好运	Good luck	38418	26	惊喜	Surprise	9716
7	期待	Expectation	29203	27	爆发	Outbreak	9626
8	时光	Time	27605	28	中国	China	9494
9	激情	Passion	21502	29	喜欢	Like	9423
10	备战	Preparation	19508	30	首发	First release	9364
11	电器	Electrical appliance	17356	31	体验	Experience	9361
12	会员	Member	16724	32	开门红	Good start	9347
13	努力	Endeavour	14153	33	不停	Non-stop	9121
14	生活	Life	14117	34	越来越	More and more	8943
15	苏宁	Suning	13962	35	运气	Luck	8836
16	京东	Jingdong	13787	36	电商	E-commerce	8474
17	幸运	Lucky	12558	37	手机	Phone	8275
18	活动	Promotion	12013	38	关注	Follow	7938
19	彩电	Colour TV	11053	39	知足常乐	Happiness consists in contentment	7712
20	家电	Household appliances	10756	40	服务	service	7526

Figure 2.6.4: Term frequency for Guomei.

2.6.3 Most Frequent Terms for Donghang, Biyadi and Maotai

The other three companies are not natually connected, but we put them together in this subsection. The most frequently mentioned destinations for the China Eastern Airlines (Donghang) seem to be Ningbo (rank 16), Shanghai (rank 18) and Beijing (rank 26). Passengers (rank 34) or other Weibo users often posted about arrivals and departures (rank 10 and 11), and had some complaints about the lateness and cancellation of flights (rank 30 and 31). Biyadi, which is an electric auto-mobile company (rank 2 and 22), might focus on autonomous innovation and technology (rank 19 and 13). It seems that the company held a lottery draw (rank 8) for a promotion of its tenth anniversary (rank 26). Maotai is a famous white spirit (rank 7) in China (rank 5). It might be purchased for celebrating important birthdays (rank 10), and its largest competitor is Wuliangye (rank 12). Weibo users talked about Maotai's price (rank 15), stock price (rank 33) and its distributor (rank 31).

Rank	Words	(Translation)	Frequency	Rank	Words	(Translation)	Frequency
1	东方航空	China Eastern Airlines	29356	21	目前	Current	3675
2	东航	Abbreviated CEA	15531	22	延误	Delay	3435
3	机场	Airport	13778	23	管制	(Air Traffic) Control	3340
4	公司	Company	12847	24	航行	Navigation	2911
5	中国	China	11855	25	人员	Crew	2740
6	航班	Flight	8440	26	北京	Beijing	2541
7	时间	Time	7937	27	国际	International	2535
8	航空	Aviation	6765	28	机票	Airline ticket	2460
9	飞机	Airplane	6758	29	服务	Service	2421
10	到达	Arrival	6578	30	晚点	Behind schedule	2242
11	起飞	Departure	6490	31	取消	Cancel	2093
12	计划	Planned	6322	32	国航	Air China (Airline)	2033
13	动态	Trends	6268	33	虹桥	Hongqiao (Airport)	1946
14	最新	Latest	6245	34	乘客	Passengers	1827
15	小时	Hour	5480	35	提前	Ahead of schedule	1803
16	宁波	Ningbo	5465	36	电话	Phone (Number)	1743
17	分钟	Minute	5199	37	迫降	Forced landing	1710
18	上海	Shanghai	4561	38	航线	Air route	1683
19	原因	Reason	4293	39	浦东	Pudong (Airport)	1666
20	预计	Estimated	3770	40	旅客	Traveler	1664

Figure 2.6.5: Term frequency for Donghang.

Rank	Words	(Translation)	Frequency	Rank	Words	(Translation)	Frequency
1	比亚迪	Biyadi	133768	21	正式	Official	7594
2	汽车	Automobile	33140	22	电动	Electric (vehicle)	7477
3	车型	Vehicle type	18845	23	启动	Start (a car)	7196
4	关注	Follow	18678	24	动力	(Electro)dynamic	6986
5	发布	Release	16812	25	发动机	Engine	6956
6	机会	Chance	14850	26	十年	Ten years	6508
7	参与	Participate	14207	27	预计	Expect	6498
8	每天	Everyday	14095	28	万元	Ten thousand Yuan	6485
9	抽奖	Lottery draw	13972	29	市场	Market	6471
10	小伙伴	Buddy	13939	30	配置	(Vehicle) configuration	6389
11	身边	Beside	13649	31	搭载	Carry	6051
12	含有	Contain	13408	32	心动	Attractive	6011
13	科技	Technology	11616	33	狂欢	Carnival	5982
14	上市	Be listed	11255	34	上海	Shanghai	5896
15	中国	China	9769	35	电动车	Electrocar	5886
16	复制	Copy	9135	36	系统	System	5785
17	正文	Main body	9065	37	能源	Energy	5657
18	品牌	Brand	8657	38	曝光	Exposure	5366
19	自主	Autonomous	7855	39	设计	Design	5220
20	技术	Technique	7765	40	售价	Retail price	4524

Figure 2.6.6: Term frequency for Biyadi.

Rank	Words	(Translation)	Frequency	Rank	Words	(Translation)	Frequency
1	茅台	Maotai	160119	21	支持	Support	5171
2	茅台酒	Maotai liquor	31377	22	品牌	Brand	4987
3	贵州	Guizhou	22950	23	问题	Question	4983
4	地址	Address	19150	24	行业	Industry	4920
5	中国	China	18403	25	北京	Beijing	4845
6	关注	Follow	13816	26	上海	Shanghai	4788
7	白酒	White spirit	13455	27	销售	Sales	4722
8	美酒	Good Liquor	12302	28	企业	Enterprise	4577
9	围观	Crowd	11569	29	希望	Hope	4570
10	大寿	Important birthday	11219	30	价值	Value	4437
11	河神	River god	11194	31	经销商	Distributor	4206
12	五粮液	Wuliangye	11073	32	飞天	Feitian	3998
13	公司	Company	7995	33	股价	Stock price	3803
14	市场	Market	7020	34	新闻	News	3674
15	价格	Price	6639	35	银行	Bank	3596
16	一瓶	A bottle	5966	36	分享	Share	3440
17	投资	Invest	5964	37	股票	Stock	3431
18	集团	Group	5424	38	产品	Product	3374
19	旅行	Travel	5276	39	董事长	Board Chairman	3281
20	消费	Consume	5177	40	啤酒	Beer	3248

Figure 2.6.7: Term frequency for Maotai.

2.6.4 Beyond Term Frequency

In addition to the findings in the previous subsections, further text mining such as clustering and topic modelling are valuable to "combine" the most frequent words and to help the reader extract comprehensive meanings from the text. If we generate the term frequency charts or topics for a different period of time, the differences between previous topic terms and new topic terms can also provide us with an idea of the evolution of topics. A word cloud based on term frequency can be created and found in Figure A.7.1 in the Appendix. It displays the most frequent words in an easily intelligible way.

2.7 Cluster Analysis

A term frequency chart can provide the top keywords that individuals are the most likely to mention for a specific company, but it can hardly group those posts based on their contents addressing different aspects and provide a comprehensive description of what individuals are posting about. Cluster analysis is to group a set of objects in a way such that the objects in the same group (called cluster) are more similar to each other than to those in other clusters. In cluster analysis, we treat each post as an object, and we intend to group the posts based on the key words which they contain.

Table 2.4: Structure of a term-document matrix

	Document 1	Document 2	Document 3	Document 4	...
Term 1	1	0	1	0	...
Term 2	0	1	0	1	...
Term 3	1	2	1	0	...
Term 4	0	0	1	1	...
Term 5	0	1	0	2	...
...	...	...	...	...	...

Initially, a term-document matrix is created for our dataset, which is a mathematical matrix that contains the frequency of terms that occur in each post. In a term-document matrix, rows correspond to the terms in the collection and columns correspond to the documents (posts). That is to say, each column represents a post, and row names are all the terms that occur in this corpus. The structure of a term-document matrix can be found in Table 2.4. The element of the matrix is the number of occurrence of the terms in those posts. It is an effective way to transfer textual contents into a mathematical form. In this way, each post can be located in a high-dimensional space for terms, and the dimensions can be

reduced by controlling the sparsity, which will be described later.

Using term-document matrix, we are able to calculate the distances between posts, and then group the posts into different clusters by different clustering methods in a high-dimensional space. The clusters can be visualised in a bivariate or a trivariate plot, in which each post is represented by a point according to principal components or multidimensional scaling. As an example, in Figure 2.7.1 the clusters are visualised by two leading principal components in the two-dimensional space. Three different cluster algorithms are adopted in our research: k-means (Hartigan and Wong, 1979), k-medoids (Kaufman and Rousseeuw, 1990) and hierarchical clustering (Kaufman and Rousseeuw, 1990).

First, the traditional k-means method is applied to the term-document matrix. k-means clustering aims to partition n observations (documents) into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. Figure 2.7.1 shows a clustering result with 3 clusters from the k-means method for 2000 posts of Vanke.

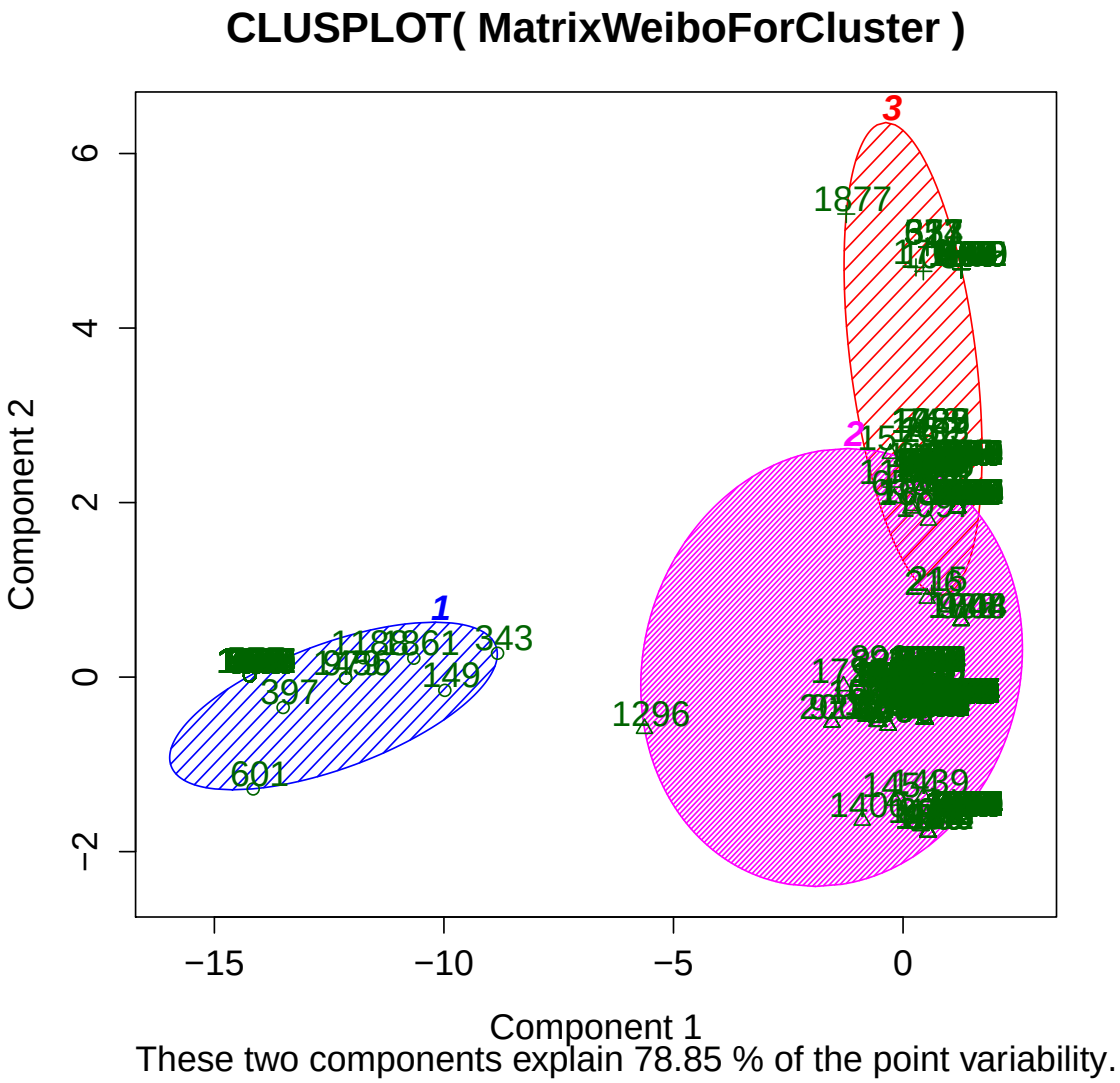

Figure 2.7.1: Cluster plot from k-means. Visualised by two leading principal components. Sparsity = 0.94. Number of clusters k = 3.

The Partitioning Around Medoids (PAM) (k-medoids) method by Kaufman and Rousseeuw (1990) is an alternative method for clustering. It partitions the data into k clusters around medoids, which are the most centrally located points whose average dissimilarity to all the data points in this cluster is minimal.

The k-medoids (PAM) algorithm first selects k representative objects (medoids) from all the data points as initialisation, then assigns other data points to the closest medoid. For each medoid m and each data point n associated to m, the algorithm swaps m and n and recomputes the total cost of the configuration (that is, the average dissimilarity of n to all the data points associated to m). If the total cost of the configuration increases in the previous step, undo the swap. As the swaps are repeated and the cost are calculated, the algorithm updates the medoids until there is no change in the assignments.

Compared to the k-means approach, the function PAM has the following beneficial features (Reynolds et al., 2004). The k-medoids minimises a sum of general pairwise dissimilarities instead of a sum of squared Euclidean distances, and there are many possible choices of the dissimilarity measures, e.g. Manhattan distance, Euclidean distance. It is more robust as it follows the same logic as using median instead of mean, which is less affected by outliers and skewed data.

An extension for PAM is CLARA (CLustering LARge Applications), which relies on the sampling approach to handle large data sets. Due to the complexity of the k-medoids, it takes longer time than the k-means method. Instead of finding medoids for the entire data set, CLARA draws a small sample from the data set and applies the PAM algorithm to generate an optimal set of medoids for the sample. The quality of resulting medoids is measured by the average dissimilarity between every object in the entire data set and the medoid of its cluster.

A drawback of the clustering methods lies in the number of clusters needing to be selected (Pelleg et al., 2000). The correct choice of k is often ambiguous, with interpretations depending on the shape and scale of the distribution of points. In

our study, the number of clusters is set to three, but from Figure 2.7.1, we can just see two completely separate blocks and the number of clusters which we can identify from the second block is ambiguous, i.e. it can be one or more, although we set two in this example.

A silhouette plot (Kaufman and Rousseeuw, 1990) is a graphical display which allows the user to select the optimal number of clusters. A result based on Biyadi's data can be found in Figure 2.7.2, with three obvious clusters and two very small clusters. The S_i for a data point i in the corresponding silhouette plot (see Figure 2.7.3) is calculated as:

$$S_i = \frac{b_i - a_i}{\max(a_i, b_i)} \tag{2.1}$$

In this formula, the a_i is the average dissimilarity of i with all other data within the same cluster, and the b_i is the lowest average dissimilarity of i to any other points not in this cluster. When S_i is close to one, it means i is properly clustered; but when S_i is close to minus one, it means this point should be grouped to its neighbouring cluster. An S_i near to zero means that the datum is on the border of two clusters. It can be concluded that the larger the average S_i is, the tighter all the data in the cluster are. The silhouette plot in Figure 2.7.3 for the k-medoids clustering (Figure 2.7.2) shows Cluster 3, 4, 5 are tightly grouped, Cluster 1 is moderately grouped, but Cluster 2 is not well grouped.

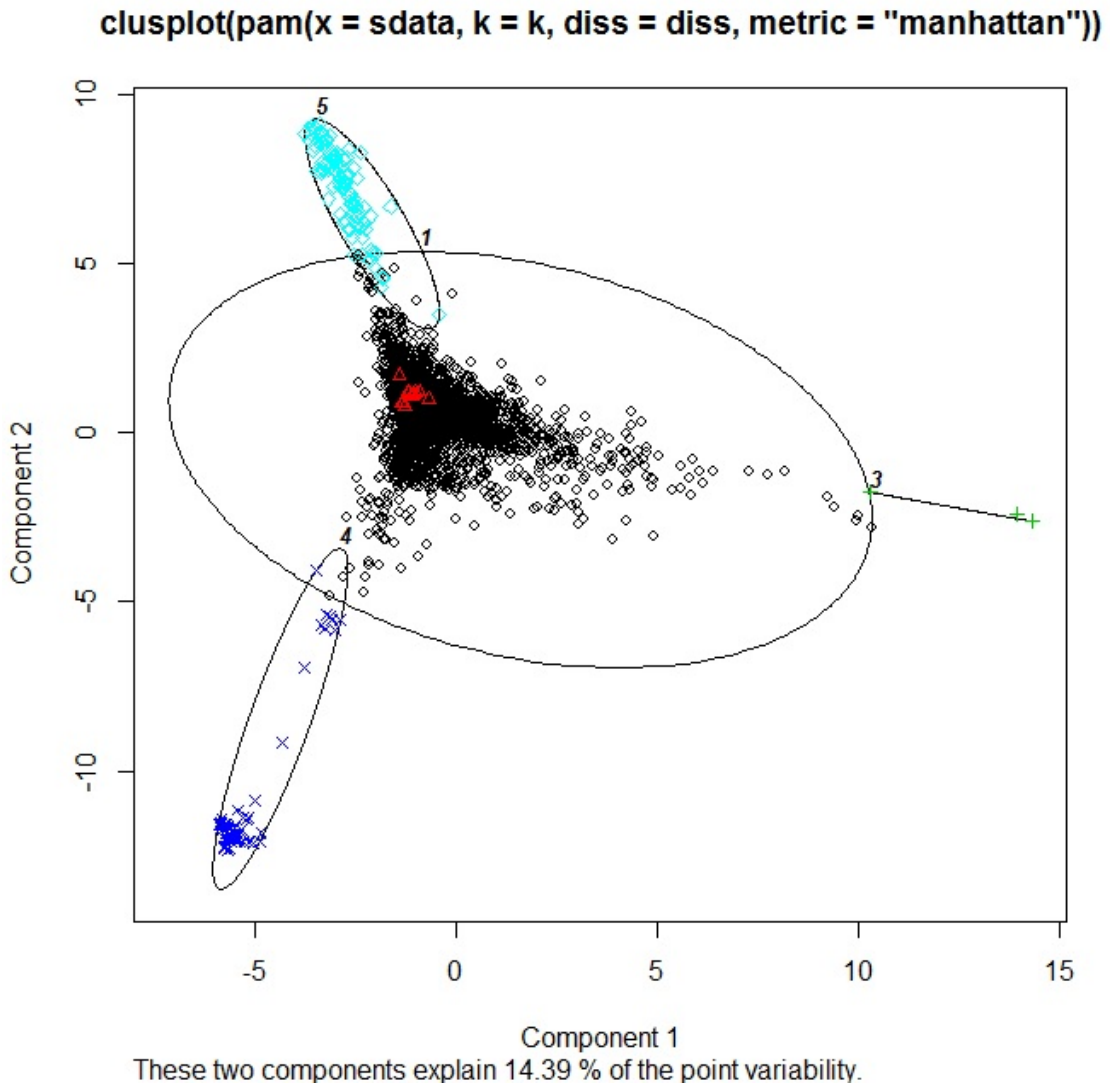

Figure 2.7.2: Cluster plot from PAM. Visualised by two leading principal components. Number of clusters k = 5. Sparsity = 0.97. Contains 116 terms.

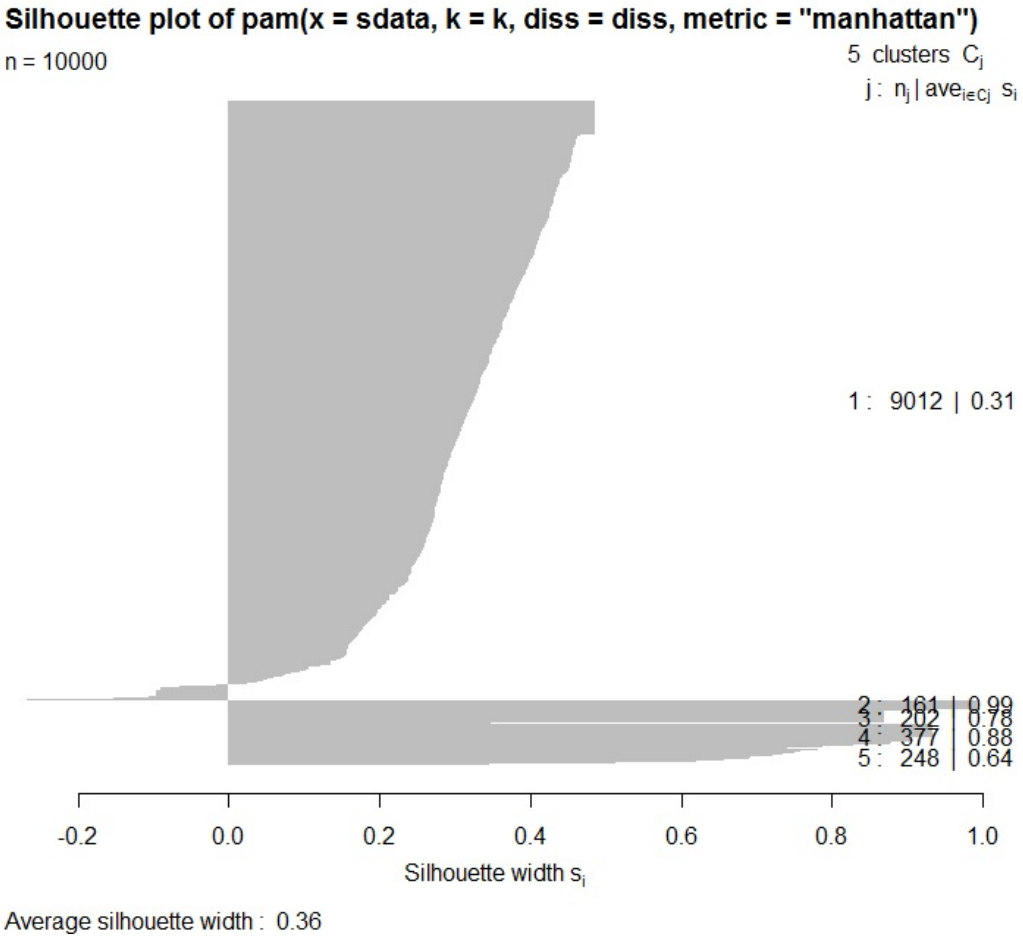

Figure 2.7.3: Silhouette plot for PAM. Sparsity = 0.97.

Sparsity is an important index for data pre-processing before clustering. Setting an appropriate sparsity rate is useful for deleting redundant words and reducing dimensionality. The term-document matrix will keep those terms with high occurrence and remove those terms with low occurrence. A sparsity rate of range 0 to 1 is set: the higher the rate, the larger amount of terms that the matrix keeps. For instance, in k-medoids, when sparsity is set to be 0.95, it means that all the words that remain appear in at least 5 % of the documents. From the results based on our dataset, if the sparsity rate is set to be 0.97 (see Figures 2.7.2 and 2.7.3),

the amount of remaining terms will be 116 (out of 19,726 terms); and when it decreases to 0.96 (see Figures A.8.3 and A.8.4 in Appendix) or 0.95 (see Figures A.8.5 and A.8.6 in Appendix), the amounts of remaining terms become 70 and 49. If sparsity increases to 0.98, there are too many terms left in the term-document matrix and the result for PAM can not be visualised well. We may switch to CLARA to calculate the results, but accuracy will decrease at the same time. The clustering result from CLARA and its corresponding silhouette plot can be found in Figures A.8.1 and A.8.2 in Appendix A.8.

Hierarchical clustering (Ward, 1963) is also considered in our research. The results of applying k-means or k-medoids clustering algorithms depend on the choice for the number of clusters. In contrast, for hierarchical clustering methods, this specification is unnecessary. Because the result from hierarchical clustering is a dendrogram, the number of clusters can be fixed by setting the cut positions in a tree. It can be divided into two categories: agglomerative (bottom-up) and divisive (top-down). Agglomerative means merging a selected pair of clusters into a single cluster; and divisive means dividing one of the existing clusters into two at each stage. Furthermore, there are three possible types of agglomerative clustering: single linkage (Gower and Ross, 1969), complete linkage (Defays, 1977) and group average (Kaufman and Rousseeuw, 1990).

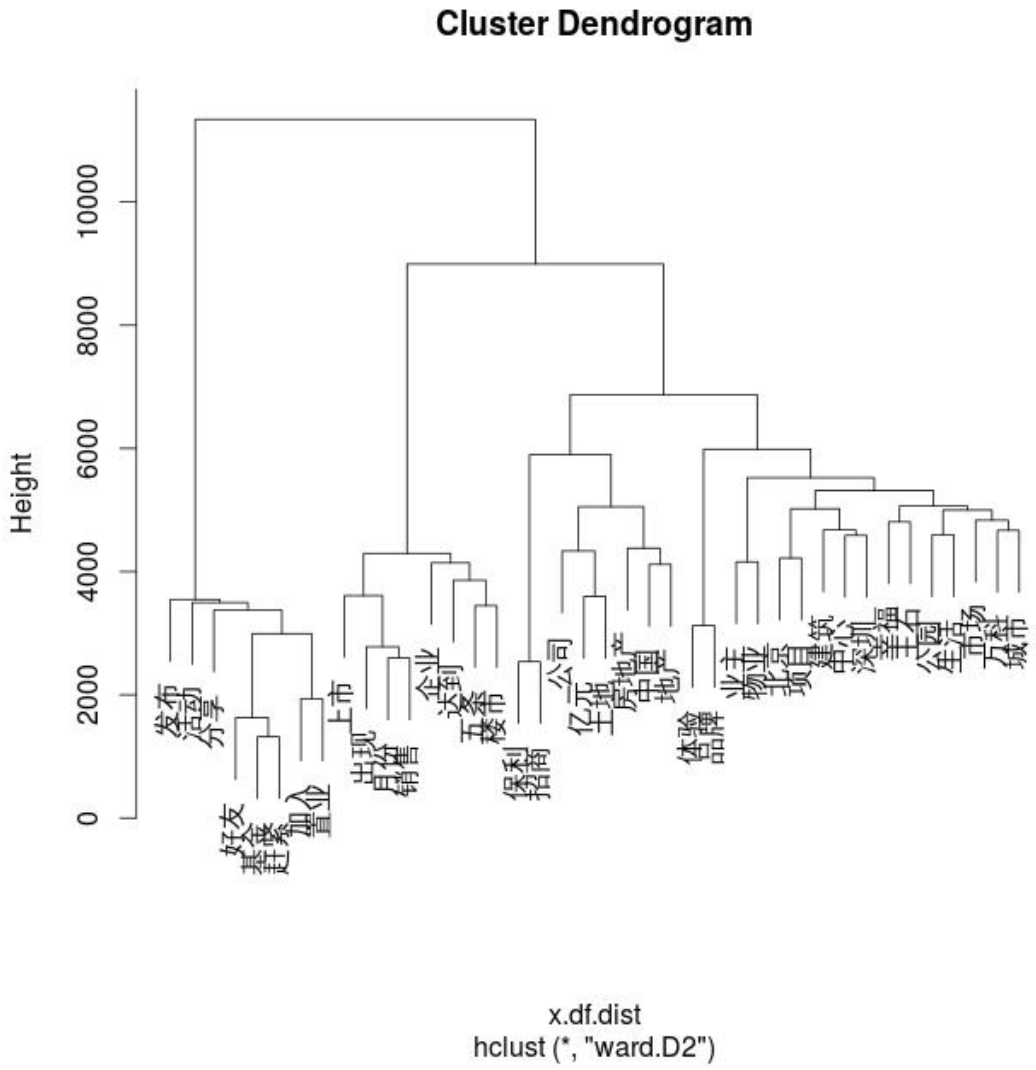

Figure 2.7.4: Agglomerative hierarchical clustering. Sparsity = 0.965

The agglomerative hierarchical clustering result generated in Figure 2.7.4 appears to be the most appropriate one visually with a sparsity rate of 0.965. It shows the remaining terms on the leaves. For hierarchical clustering, without setting an suitable sparsity, an unclear result shown in Figure A.8.7 in the Appendix with a sparsity rate of 0.97 can be generated with excessive dimensions (terms). Only a

little difference in the sparsity setting can result in very diverse results.

The output of clustering can bring in useful information. For instance, digging into the clusters of posts from k-means (Figure 2.7.1) provides us some insights into why those posts are clustered in these three clusters. By investigating the posts by indexes in the cluster 1, we can find most posts in this cluster are about a promotion named "I love my family relay race" which was held by Vanke official account and if the followers re-posted this promotion, there would be a chance to win gifts. Cluster 3 is mostly dominated by different news for real estate and also includes some small promotions. Most posts are grouped into Cluster 2, but it is hard to find out any common points for Cluster 2: it is a mixture with news, promotions, users' comments, etc.

The result from hierarchical clustering shows only a few key words due to the limitation of display. For the example in Figure 2.7.4, if we cut around the height of 6000, we can have four clusters. The terms from the first cluster indicate that this cluster is about a promotion for joining, sharing and reposting. The second cluster is about news of real estate, including sales and IPO. The third one mentions Vanke's competitor Baoli, and the fourth one can be related to the experience of Beijing and Shengzhen projects for the property owner and property management. Although the results from hierarchical clustering seem to be more easily interpreted than the other clustering methods, at the same time, they lose much information. To make sure the graph is visible, the sparsity of hierarchical clustering must be set relatively low, and for this reason many informative terms disappear.

Although the clustering methods can be an intial step to group the posts, we can see that both k-means and hierarchical clustering are neither an efficient way to

visualise the clusters, nor an effective way to figure out the reason of clustering. For the k-means and k-mediods, we can only represent the high dimensional clusters in the bivariate or trivariate plots, and it is necessary to examine the posts one by one to figure out why the algorithm generates those clusters. For the hierarchical clustering, we can only cluster a few posts and much information would be lost for visualisation, because the number of terms presented is limited, i.e. the display of the terms becomes unclear as the number of terms increases. To tackle these difficulties, we will introduce topic modelling in Chapter 4, which better achieves the goal of clustering posts and deriving understandable information from text based on the most important terms.

2.8 Summary

This chapter has began with an extensive literature review for Twitter and Weibo data analyses. There were many papers that explore hotspots and make predictions based on Micro-blogs. We presented previous work which employed Twitter data to predict disease outbreaks, voting behaviours, and stock price movements. Approaches of sentiment analyses for Twitter were introduced afterwards. Some research closely linked to our analysis have been discussed in detail.

We reviewed several methods for data acquisition, which was an initial challenge for Weibo data mining. Seven well-known but moderately popular companies were chosen as our research data source to guarantee a certain number of posts every day as well as avoid hitting the maximum limit. Following the literature review and the description of data collection, we developed various angles for exploratory

data mining. Pure quantitative approaches were considered first. The analysis of the time of the posts concluded the general patterns of time that the users follow for posting. The relationship between the intensity of posts and share prices was investigated using linear regression and kernel regressions, but the results showed a limited relationship between share price and post amount.

Initial text mining then began with Chinese word segmentation as all Chinese characters are written together. Term frequency charts provide basic understandings of what the public generally mention about the companies, check the quality of pre-processing for texts, as well as offer insights into further text mining. Cluster analyses, i.e. k-means, K-mediods, and Hierarchical clustering, have been introduced to group the posts based on their contents addressing different aspects.

Motivated by these exploratory data analyses and the relevant literature discussed in this chapter, the next two chapters will examine these in depth and look towards combining the sentiment analyses with the univariate and multivariate time series modelling, and developing the Randomness Reduction algorithm for topic modelling.

Chapter 3

Time Series Modelling on Sentiment

3.1 Introduction to Sentiment Analysis

It is crucial to analyse the sentiment tendencies of posts, because the amount of posts per day mentioning a company can only reflect the "popularity" of this company, but not the general attitudes, i.e. there might be a huge amount of negative comments. Sentiment analysis is a way to obtain the overall emotional polarity of all the posts and generate fundamental public attitudes and views. Sentiment analysis refers to an application of natural language processing to identify and extract subjective information from source materials, and it aims to determine the attitude of a speaker or a writer with respect to some topics or the overall contextual polarity of a document. The simplest sentiment analysis is choosing a word list containing different polarities, such as positive and negative. This "dictionary" can be compared with the real text, and then, by word matching, a score can be calculated.

Hownet Chinese Message Structure Base designed by Dong and Dong (2006) of

the Chinese Academy of Sciences is a well-known Chinese lexicon. I choose to use the positive/negative critical dictionary, but not the positive/negative emotional dictionary, because in most cases, the words from emotional dictionary in the posts mainly reflect the authors' temporary emotions, but not the attitudes about these companies. The positive and negative critical vocabulary includes 3730 and 3116 Chinese words respectively, which means positive words occur 20 percent more than negative words. A limitation is that this dictionary includes mostly formal and relatively old-fashioned words, so it may not suit short, colloquial and up-to-date Weibo posts.

After data cleaning and word segmentation, the matching is carried out by comparing the words in the posts to the lexicons of positive and negative terms. An example can be found in Table 3.1: one sentence such as "I like Barnet" will give out one True and two False for positive matching ("like" is in the positive lexicon, and "I" and "Barnet" are not), and three False for negative matching (none of the three terms are in the negative lexicon).

Table 3.1: Example of lexicon-based sentiment matching

Matching:	" I	like	Barnet "
Positive	False	True	False
Negative	False	False	False

We add up the number of True matches for both positive and negative sentiments, and then the sentiment score was calculated by the sum of positive matches minus the negative matches. That is to say, the higher the score, the more positive the sentiment.

As the first attempt, sentiment scores for the posts about one company, Huawei,

over three months are calculated. Each post is given an individual score for its positive/negative polarity. The average result for these three months was around 5, which seems to be too high. After checking, "shi" (its meaning in English can be "yes", "right", or "is") is included in the positive vocabulary, but mostly it just means "is". By deleting "shi", the score seems to be more reasonable with 2.203 on average for 139,514 posts. This word list is problematic in that the words included are mostly formal and old-fashioned (not designed for microblogs).

Another lexicon resource entitled Chinese Emotional Words Ontology was created by Chen (2009). It is based on a well-known Six Universal Emotions System (Ekman, 1993). In this psychological research, facial expressions were classified into six distinct universal emotions: Disgust, Sadness, Happiness, Fear, Anger and Surprise. On the basis of Ekman's classification, this Chinese version of Emotional Words Ontology divided the happiness emotions into two further categories: pure Happiness and Goodness (positive attitude). In total, there are 27,466 words, which were divided into 7 categories of 21 small sub-groups. It not only assigned emotional categories to the words and labelled the words as noun, verb, adjective, adverbial, network language, idiom or preposition, but also marked every word's strength. It is intended to provide a convenient and reliable means of effective computing for Chinese text sentiment tendency analysis, which can be used to solve the problem of general orientation analysis (positive/neutral/negative), and at the same time can also be used to solve the multiple category classification of emotional problems.

Firstly, we consider the positive and negative polarities for our empirical study. The positive/negative histogram of sentiment scores for a company, Biyadi, over

three months are shown in Figure 3.1.1. The comparative polarity histograms using Chinese Emotional Words Ontology and Hownet Chinese Message Structure Base for Guomei are shown in Figure B.1.1 in the Appendix. It can be discovered that all the positive/negative polarity histograms are positively skewed, with the highest frequency at zero, diminishing much faster on the left hand side (negative values) than on the right hand side (positive values).

By performing the D'Agostino test (D'Agostino, 1970) for skewness of normally distributed data (see Figure B.1.2 in the Appendix), it is shown that the distribution is skewed to the right with a rate of 0.9832 and with high significance. The rate of skewness is calculated using R and the detailed formula can be found in the paper by D'Agostino (1970). Any threshold or rule of thumb is arbitrary, but here it is suggested that if the skewness is greater than 1.0 (or less than -1.0), the skewness is substantial and the distribution is far from symmetrical. That is to say, our data is with mild skewness. As the sentiment's distributions are discrete, Poisson and Negative Binomial distribution fitting (results in Figure B.1.3) are tested, but neither of them fits well.

The 3D plot of positive and negative sentiments in Figure 3.1.2 illustrates a clearer relationship between positive and negative sentiments. The coincidental occurrence of some posts that contain both positive and negative sentiments does not seem rare. Therefore, the correlation test was considered to test if there is any correlation between positive and negative sentiments. The result in Figure B.1.5 shows that the correlation is significant at a rate of 0.06, and after checking and deleting some redundant sentiment words, the correlation becomes even higher with a rate of 0.17. This phenomenon might suggest that some posts contain both positive and

negative sentiments: individuals might express contrasting opinions in their posts. Figures B.1.6 and B.1.7 give the top 50 positive and negative words. The top 50 positive words include wonderful, sharp, positive, leading, faithful, etc.; and top 50 negative words include drawback, incident, pollution, punish, etc.

Figure 3.1.3 shows the seven dimensions sentiment charts: Goodness, Happiness, Surprise, Anger, Fear, Sadness and Disgust. The specific scores can be found in Appendix B.2.1. The Goodness score has the widest range and the largest value, followed by the Happiness score and the Disgust score. The scores for Surprise and Anger are low. Correlations and paired sentiment plots are listed in Figures 3.1.4, B.2.2, B.2.4. The high significance level of correlations might result from the discrete nature and the small figures for seven sentiments. All the values of correlations are not large, with the highest correlation between Happiness and Goodness, which is larger than 0.15. It is reasonable as the two sentiments are closely related to each other. Surprisingly, the Disgust score positively correlated with the Goodness and Happiness scores, and it might indicate some posts contain more than two sentiments and reflect extreme emotional polarities.

Besides the lexicon-based approach, other approaches applying supervised learning tools such as support vector machine (SVM) for classification could be adopted for sentiment analysis. Due to the difficulties for obtaining relevant training data and time-consuming of manually tagging, our analyses will be based on the lexicon-based approach.

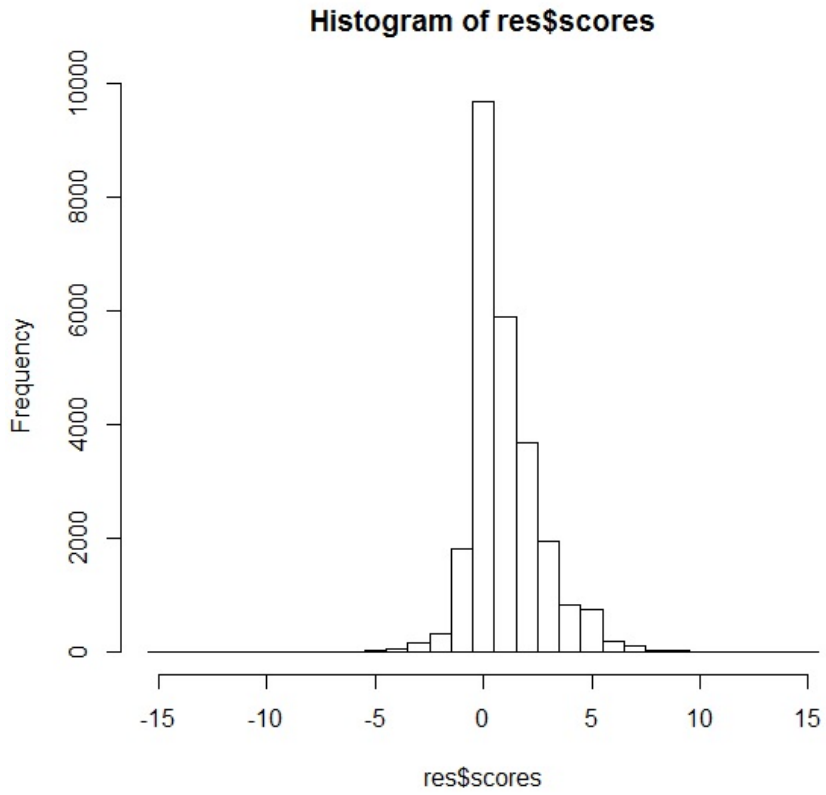

Figure 3.1.1: Sentiment polarity for company Biyadi using 3 months' data.

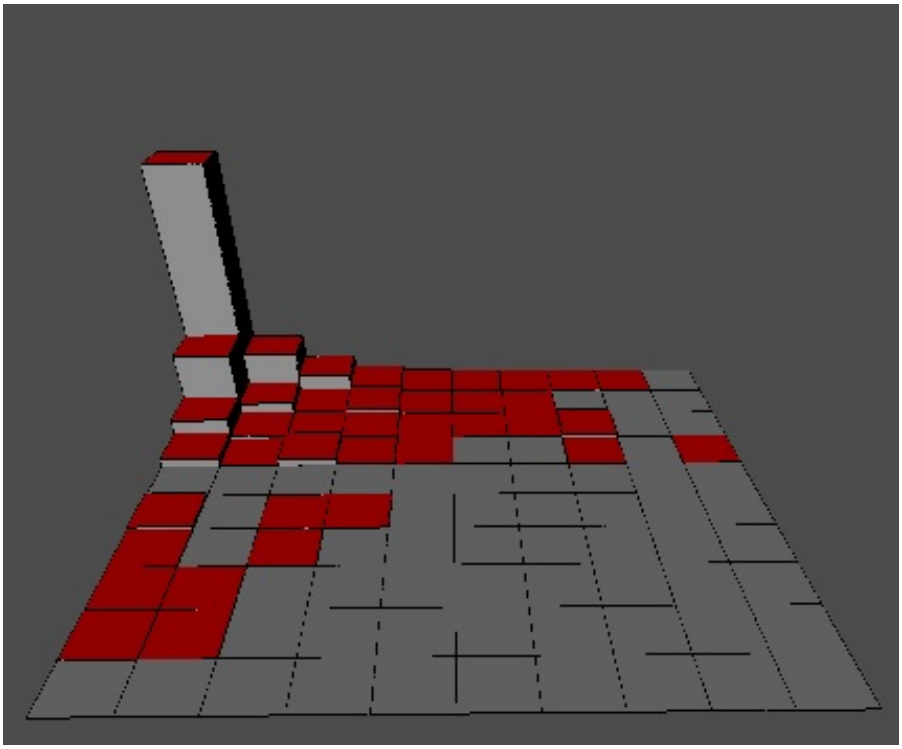

Figure 3.1.2: 3D plot of positive and negative sentiments.

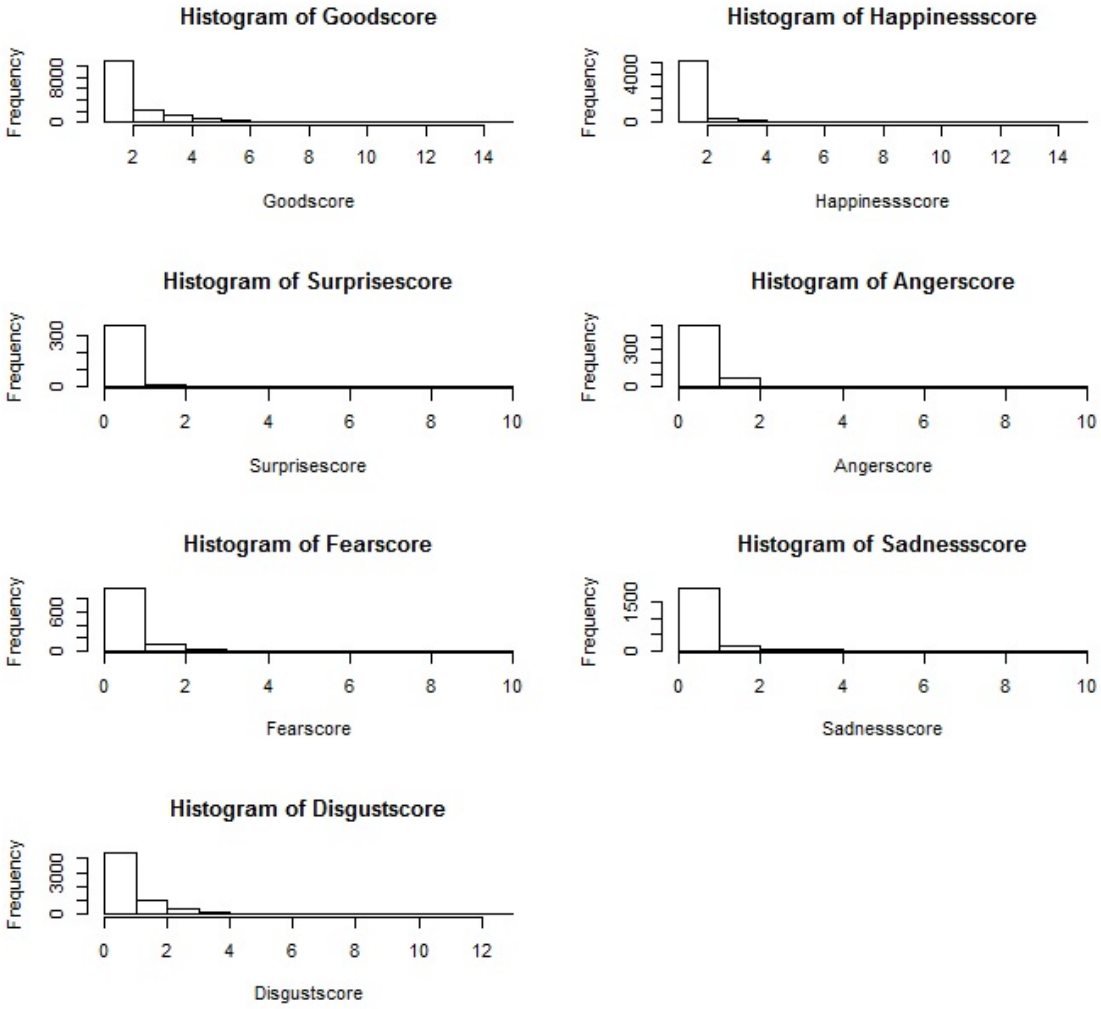

Figure 3.1.3: Histograms for seven sentiment dimensions.

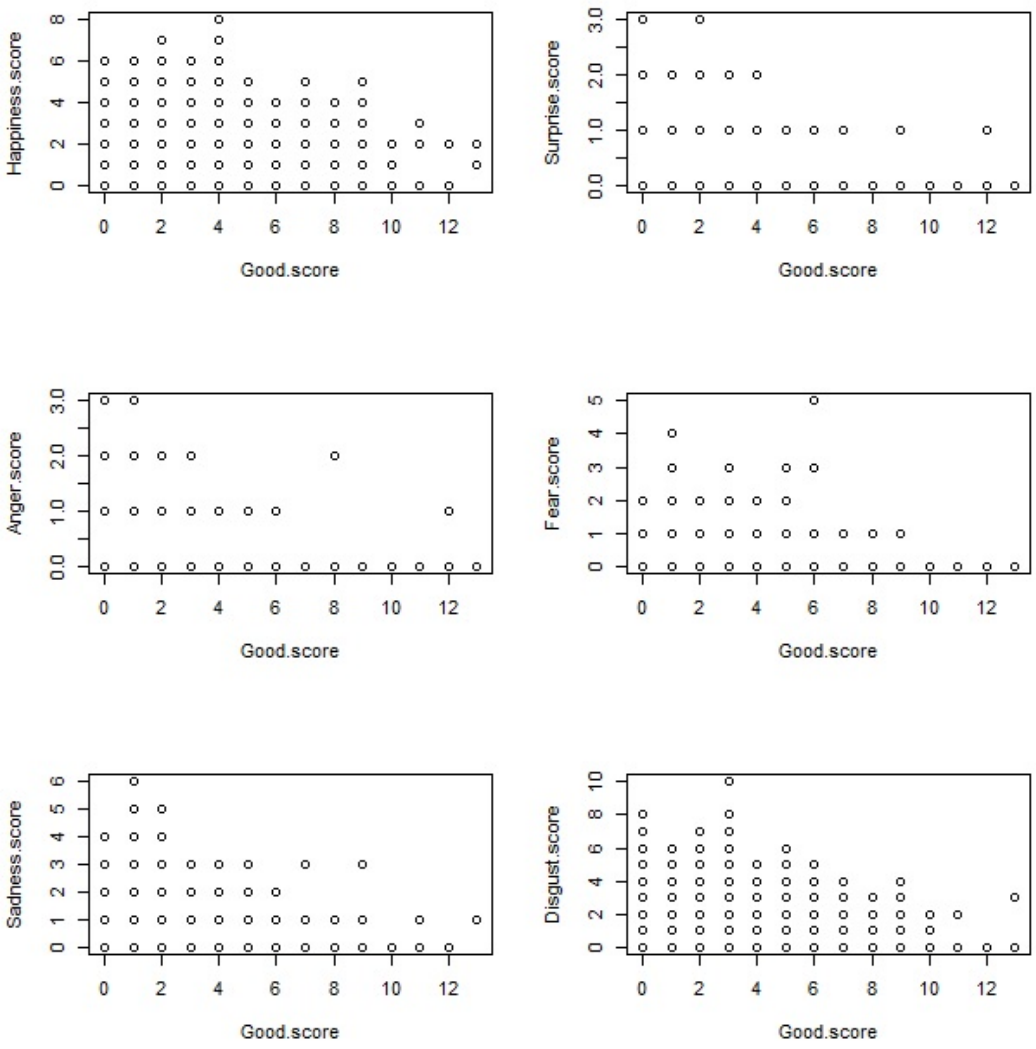

Figure 3.1.4: Plots of Good score vs all other scores.

3.2 Time Series of Sentiment

Our sentiment time series modelling will focus on general orientation analysis for positive and negative sentiments. Posts about Vanke, one of the seven companies which are listed in Tables 2.1 and 2.2, are our data source for further sentiment time series analysis. Daily sentiment scores are calculated by adding up all the sentiment scores for each day. Figure 3.2.1 shows the sentiment time series of overall, positive, and negative scores. It is based on the scores generated by the dictionary approaches illustrated in Section 3.1. The second graph shows the time series for positive scores; the third graph shows the time series for negative scores; and the top graph is the combined overall sentiment time series, created by simply using positive score minus negative score for each post and then aggregating on a daily basis.

Our underlying dataset for sentiment modelling includes 247,373 posts over 221 days for company Vanke. In the data pre-processing part, it is found that there are two one-day gaps due to data downloading interruption: one is on 26th June (Day 55) and the other is on 22nd July (Day 81). Holt-Winters exponential smoothing (Kalekar, 2004) is applied to fix the gaps. The general equation for exponential smoothed statistics is: $P_{t+m} = \alpha Y_t + (1-\alpha)P_t$, where P_{t+m} is the exponential smoothed or forecast value in period $t+m$ for $m = 1, 2, 3, 4...$, α is a smoothing constant with value between 0 and 1, Y_t is the actual value in time period t, and P_t is the forecast of smoothed value for period t. When α approaches one, the forecasts will be based on mostly recent observations. For our data, we take $m = 1$ for prediction and the parameter α is determined by minimising the squared

prediction error for historical data. Using exponential smoothing, point forecast values could be estimated. The gaps of 26th June and 22nd July are filled in using this Holt-Winters exponential smoothing prediction based on previous data points.

A huge spike in negative sentiment time series between 24th Nov and 26th Nov (Days 206 to 208) can be also noticed. By figuring out the negative keywords from initial posts, it was found that, during this period, a scandal about tax evasion of Vanke was reported by a programme on the official Chinese TV Channel. Although the CEO of Vanke denied it and explained it later on, the effect of the news is still very huge. There will be two ways to deal with this huge spike, which will be introduced when modelling negative sentiments in Section 3.3.3.

We denote the scores at time t as:
Pos_t - Positive Sentiment Score
Neg_t - Negative Sentiment Score
$\text{Overall}_t = \text{Pos}_t - \text{Neg}_t$ - Overall Sentiment Score

Accordingly, we define and calculate Proportional Positive, Proportional Negative and Proportional Overall indexes at time t:

$$\begin{aligned} \text{PPos}_t &= \frac{\text{Pos}_t}{\text{PA}_t} \\ \text{PNeg}_t &= \frac{\text{Neg}_t}{\text{PA}_t} \\ \text{POverall}_t &= \frac{\text{Overall}_t}{\text{PA}_t} \end{aligned} \tag{3.1}$$

where PA_t denotes daily Post Amount.

The daily proportional positive, negative and overall indexes can be explained as

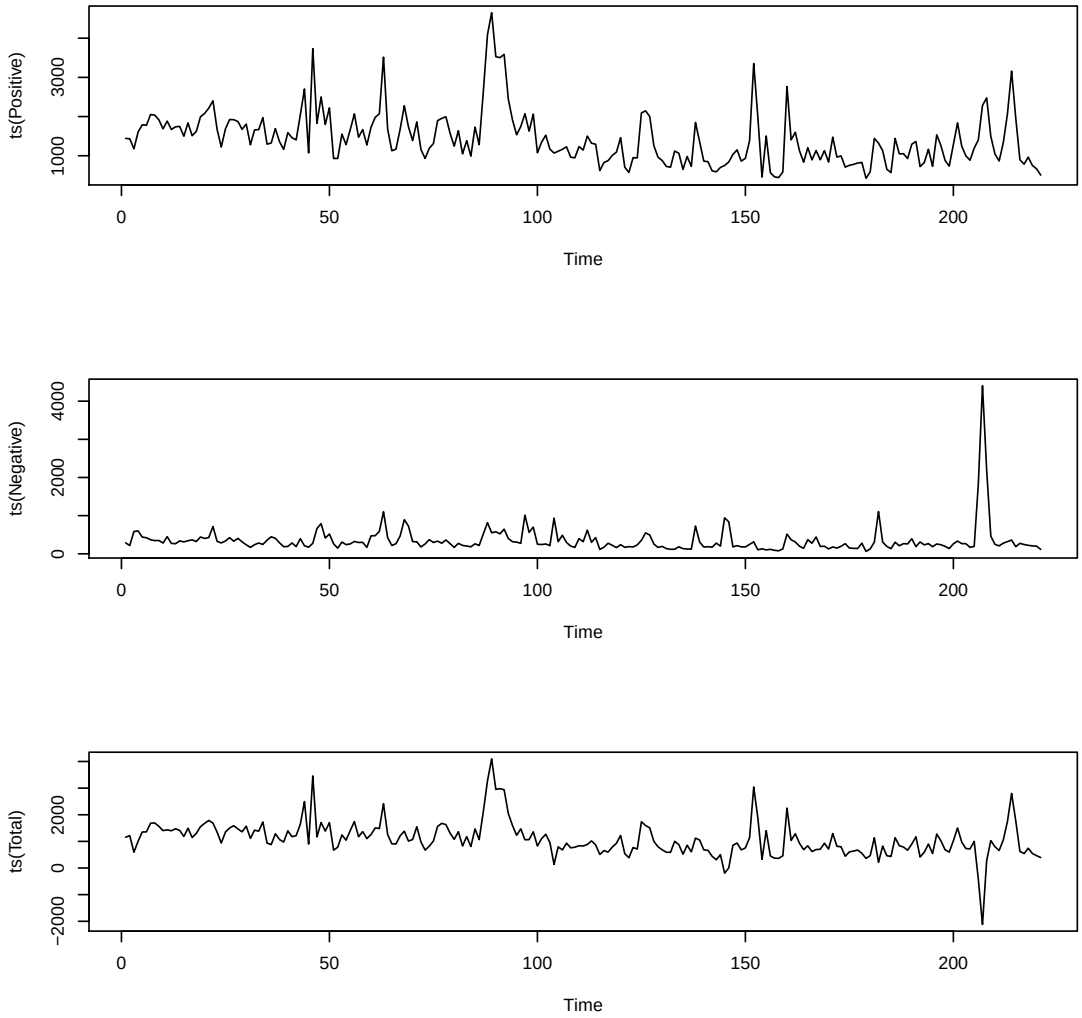

Figure 3.2.1: Original time series for sentiments (top to bottom: Positive, Negative, Overall).

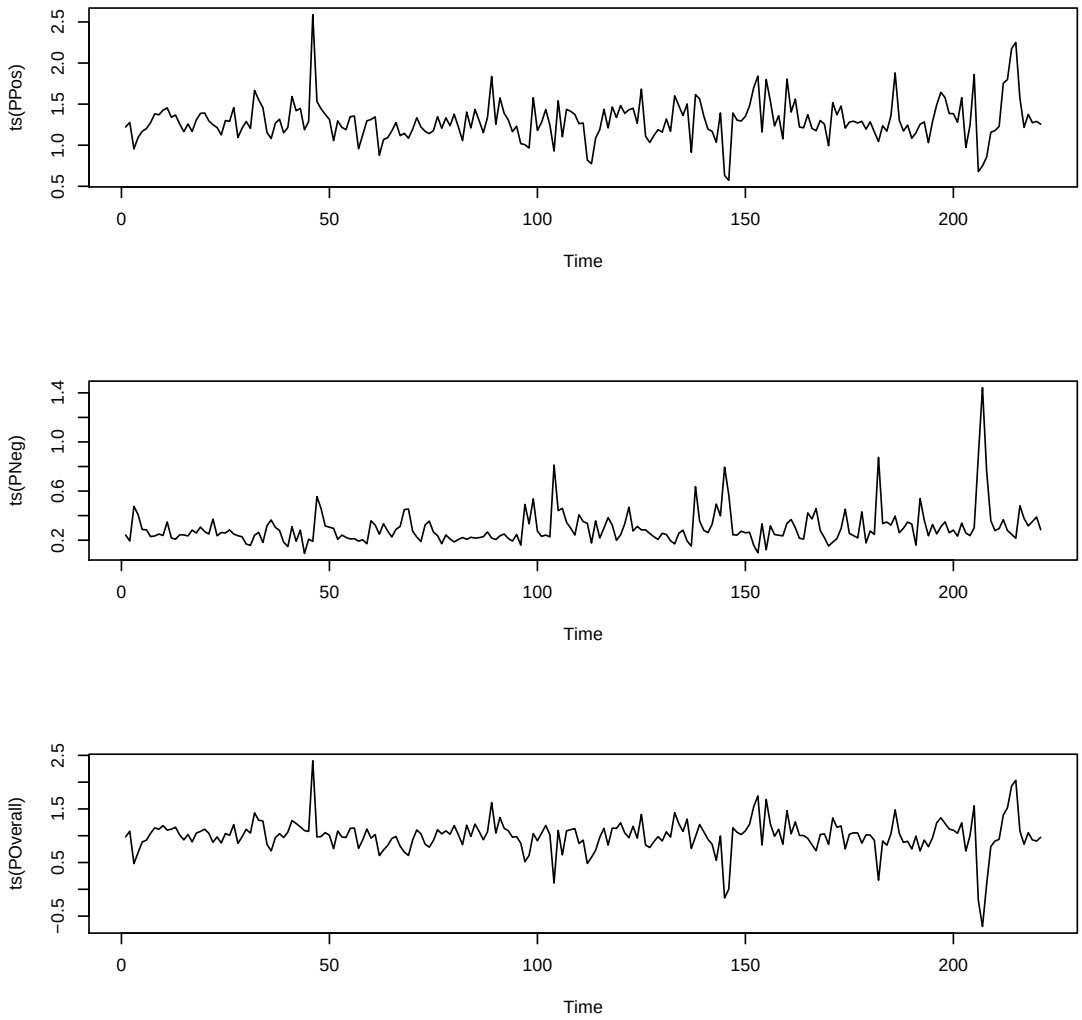

Figure 3.2.2: Proportional time series for sentiments (top to bottom: Positive, Negative, Overall).

the average number of positive or negative words per post for each day. They are not percentages, as they can be larger than one when on average there is more than one positive word in each post. This proportioning can be viewed as a standardising method to describe the daily sentiment polarity, controlling and excluding the effect of general daily post amount (see Figure 3.2.2). Because the magnitudes of positive and negative sentiment time series are diverse, the overall sentiment time series mostly captures the pattern of the positive one, which can be seen in Figure 3.2.1. As a result, we will focus on modelling positive and negative sentiment respectively, rather than building and fitting models for the overall sentiment time series.

We also conduct the Stationarity tests on proportional sentiment time series, and the results in Table 3.2 for all the tests (KPSS, ADF and PP, same as the tests in Table 2.3) show that all the time series are approximately stationary.

Table 3.2: Proportional sentiment time series stationarity test results

Series	KPSS test	ADF test	PP test	Stationarity
PPos_t	0.06976	0.01	0.01	stationary
PNeg_t	0.02375	0.01	0.01	stationary [1]
POverall_t	0.1	0.01	0.01	stationary

In the next two sections, we will examine various attributes of proportional positive and negative sentiment time series in detail and also consider multivariate time series models in order to determine which fits best.

[1] Although the KPSS hypothesis for the trend-stationary is rejected, as the ADF and PP test suggest that PNeg_t is stationary, we still classify it as stationary.

3.3 Univariate Time Series Model Fitting

3.3.1 A General Framework

In this section, we intend to provide a general framework for fitting univariate time series model for real data. The framework incorporates the Box-Jenkins method (Box et al., 2015) to obtain the best fit for the time series' conditional mean, and then examine residuals to determine whether to include a model for conditional variance. The framework will be introduced step by step with instructions, together with a brief literature review and relevant tests. After demonstrating the general framework, applications on both proportional positive and negative sentiment time series will be illustrated thoroughly.

• **Model Identification and Estimation**

Model identification methods, which will be presented in this section, are the rough procedures applied to a set of data to discover the representational models which are worthy of further investigation. The obtained tentative model acts as a starting point for applying further estimation methods.

Stationarity is widely regarded as the initial consideration for time series model identification and fitting. A time series process is stationary if the joint probability distribution of the stochastic process does not change over time. When the assumption of stationarity is satisfied, the statistical properties of the series would be the same in the future as they have been in the past. In this way, statistics such as means, variances and correlations become useful descriptors of future behaviour.

It is essential to take a careful look at the time series plot over time before performing any tests or fitting. If a time series contains an apparent trend or shows clear seasonality, it is possible to observe these obvious non-stationarities visually, and a further stationary test should be able to verify them. From the original time series plot, we can also have a rough idea of distinct outliers or spikes and start to find various way to deal with them. An effective fitting of time series models requires at least a moderately long series. Chatfield (2004) recommends at least 50 observations; while more than 100 observations are recommended by many other researchers.

As introduced in Section 2.4, the KPSS test, the ADF test and the PP test can be adopted for testing stationarity. If the p-value from the KPSS test is larger than 0.05, we treat the process as stationary; while for the ADF test and the PP test, p-values smaller than 0.05 indicate stationarity. If the tests showed the process is non-stationary, there are several common methods to stationarise a time series: taking differences, de-trending, transforming, and adjusting for seasonality. We will not go into a detailed description of the methods, as they can be found in a standard time series text book, e.g. *Introduction to time series and forecasting* (Brockwell and Davis, 2006) and *Time series analysis: forecasting and control* (Box et al., 2015).

Autoregressive moving average (ARMA) models are arguably the most popular time series models used in applied science (Brockwell and Davis, 2006). ARMA models are fitted to understand better the serial correlation and partial serial correlation of the time series process and can be applied for predicting future points.

An ARMA (m, n) process X_t is defined as

$$\begin{aligned} X_t &= \mu + \sum_{p=1}^{m} \alpha_p X_{t-p} + \sum_{q=1}^{n} \beta_q \epsilon_{t-q} + \epsilon_t \\ &= \mu + a(\mathcal{B}) X_t + b(\mathcal{B}) \epsilon_t \end{aligned} \tag{3.2}$$

with mean μ, autoregressive coefficients α_i and moving average coefficients β_j. It can be expressed using back shift operator $\mathcal{B}$, and functions $a(\mathcal{B})$ and $b(\mathcal{B})$ are polynomials of degree m and n.

After confirming the stationarity, we use graphical methods to observe the serial correlations using the auto-correlation function (ACF) and the partial auto-correlation function (PACF). The ACF of a time series illustrates the correlation between values at different times. If the time series is a stationary process, the auto-covariance function of the process depends only on the time difference between x_t and x_{t-p} as a function of the time lag p.

The PACF for a zero mean stationary process x_t is defined as:

$$\begin{aligned} &\pi(1) = \text{corr}(x_2, x_1) \\ &\pi(2) = \text{corr}(x_3 - E(x_3|x_2), x_1 - E(x_1|x_2)) \\ &\pi(3) = \text{corr}(x_4 - E(x_4|x_3, x_2), x_1 - E(x_1|x_3, x_2)) \\ &\text{etc...} \end{aligned} \tag{3.3}$$

Here, $E(x_4|x_3, x_2)$ represents the part of x_4 that is linearly explained by x_3, x_2. Thus, $x_4 - E(x_4|x_3, x_2)$ is the part of x_4 that is unexplained by x_2,x_3. The PACF measures the dependence between x_t and x_{t-p} after the effect of the intervening

values has been removed.

By looking at the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots, one can tentatively identify the orders of AR and MA terms that are required. The cut-off showed by ACF and PACF plots can be used to determine initially the orders of the model. The autocorrelation function of the MA(q) process cuts off after q lags, but the partial autocorrelation function tails off gradually. The autocorrelation function tails off for the AR(p), while its partial autocorrelation function has a cut-off after lag p. If both the autocorrelations and partial autocorrelations tail off, an ARMA process is suggested.

The next step is using computation algorithms to estimate the coefficients which fit the selected ARMA model best. The most common methods use maximum likelihood estimation or least-squares estimation. We use R packages at this stage to estimate the parameters and check the significance. The default estimation is to use the conditional-sum-of-squares to find starting values and then apply the maximum likelihood estimation (MLE). The estimation process is described in Box et al. (2015).

- **Model Diagnostic Checking**

After fitting the model, obtaining all the estimated coefficients and ensuring they are significant, we move to the next procedure: model diagnostic checking. The main objective is to examine whether the fitted model conforms to the specifications of a stationary univariate process and to check the adequacy of the selected model. Under the assumption of stationarity, the residuals should be independent of each other and constant in mean and variance over time.

Plotting the residual time series to check visually whether there are obvious patterns on the plot of standardised residuals and whether the mean and variance of residuals stay constant over time are usually the first step. The ACF and PACF of the residuals should be investigated: if the model is adequate, the autocorrelations and partial autocorrelations between different lags of residuals should not be statistically significant.

The Box-Pierce Q-statistic (Box and Pierce, 1970) or Ljung-Box (Ljung and Box, 1978) test can be conducted to formally assess autocorrelation and examine the independence of the residuals. It provides the joint hypothesis that the first k autocorrelations of the adjusted error terms are jointly zero. The small p-value of the Box test indicates the rejection of the null hypothesis that the residuals are independently distributed.

The BDS test (Caporale et al., 2005) can further detect non-linear serial dependence in residual time series with the hypothesis that the remaining residuals are a series of independent and identically distributed (i.i.d) random variables. Rejection of the i.i.d. hypothesis implies that there is remaining structure in the time series, which could include a hidden nonlinearity, hidden nonstationarity or other type of structure missed by model fitting.

The error terms are generally assumed to be i.i.d. variables sampled from a normal distribution with zero mean. If the assumption of normality is made, the model would be fitted using maximum likelihood estimation of Gaussian random variables. Thus, if the underlying Gaussian assumption does not hold, the likelihood function can be messed up and the MLE becomes unreliable. As a result, we attempt to test the normality of the residual series using graphical and testing procedures.

The histogram of residuals and the QQ plot against normal can provide an initial observation of normality. The small p-value from the Shapiro-Wilk test (Royston, 1982) indicates the rejection of hypothesised normality.

If the fitting is appropriate for the data set, the part of the data unexplained by the model, which are the residuals, should be small and no systematic or predictable patterns remain in the residuals. However, the model diagnostic checking is not always satisfactory. If the estimation is inadequate, we have to return to step one and attempt to build a better model.

If there is a serial dependence in residuals series, one may try higher order fitting for ARMA. If the normality test is rejected and there are some patterns in the QQ plot such as S shape or heavy-tailed shape, one may attempt alternative transformations on the original time series or use different distributions for the errors. R package *rugarch* (Ghalanos, 2012) provides various specifications for a broad class of distributions included those with skew and heavy tails. The range includes normal distribution, Student-t distribution, Generalized Error distribution, and their skew variants, which can be chosen for the residuals.

If the original time series or residual time series is observed to fluctuate around a constant level but exhibits volatility clustering, one should inspect the ACF and PACF of squared residual series. Volatility clustering means both the large and small changes in the returns tend to cluster together. The significance of ACF or PACF for squared residuals can be a sign for conditional heteroskedasticity, which is usually modelled by the autoregressive conditional heteroskedasticity (ARCH) model.

To further confirm this ARCH effect, we conduct the ARCH Lagrange Multiplier (LM) test with the null hypothesis that there is no ARCH effect. The rejection indicates the possibility of adding ARCH effect in conditional variance to the original model with ARMA conditional mean.

The goodness-of-fit of the model can be assessed with the information criteria, which generally embody two factors: one is a function of the log likelihood and the other factor penalizes for the loss of degrees of freedom from adding extra parameters. The objective is to choose the number of parameters, which minimizes the value of the information criteria. The most common information criteria are Akaike information criterion (AIC) and Bayesian information criterion (BIC). The AIC is defined to be:

$$\mathrm{AIC}(k) = 2k - 2\log(L) \tag{3.4}$$

and BIC is defined to be:

$$\mathrm{BIC}(k) = k\log(n) - 2\log(L) \tag{3.5}$$

where L is the maximum value of the likelihood function for the model, k is the total number of parameters estimated, and n is the number of observed data points.

• ARMA Conditional Mean with GARCH Type Conditional Variance

In order to include conditional heteroscedastic effects and the patterns of fat tails as clustering of volatilities, which are typical in a set of time series processes, the ARCH model was introduced by Engle (1982) and generalised to the GARCH model by Bollerslev (1986). After eliminating serial correlation in conditional mean

using ARMA, we make use of GARCH type model for conditional variance.

The GARCH (p, q) conditional variance is defined as:

$$\begin{aligned} \epsilon_t &= z_t \sigma_t \\ z_t &\sim \mathcal{D}_\vartheta(0,1) \\ \sigma_t^2 &= \omega + \sum_{i=1}^{p} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 \\ &= \omega + \alpha(\mathcal{B})\epsilon_{t-1}^2 + \beta(\mathcal{B})\sigma_{t-j}^2 \end{aligned} \tag{3.6}$$

where the $\mathcal{D}_\vartheta(0,1)$ is the probability density function of the i.i.d. innovations with zero mean and unit variance. This probability density function is an extended form of $\mathcal{N}(0,1)$, which can be changed to other distributions and includes an additional distributional parameter ϑ to describe the skew and the shape of the distribution. Thus, this specification provides many alternative conditional distributions and their skewed versions other than Gaussian.

Models with ARMA specification in the conditional mean and GARCH-type conditional variance were developed by Wurtz et al. (2006). This enables us to model the residuals as a GARCH process with different specifications in innovations, because the residuals sometimes appear to be non-Gaussian and somewhat heteroscedastic. Therefore, the time series X_t is defined using the following process:

$$X_t = \mathrm{E}[X_t|\Omega_{t-1}] + \epsilon_t \tag{3.7}$$

where $\mathrm{E}[\cdot|\cdot]$ denotes the conditional expectation operator, Ω_{t-1} is the information set at time $t-1$, and ϵ_t indicates disturbances with zero mean and acts as the

unpredictable part of the time series. We can split the model into two parts: mean equation as an ARMA process from Formula (3.2) and residual equations as a GARCH process from Formula (3.6).

3.3.2 Fitting Proportional Positive Sentiment

The original proportional positive sentiment time series plot is presented in Figure 3.3.1. The mean of this series appears to keep unchanged over time, but the series seems to contain heteroscedasticity. The series seems to also contain a few large spikes, which may result in heavy tails in the distribution. Several tests are conducted to further investigate the series.

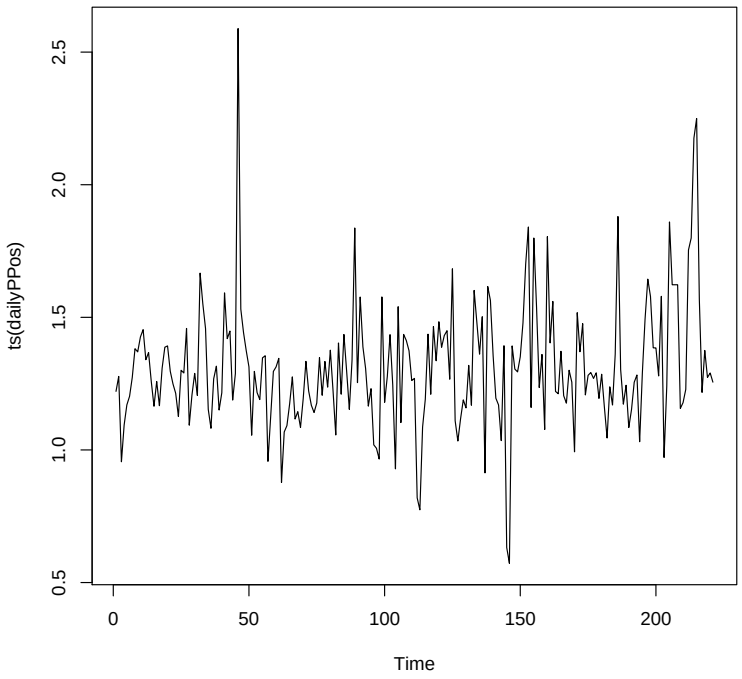

Figure 3.3.1: Time series plot for proportional positive sentiment.

Figure 3.3.2 shows the ACF and PACF of the proportional positive sentiment time series. In the plot of ACF, the first two lags are above the line. For PACF, only the first lag is significantly above the line. The cut-offs showed by ACF and PACF plots are used to initially choose the orders: ARMA(1,2) would be the preliminary choice. We first try this ARMA fitting, observe the patterns of the residuals, and thereby check the validity of the ARMA model.

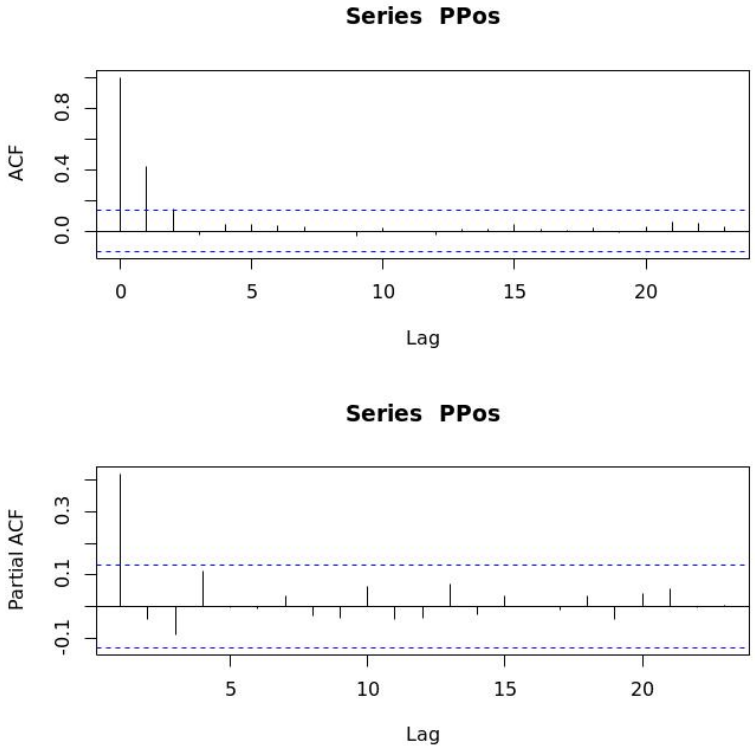

Figure 3.3.2: ACF and PACF plots for proportional positive sentiment time series.

The fitted result of ARMA(1,2) shows that the p-values of both first and second moving-average components are much higher than 0.05, but the autoregressive component appears to be highly significant. Thus, we try to fit the model with only an autoregressive part: AR(1). The residual plots in Figure 3.3.3 include the

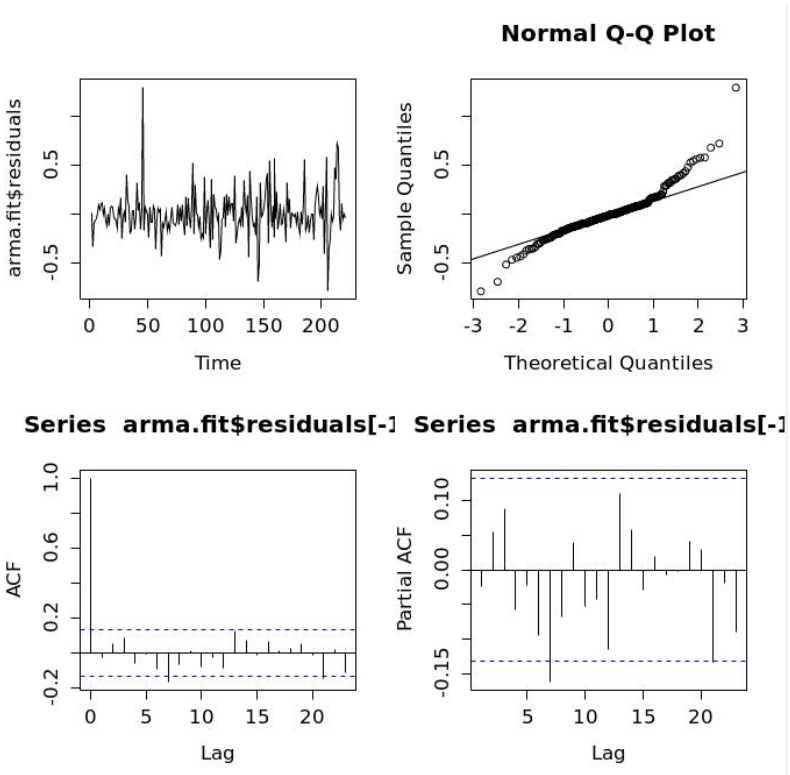

Figure 3.3.3: Residual plots of AR(1) for proportional positive sentiment time series.

time series plot for residuals, the QQ plot of residuals against normal, and the ACF PACF plots for residuals. We can still see some patterns for the innovations in the residual time series, and the normal QQ plot is not fitted very well. There is no very large auto-covariance that can be observed from both the ACF and PACF chart, although a small negative spike appears around lag seven.

Even though the p-value of the Box test was quite large (0.7211), which indicates we cannot reject the hypothesis that the residuals are independently distributed, the p-values of the BDS test are quite small, which suggests the residuals are not i.i.d. random variables. Furthermore, the extremely small p-value in the Shapiro-Wilk test (Royston, 1982) shows the rejection of the hypothesis of

normality. In conclusion, residual diagnosis indicates that the AR(1) model fitting is not appropriate for the data set.

To understand the time series better, several trials have been made. The result shows that the fitting of ARIMA model including the integration part is no better than AR(1). Log transformation on the original data has been applied, but the results remain unsatisfactory. Model covariances for different periods of time are calculated using different window sizes or using exponentially weighted methods, which show some degree of time-varying.

As the residuals from the pure ARMA model appear to be non-Gaussian and with some heteroscedasticity, we consider modelling the residuals as a GARCH process with possibly different distributions. Models with ARMA specification in the conditional mean and GARCH-type conditional variance described previously are applied rather than pure ARMA process.

The model ARMA(1,0) with GARCH(1,1) is fitted and estimated using the R package. β is not statistically significant, so we get rid of generalised part of the GARCH model, and the final model with estimated parameters would be:

$$\begin{aligned} x_t &= 1.29 + 0.29x_{t-1} + \epsilon_t \\ \epsilon_t &= z_t\sigma_t \\ z_t &\sim \mathcal{D}_\vartheta(0,1) \\ \sigma_t^2 &= 0.041 + 0.24\epsilon_{t-1}^2. \end{aligned} \tag{3.8}$$

Residual diagnostics provide tests for normality, independence and ARCH effects. The p-values for both the Jarque-Bera test and the Shapiro-Wilk test are quite low, which indicates that the hypothesis of samples that come from a normal distribution is rejected. The Ljung-Box test demonstrates that the series is independently distributed, and the ARCH LM test shows there is no ARCH effect in the residuals. The residual plots and QQ Plot against normal (Figure 3.3.4) also indicate that it is not reasonable to assume the innovations are normally distributed.

The QQ Plot is fitted much better in Figure 3.3.5 when the conditional distribution $\mathcal{D}_\vartheta(0,1)$ changes to Student-t distribution. This change includes an extra estimate for the shape parameter v, and the final model becomes:

$$\begin{aligned} x_t &= 1.29 + 0.32x_{t-1} + \epsilon_t \\ \epsilon_t &= z_t\sigma_t \\ z_t &\sim \mathcal{D}_\vartheta(0,1) \\ \sigma_t^2 &= 0.039 + 0.64\epsilon_{t-1}^2 \\ \vartheta &= 3.02 \end{aligned} \tag{3.9}$$

Our proportional positive sentiment data fit the general framework well, and we employ AR(1) for conditional mean and ARCH(1) for conditional variance with Student-t distribution. The volatility clustering and heavy-tailed properties are satisfactorily explained using ARCH and Student-t distribution.

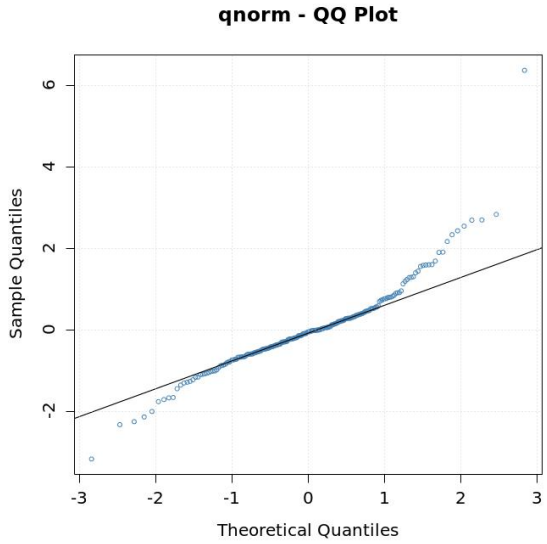

Figure 3.3.4: QQ plot against normal of AR(1) + ARCH(1) for proportional positive sentiment time series.

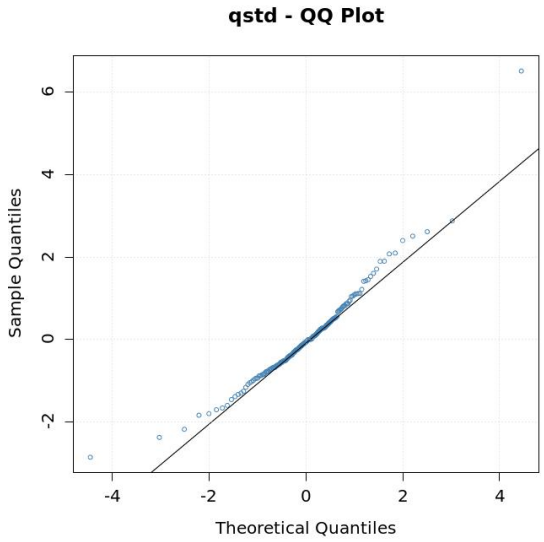

Figure 3.3.5: QQ plot against t-distribution of AR(1) + ARCH(1) for proportional positive sentiment time series.

3.3.3 Fitting Proportional Negative Sentiment

Model fitting for proportional negative sentiment is more complicated than for the positive one, because there are many more spikes and the distribution is more heavy-tailed for the negative sentiment. After looking into the spikes, it becomes clear that most large ones occurred occasionally and unpredictably because of influential negative news, and negative news usually has bigger effects than good news. Due to the spikes and heavy-tails, the time series for proportional negative sentiment is not strictly stationary as indicated in Table 3.2. As the main objective for us is to model the general movement of sentiment, we will not analyse these spikes in depth, but investigating these huge spikes might be an interesting task. In our analysis, we adjust the biggest spike in two ways and conduct further analyses based on the adjusted time series. In this Section 3.3.3, we use the exponential smoothing to smooth the huge spike; while in the next Section 3.3.4, we choose to employ the threshold clearance to preserve the spike to a certain extent.

In this subsection, the huge spike between 24th Nov and 26th Nov in time series is adjusted using the Holt-Winters exponential smoothing method introduced in the last section for one-day gaps. The original time series and the adjusted one are shown in Figures 3.3.6 and 3.3.7.

The ACF plot and PACF plot are observed first in Figure 3.3.8. The first and second order of ACF are around the line and the first order PACF is above the line: we can start from ARMA(1,2). However, an interesting finding is that both the ninth order ACF and PACF are beyond the line, which may indicate that a higher order ARMA conditional mean might be also considered.

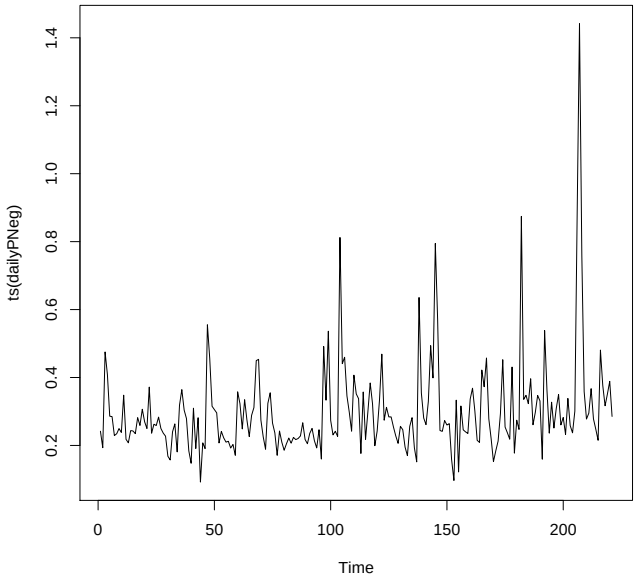

Figure 3.3.6: Time series plot for proportional negative sentiment.

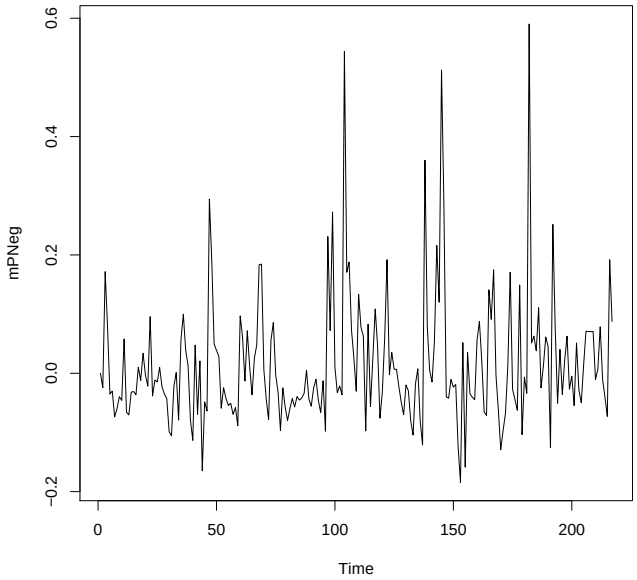

Figure 3.3.7: Time series plot for proportional negative sentiment (adjusting the huge spike between Day 206 and Day 208 by exponential smoothing).

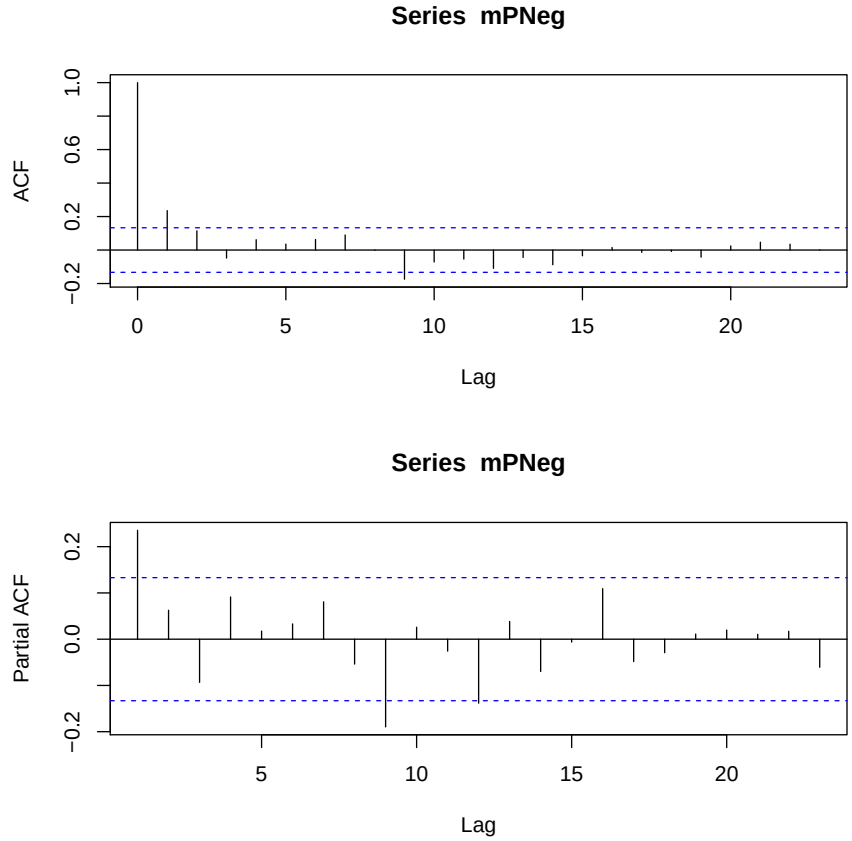

Figure 3.3.8: ACF and PACF plots for proportional positive sentiment time series (adjusting the huge spike between Day 206 and Day 208 by exponential smoothing).

Due to probable heteroscedastic and heavy-tailed residuals, we began to fit models using the general framework with ARMA specification in the conditional mean, GARCH-type conditional variance and normal distributed innovations. Nevertheless, the residual diagnostics seem to be inadequate. Furthermore, the α in the ARCH specification is generally with a p-value larger than 0.05, which means the ARCH effect might not be significant. Models more complex than standard ARMA mean and standard GARCH variance with normal distribution should be considered.

R package *rugarch* provides different settings for conditional mean, various extensions for GARCH variance specification and different types of distributions for innovations. Furthermore, it provides advanced model fittings for a possible ARCH-in-mean effect or to include external regressors. Details for the extensions can be found in Appendix B.3.1.

There are numerous choices for model specification and fitting: initially, we select the order of ARMA(m,n) and GARCH(p,q), choose the specific GARCH extension and conditional distribution; there are various other options such as whether or not to include ARCH-in-mean, add external regressors, or fix certain parameters. After a large number of trial-and-errors, two possible models stand out, but there are also weaknesses for each of them.

• **Model 1: ARMA(4,2) conditional mean and GARCH(1,1) conditional variance with skewed normal distribution**

A model containing conditional mean with ARMA(4,2), conditional variance with GARCH(1,1) and skewed normal distribution seems to be a feasible choice among all the standard GARCH specifications. A skewed normal distribution fits the innovations much better than the standard normal distribution. The result is generally robust, as the p-values of robust standard error for the parameters from quasi-maximum likelihood estimation (White, 1982) remain smaller than 0.05. The

estimated **Model 1** is:

$$\begin{aligned} x_t &= 0.039 - 0.82x_{t-1} - 0.61x_{t-2} + 0.31x_{t-3} + 0.17x_{t-4} \\ &\quad + 1.05\epsilon_{t-1} + 1.08\epsilon_{t-2} + \epsilon_t \\ \epsilon_t &= z_t\sigma_t \\ z_t &\sim \mathcal{SN}_\xi(0,1) \\ \sigma_t^2 &= 0.001 + 0.085\epsilon_{t-1}^2 + 0.79\sigma_{t-1}^2 \\ \xi &= 1.99 \end{aligned} \tag{3.10}$$

ξ denotes the rate of skewness (Ghalanos, 2013), and the related literature for univariate skewed distributions can be found in Fernández and Steel (1998) and Ferreira and Steel (2012).

The diagnostics plots are shown in Figure 3.3.9. The first plot for the series with 2 conditional standard deviation superimposed seems to fit the pattern well. The QQ plot fits well for smaller quartiles, but not for large ones due to the extremely large values in the time series. The information criteria AIC and BIC calculated using the Formula (3.4) and (3.5) for this model are around -1.9.

• **Model 2: ARMA(2,1) conditional mean and apARCH(1,1) conditional variance with skewed normal distribution**

After observing the asymmetric and heavy-tailed properties of our time series, we examine other alternative GARCH extensions, and in the end a model with ARMA conditional mean, Asymmetric Power ARCH (apARCH) conditional variance, and skewed normal distribution shows a good fit.

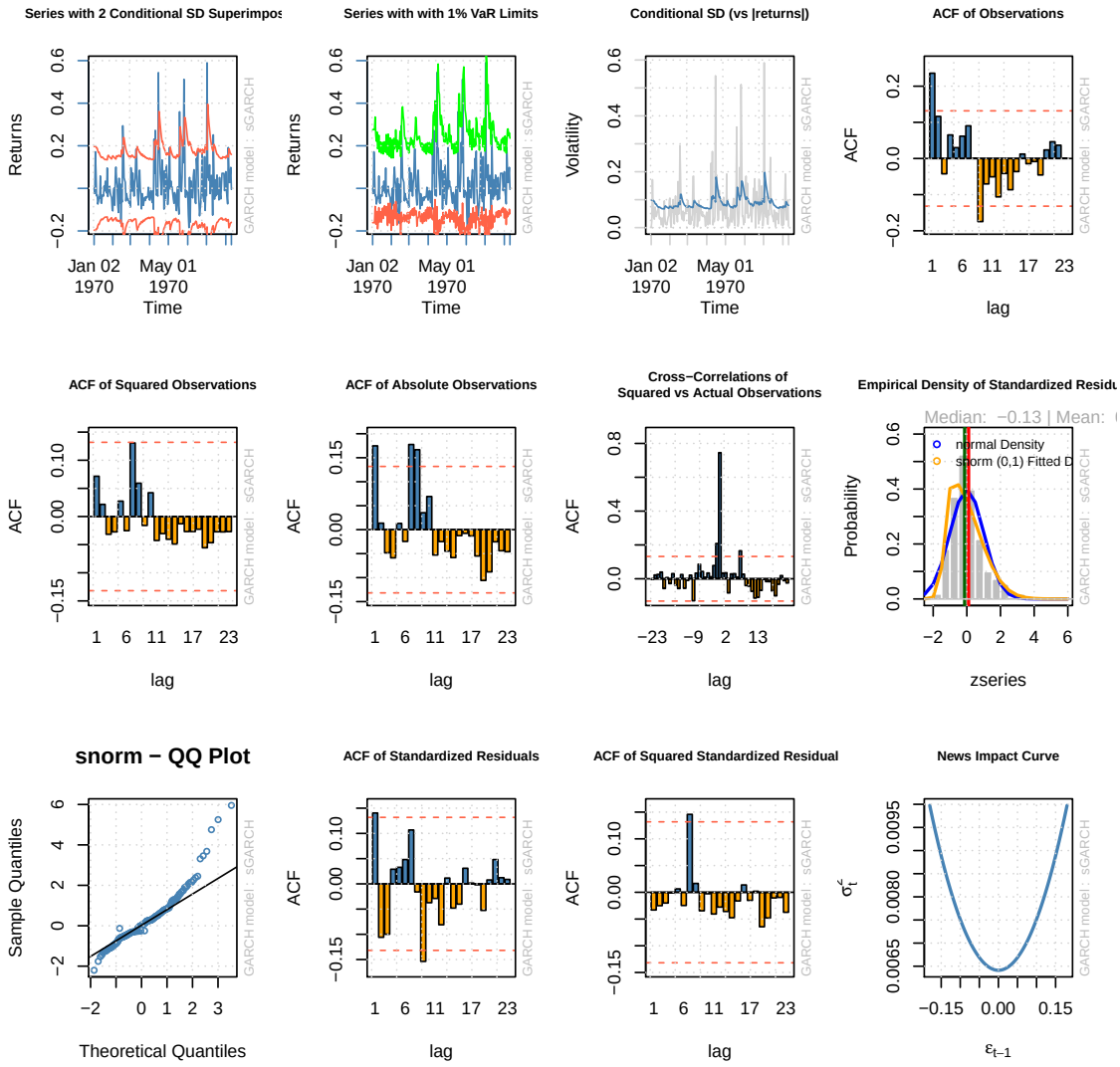

Figure 3.3.9: Diagnostics plots of Model 1 for proportional negative sentiment time series.

Ding et al. (1993) invented the Asymmetric Power ARCH model by adding leverage (γ) and the Taylor effect (δ). The leverage effect (Black, 1976) models the phenomenon that the positive and negative information leads to a different level of effect to volatility, and the current value and future volatility have a negative correlation. The Taylor effect (Taylor, 1986) describes the difference in the sample

autocorrelations of absolute and squared returns. The complete model can be expressed as follows:

$$\sigma_t^\delta = (\omega + \sum_{j=1}^{m} \zeta_j v_{jt}) + \sum_{j=1}^{q} \alpha_j (|\epsilon_{t-j}| - \gamma_j \epsilon_{t-j})^\delta + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^\delta \tag{3.11}$$

Here, v_{jt} is for external regressors, which are not included in our model, $\delta \in \mathbb{R}^+$ is a Box-cox transformation of σ_t , and γ_j is the coefficient for the leverage effect.

It can be found that all the parameter estimations become significant for the model with ARMA(2,1) and apARCH(1,1) with skewed normal distribution. The ξ for skewness of normal, δ for the Taylor effect, and γ for leverage effect are also significant. The estimated **Model 2** is:

$$\begin{aligned} x_t &= 0.016 - 0.603 x_{t-1} + 0.29 x_{t-2} + 0.81 \epsilon_{t-1} + \epsilon_t \\ \epsilon_t &= z_t \sigma_t \\ z_t &\sim \mathcal{SN}_\xi(0,1) \\ \sigma_t^{2.04} &= 0.19(|\epsilon_{t-1}| - 0.48\epsilon_{t-1})^{2.04} + 0.80\sigma_{t-1}^{2.04} \\ \xi &= 2.45 \end{aligned} \tag{3.12}$$

• Brief Comparison Between Model 1 and Model 2

The values of both AIC and BIC calculated using the Formula (3.4) and (3.5) for Model 2 become smaller (around -2.0) and slightly better than Model 1. However, the residual diagnostics plot shows that this model might be worse than Model 1. The graphical comparisons between the fittings of Model 1 and Model 2 can be seen in Figures 3.3.10 and 3.3.11. In conclusion, Model 1 fits the spikes well

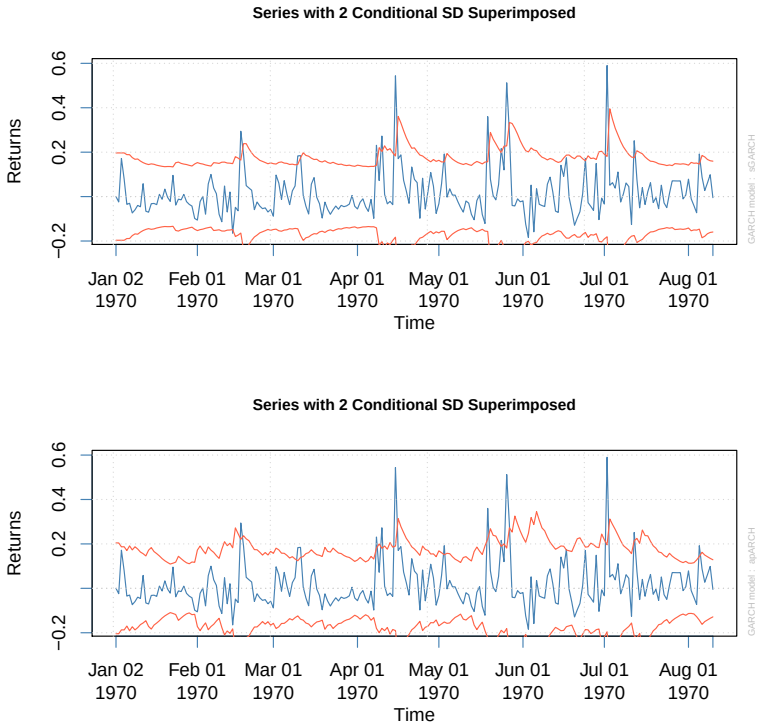

Figure 3.3.10: Plots of series with 2 conditional standard deviation superimposed for proportional negative sentiment time series (Top: Model 1 vs Bottom: Model 2).

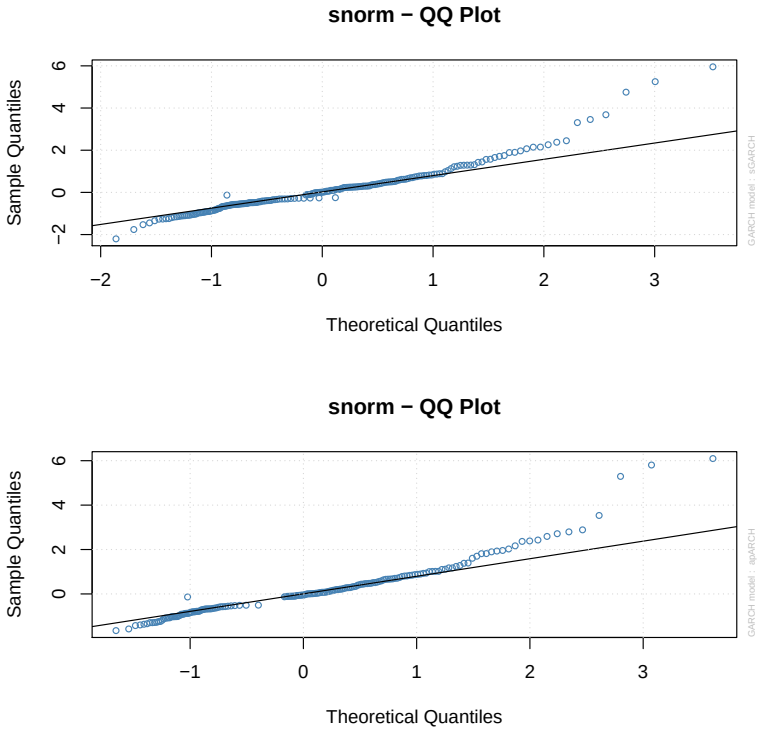

Figure 3.3.11: QQ plots against skewed normal for proportional negative sentiment time series (Top: Model 1 vs Bottom: Model 2).

and the residual fitting is slightly better; Model 2 which includes leverage and Taylor effects in GARCH process has a lower information criterion. However, neither of the models are highly satisfactory. They will be further compared using cross-validation in Section 3.3.5.

3.3.4 Other Approaches for Fitting Proportional Negative Sentiment

In the last section, we attempted to fit the proportional negative time series using a general framework of ARMA conditional mean and GARCH type conditional variance with a range of distributions. Nevertheless, it is still possible that neither the models from this general framework nor their variates are adequate. In this case, we have to turn back to the dataset, undertake more adjustments and consider other possible approaches.

We employed another spike removal method threshold clearance (Boudt et al., 2007), instead of conducting simple exponential smoothing, as this method recognises and keeps those spikes with lower values instead of completely smoothing them out. It robustly cleans a time series to reduce the magnitude, but not the number or direction, of the observations that exceed the $1-\alpha\%$ percent risk threshold.

There are three steps for this threshold clearance. The first is to rank the observations according to their extremeness. Suppose we have the data points $r_1, r_2, ...r_T$, and we denote μ and Σ the mean and covariance matrix of the data. The Mahalanobis distances for each r_t can be defined as $d_t^2 = (r_t - \mu)^T \Sigma^{-1} (r_t - \mu)$. We order the distances from the largest to the smallest for each time point. The second is to identify the outliers. Observations are recognised as outliers if their Mahalanobis distance d_t^2 is larger than the empirical $1-\alpha$ quantile $d_{(1-\alpha)T}^2$ and beyond an extreme quantile of the Chi-square distribution with n degrees of freedom (e.g. $\chi_{n,0.999}^2$ denotes 99.9% quantile). n is the dimension of the series, and we have

$n = 1$ here. The last step would be replacing these outliers identified by step 2 with $r_t\sqrt{\frac{\max(d^2_{(1-\alpha)T}, \chi^2_{n,0.999})}{d_t^2}}$. The cleaned data have the same orientation as the original vector, but the spikes' magnitude becomes smaller (Khan et al., 2007). That is to say, the robust method proposed here does not remove any data points from the series, but only decreases the magnitude of the extreme events.

The adjusted time series with threshold clearance for proportional negative sentiment can be found in Figure 3.3.12. The next two alternative approaches are based on this adjusted time series.

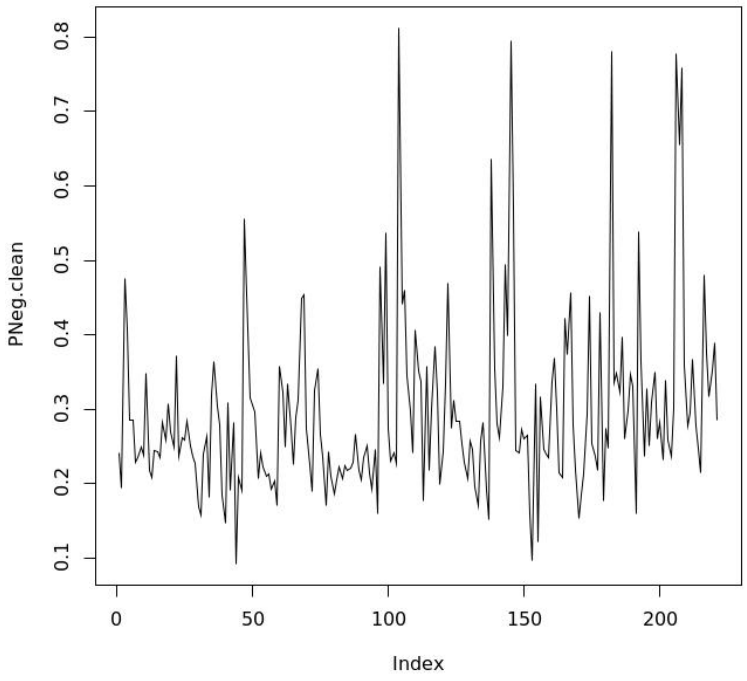

Figure 3.3.12: Time series plot for proportional negative sentiment (adjusting the huge spike between Day 206 and Day 208 by threshold clearance).

• Model A: ARMA Model with Skewed t-distribution

To start with, we check the ACF and PACF plots (see Figure 3.3.13), and the

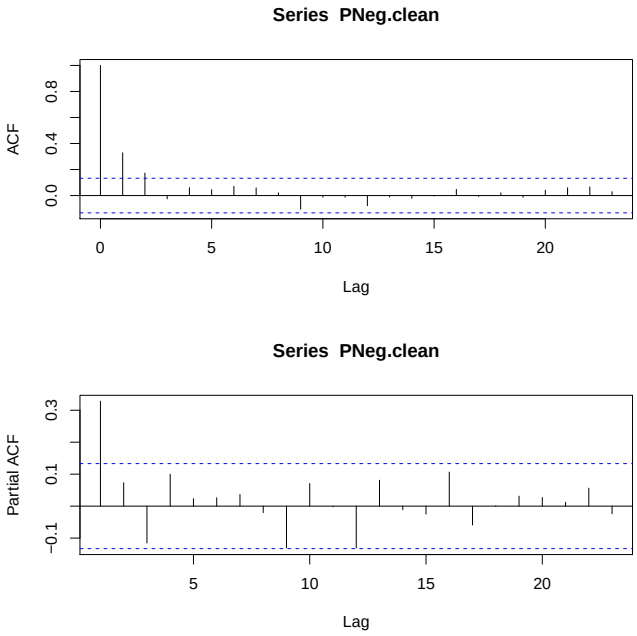

Figure 3.3.13: ACF and PACF plots for proportional negative sentiment (adjusting the huge spike between Day 206 and Day 208 by threshold clearance).

ARMA(1,2) model is fitted based on the patterns. To examine the independence and existence of the ARCH effect, the ACF and the PACF of the residuals and squared residuals are plotted. It can be found that both the ACF and PACF for residuals and squared residuals are not beyond the reference line. It indicates that the residuals are independently distributed and there is no obvious ARCH effect. To further confirm this, the Box test and ARCH LM test are conducted. Both of the tests give large p-values, which verifies the findings. We may interpret disappearance of the ARCH effect as: the volatility clustering phenomenon diminishes due to the removal of spikes using threshold clearance.

The QQ plot against normal for the fitting which contains only ARMA component seems to be not very satisfactory (Figure B.3.1 in Appendix B.3). The Null of

Shapiro-Wilk test is also rejected, which means the residuals are not normally distributed. A model without the ARCH effect in residuals but with skewed t-distribution seems to be a possible choice. The fitted **Model A** is as follows:

$$\begin{aligned} X_t &= 0.19 + 0.73X_{t-1} - 0.47\epsilon_{t-1} - 0.10\epsilon_{t-2} + \epsilon_t \\ \epsilon_t &= z_t \\ z_t &\sim \mathcal{D}_\vartheta \end{aligned} \tag{3.13}$$

The conditional distribution $\mathcal{D}_\vartheta$ is skewed t-distribution with parameter $\vartheta(\xi, \nu)$: skew estimate $\hat{\xi}$ is 1.63 and shape estimate $\hat{\nu}$ is 2.47. The corresponding QQ plot against skewed t-distribution can be found in Figure 3.3.14.

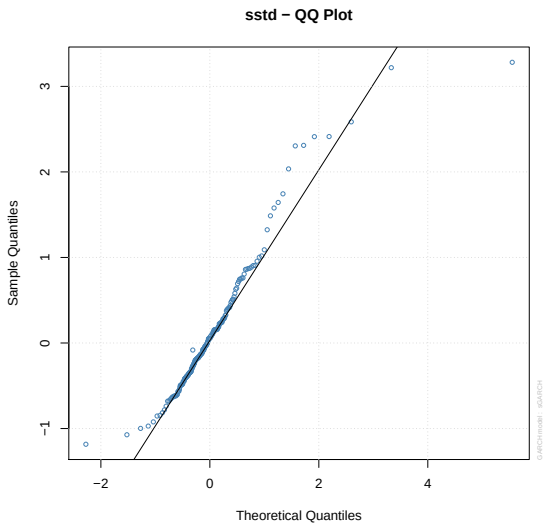

Figure 3.3.14: QQ plot against skewed t-distribution of Model A for proportional negative sentiment (adjusting the huge spike between Day 206 and Day 208 by threshold clearance).

• **Model B: Treating Series as X_t^2 and Fitting GARCH(1,1) Model**

The QQ plot 3.3.14 of the Model A in the last section still contains a considerable fat tail, although we have already considered the skewness and the shape in the conditional distribution. As the data points in our time series are all positive and away from zero, we speculate that it might be applicable to regard the original time series as $X_t^2 + c$ and fit a pure GARCH model.

After two unsatisfactory attempts, we endeavour to code a quasi-maximum likelihood estimation from scratch without any existing packages. We deduct the smallest value from each point for the daily proportional negative sentiment time series, then the smallest value of the series becomes zero. They are treated as X_t^2 and then the parameters of the GARCH(1,1) model can be estimated using maximum likelihood estimation. The model now is

$$\begin{aligned} \widetilde{X_t}^2 &= z_t^2 \sigma_t^2 \\ z_t &\sim \mathcal{N}(0,1) \\ \sigma_t^2 &= \omega + \alpha X_{t-1}^2 + \beta \sigma_{t-1}^2 \end{aligned} \tag{3.14}$$

where $\widetilde{X_t} = X_t - \min_{0 \le i \le T}(X_i)$.

The Quasi-likelihood function for general GARCH process is defined as:

$$\begin{aligned} L_n(u) &= \sum_{1 \le k \le n} -\frac{1}{2}\left\{ \log w_k(u) + \frac{x_k^2}{w_k(u)} \right\} \\ w_k(u) &= c_o(u) + \sum_{1 \le i < \infty} c_i(u) x_{k-i}^2 \\ w_k(u) &= \sigma_k^2. \end{aligned} \tag{3.15}$$

The general steps for maximum likelihood estimation are: initialise the model parameters and bounds for the time series globally, set the conditional distribution function, compose the log-likelihood function, estimate parameters, and compute the Hessian numerically to give the inference.

Our code is tested and examined using both built-in GARCH data from an R package and simulated GARCH data produced by standard GARCH models. It is proved that the results generated by our code provide accurate estimations of the true parameters of the models from these two data sets. Then we apply the code to our own data sets of proportional negative sentiment time series. The potential problem is that the p-values of the estimated parameters are relatively large (around 0.1 to 0.3). The estimated values of ω, α and β can be brought back into Formula (3.14) and we can have explicit expression for our estimated **Model B**:

$$\begin{aligned} \widetilde{X_t}^2 &= z_t^2 \sigma_t^2 \\ z_t &\sim \mathcal{N}(0,1) \\ \sigma_t^2 &= 0.119 + 0.251 X_{t-1}^2 + 0.159 \sigma_{t-1}^2 \end{aligned} \tag{3.16}$$

where $\widetilde{X_t} = X_t - 0.0923$.

In the last two sections, we have Model 1 and Model 2 fittings for the time series with the spike adjusted by exponential smoothing, and Model A and Model B fittings for the time series with the spike adjusted by threshold clearance. In the next section, we will compare those models as pairs and apply cross-validation to find out the better one respectively.

3.3.5 Model Comparison and Validation for Proportional Negative Sentiment

There are two adjusted forms for proportional negative sentiment: the demeaned time series and the spike removed by exponential smoothing; the original time series and the spike reduced by threshold clearance. Our object is to find the best model for each of these two adjusted forms of time series.

Cross-validation (Picard and Cook, 1984) is implemented here for our time series model selection. This method can be briefly described as a measure of fit by assessing the errors using out-of-sample estimates, and then we can derive a more accurate estimate of model prediction performance (Blum et al., 1999). If we fit the model and compute the MAE only on the training set, we will obtain a biased assessment of how well the model will fit an independent data set. This biased estimate is called the in-sample estimate of the fit, whereas the cross-validation estimate is an out-of-sample estimate.

The general procedure of time series cross-validation can be illustrated as follows: take the first k days out of n days as training data, conduct the model fitting and estimate the model parameters, then predict the values of the following one or more time points using the fitted model, and compare those predictions with the observed values by calculating the Mean Absolute Error (MAE). We can gradually add more data points into the training set (e.g. use the first $(k + j)$ days) for calculating the MAEs and combine them to have an overall estimation of MAE.

The specific cross-validation application (Hyndman, 2011) employed for our time series can be described as follows: we use around half of our data points as subseries

($k = 112$ days out of $n = 217$ days) to fit our models and estimate the parameters. These subseries start at rolling origins, adding one day each time. The prediction is made for the following 7 days, as in this way, the prediction can follow a weekly cycle. We compute the MAE values by the differences between the prediction and the actual values. For each training subseries i, we get seven MAE values $\text{MAE}_{i1}, \text{MAE}_{i2}, \ldots, \text{MAE}_{i7}$. Thus, in the end, we have $(n - k) = 105$ training subseries, making 105 times fittings and conducting 105 times predictions. Finally, a MAE matrix of 105 rows and seven columns can be produced.

For the purpose of comparison, we also make predictions using the exponential smoothing and intend to show the superiority of our fitted models. For the first form of data set, Model 1 from Formula (3.10) and Model 2 from Formula (3.12) are compared. The means of each column are calculated and Figure 3.3.15 shows the differences between the first set of models. From the figure we can find that our models have a smaller overall MAE and Model 2 has the best predictive power.

Similarly, cross-validation is conducted for the two models based on the original time series with the spike adjusted by threshold clearance (Model A from Formula (3.13) and Model B from Formula (3.16)). In this case, a one day forward prediction is applied, as the estimation and prediction for Model B is coded from scratch. The MAE figures can be found in Figure 3.3.16. Because the lines in the Figure represent the MAEs for each iteration, more fluctuations and patterns can be captured. The result shows there is a significant spike from Model B, but Model B performs very well for other parts. Model A does not capture patterns very well. The overall means of MAE for these three models are calculated respectively: 0.223, 0.090, 0.099. We can see that our time series prediction using Model B is still slightly

better than the exponential smoothing prediction.

In conclusion, for the first form of data (the demeaned time series and the spike removed by exponential smoothing), the best model with predictive ability is Model 2; for the second form of data (the spike reduced by threshold clearance), Model B has the greatest predictive power.

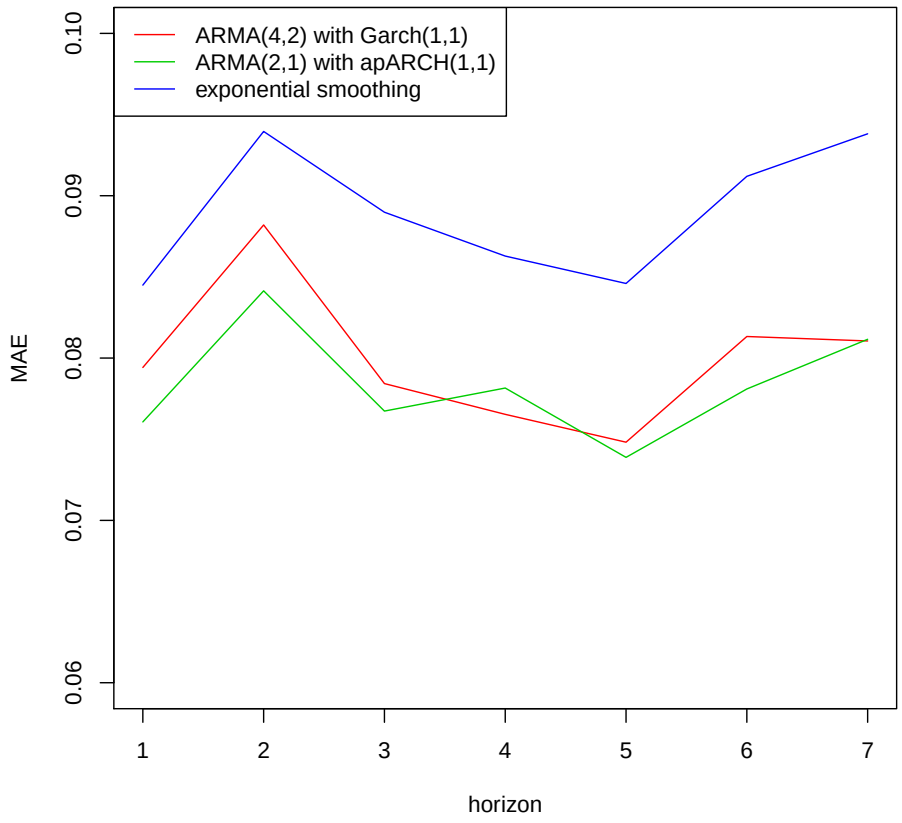

Figure 3.3.15: Comparison among Model 1 (red line), Model 2 (green line), and exponential smoothing (blue line) prediction using Mean Absolute Error for proportional negative sentiment time series (adjusting the huge spike by exponential smoothing).

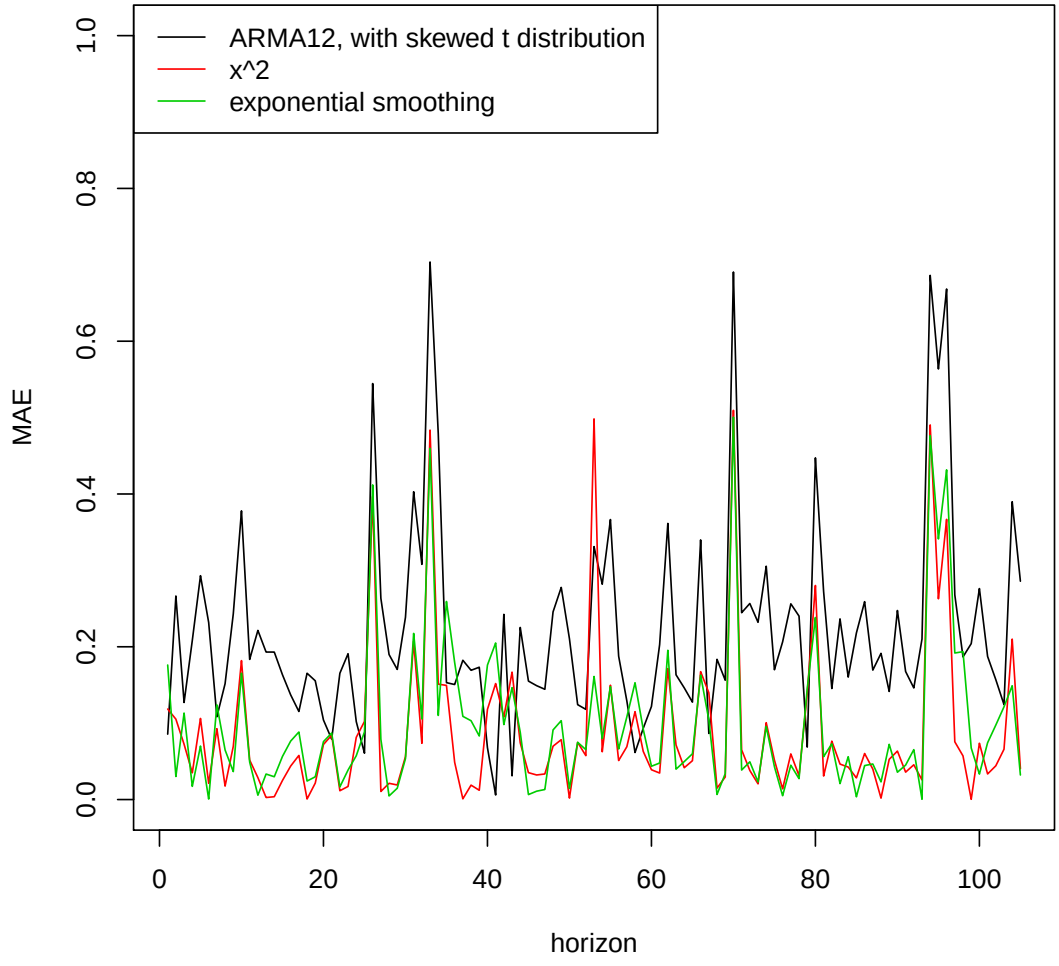

Figure 3.3.16: Comparison among Model A (black line), Model B (red line), and exponential smoothing (green line) prediction using Mean Absolute Error for proportional negative sentiment time series (adjusting the huge spike by threshold clearance).

3.4 Multivariate Time Series Model Fitting

Multivariate time series analysis is adopted when one wants to model and explain the interactions and comovements among a group of time series variables. The object of our time series analyses is the sentiment of a specific company's posts, including both positive and negative. It is reasonable to suspect there is interaction between positive and negative sentiment and to attempt to fit multivariate time series models.

The multivariate generalisation for the univariate autoregressive model (AR model) is the vector autoregressive model (VAR model), which allows for more than one evolving variable (Watson, 1994). Each variable is represented by an equation explaining its evolution based on its own lags and the lags of the other model variables, and all the variables in a VAR model incorporate into the model in the same way.

The general matrix notation of a VAR(p) model can be defined as:

$$\underbrace{\begin{bmatrix} y_{1,t} \\ y_{2,t} \\ \vdots \\ y_{k,t} \end{bmatrix}}_{\mathbf{y_t}} = \underbrace{\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix}}_{\mathbf{c}} + \underbrace{\begin{bmatrix} a^1_{1,1} & a^1_{1,2} & \dots & a^1_{1,k} \\ a^1_{2,1} & a^1_{2,2} & \dots & a^1_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a^1_{k,1} & a^1_{k,2} & \dots & a^1_{k,k} \end{bmatrix}}_{\mathbf{A_1}} \underbrace{\begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ \vdots \\ y_{k,t-1} \end{bmatrix}}_{\mathbf{y_{t-1}}} + \cdots + \underbrace{\begin{bmatrix} a^p_{1,1} & a^p_{1,2} & \dots & a^p_{1,k} \\ a^p_{2,1} & a^p_{2,2} & \dots & a^p_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a^p_{k,1} & a^p_{k,2} & \dots & a^p_{k,k} \end{bmatrix}}_{\mathbf{A_p}} \underbrace{\begin{bmatrix} y_{1,t-p} \\ y_{2,t-p} \\ \vdots \\ y_{k,t-p} \end{bmatrix}}_{\mathbf{y_{t-p}}} + \underbrace{\begin{bmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \\ \vdots \\ \epsilon_{k,t} \end{bmatrix}}_{\epsilon_\mathbf{t}}$$

Our objective now is to examine whether there are interactions and comovements between proportional positive and negative sentiment, and then fit a bivariate VAR model.

To reinforce the symmetrical characteristic, stabilise variance and make the time

series' data more normal distribution-like, we employ the Box-Cox transformation (Box and Cox, 1964) to create a monotonic transformation using power functions. We determine the value of λ by computing and plotting the log-likelihoods for corresponding vector of values of λ (Ripley et al., 2013). We calculate the transformed time series by the definition of the one-parameter Box-Cox transformation:

$$y_i^{(\lambda)} = \begin{cases} \frac{y_i^{\lambda}-1}{\lambda} & \text{if } \lambda \neq 0 \\ \ln(y_i) & \text{if } \lambda = 0 \end{cases}$$

The estimated values of best λ for proportional positive and negative sentiment time series are 0.182 and -0.222 respectively. After transformation, the symmetry test shows that the time series data become symmetrically distributed.

We produce the cross correlation plot for the transformed time series, which serves as a tentative method for identifying the order of the VAR model. From Figure 3.4.1, we can observe that the lag 2 on the right-hand side appears to be mildly significant.

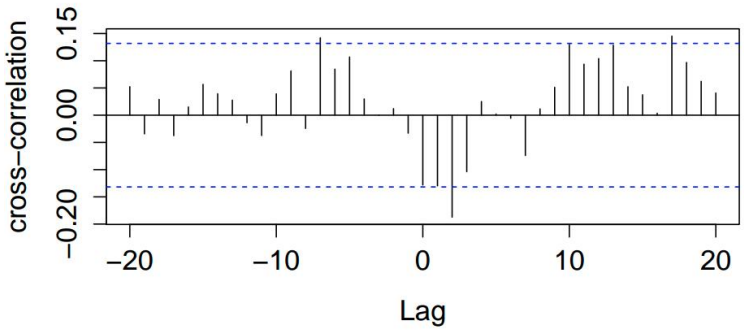

Figure 3.4.1: Cross-correlation plot for transformed proportional positive and negative sentiment time series.

The estimation of the VAR model (Tsay, 2015) verifies the significance of cross correlation at the second order. We refine a fitted VAR model by setting insignificant estimates to zero. The final fitted model is:

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} 0.086 \\ -0.987 \end{bmatrix} + \begin{bmatrix} 0.310 & 0 \\ 0 & 0.228 \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \end{bmatrix} + \begin{bmatrix} 0 & -0.063 \\ 0 & 0.119 \end{bmatrix} \begin{bmatrix} y_{1,t-2} \\ y_{2,t-2} \end{bmatrix} + \begin{bmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \end{bmatrix}$$

From the results of the estimation, we can see that the proportional positive sentiment $y_{1,t}$ is autoregressive on order one $y_{1,t-1}$, and the proportional negative sentiment $y_{2,t}$ is autoregressive on order one $y_{2,t-1}$ and order two $y_{2,t-2}$. The only significantly cross-correlated coefficient is at the second order between $y_{1,t}$ and $y_{2,t-2}$. It shows that positive sentiment after two days is negatively correlated with negative sentiment today. That is to say, if Weibo users post many negative sentiments today, it is likely that after two days there will be fewer positive sentiments about a specific company. For instance, the negative series reached peaks on 14th August and 10th November 2013, and the positive series dropped considerably 2 days later on 16th August and 12th November 2013.

The residual time series plot for $\boldsymbol{\epsilon_t}$ can be found in Figure 3.4.2 and the ACF and PACF plots in Figures 3.4.3 and 3.4.4 respectively. The ACF and PACF for the residuals of the updated model seem to be beyond the line for higher orders (9th or 12th), but they are too large to take into the model. Result also shows that there is no cross-correlation in the residual vectors.

The multivariate ARCH (MARCH) test (Tsay, 2013), which checks the presence of the conditional heteroscedasticity in a vector time series, shows that the

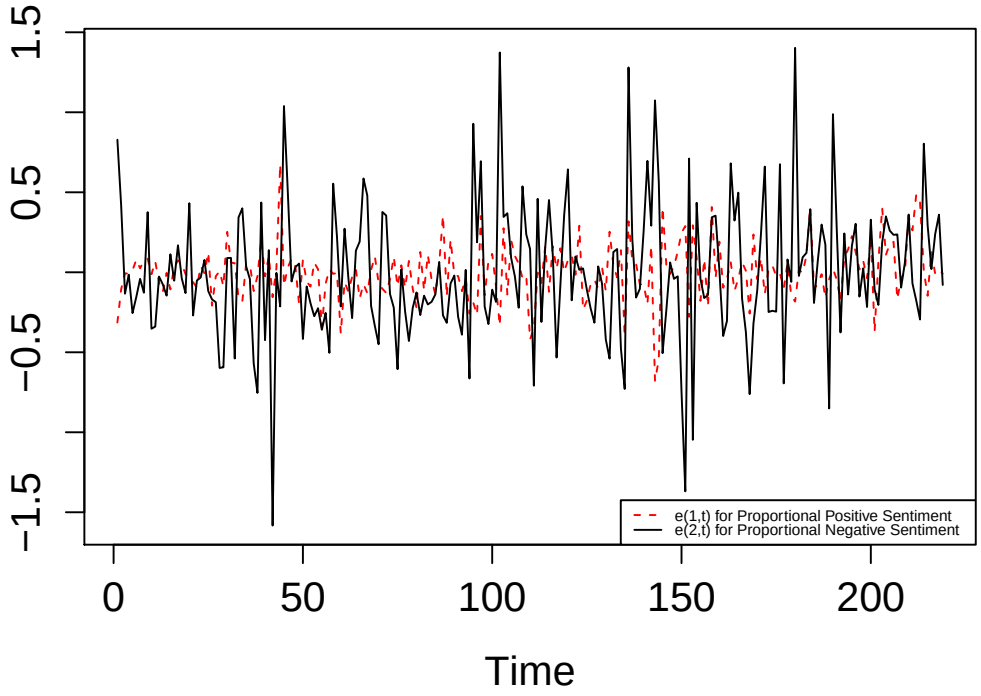

Figure 3.4.2: Residual plot for VAR model.

ARCH effect is significant in the residuals from the VAR model. Then we consider the BEKK model by Engle and Kroner (1995) to model the conditional heteroscedasticity, and it is defined as:

$$\begin{aligned} \epsilon_{\mathbf{t}} &= \mathbf{\Sigma_t^{1/2} z_t} \\ \mathbf{\Sigma_t} &= \mathbf{A_0 A_0'} + \mathbf{A_1 a_{t-1} a_{t-1}' A_1'} + \mathbf{B_1 \Sigma_{t-1} B_1'} \end{aligned} \tag{3.17}$$

where $\mathbf{A_0}$ is a lower triangular matrix and $\mathbf{A_0 A_0'}$ is positive-definite. $\mathbf{A_1}$ and $\mathbf{B_1}$ are $k \times k$ matrices.

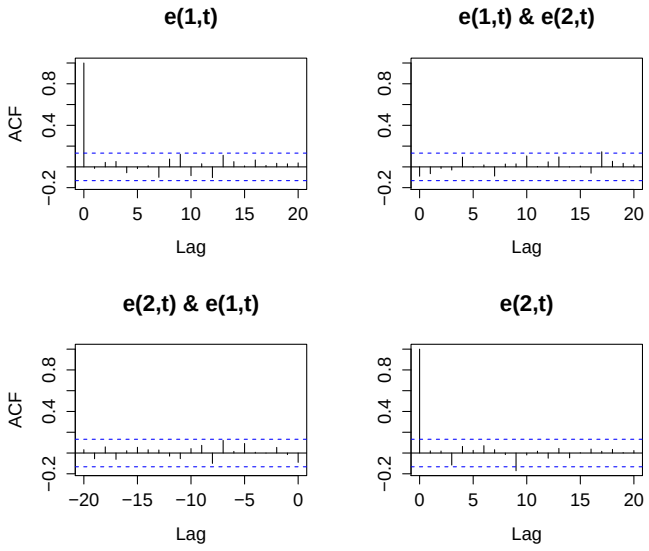

Figure 3.4.3: ACF for the residuals of VAR model.

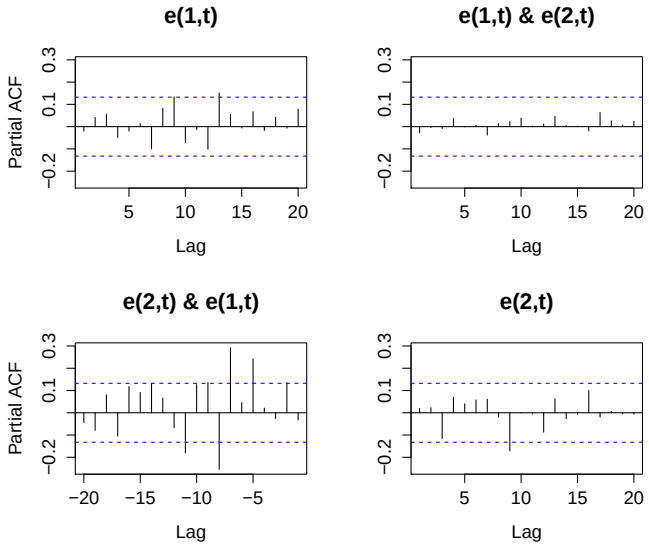

Figure 3.4.4: PACF for the residuals of VAR model.

R packages MTS (Tsay, 2015) and mgarchBEKK (Schmidbauer et al., 2016) are employed for the BEKK model estimation, but both of them fail to show a significant $\mathbf{A_1}$ matrix, which means the main ARCH part for conditional

heteroscedasticity is insignificant. It can be observed that the diagonal values for $\mathbf{A_0}$ $\mathbf{A_1}$ and $\mathbf{B_1}$ are more significant than non-diagonal values, which is sensible. The insignificance might be due to the limitation of the packages and the complexity of real data.

Attention then turns towards modelling the residuals from the positive sentiments $\epsilon_{1,t}$ and from the negative sentiments $\epsilon_{2,t}$ using some univariate time series separately. The ARCH test shows that the ARCH effect is significant for $\epsilon_{1,t}$, but insignificant for $\epsilon_{2,t}$. This is verified by the GARCH estimation for $\epsilon_{1,t}$ and $\epsilon_{2,t}$.

The α for $\epsilon_{1,t}$ is significant with p-value 0.0224 and the estimated α is 0.233. However, the β for the generalised term is insignificant. The QQ plot of residuals shows a satisfactory fit. The fitted model for $\epsilon_{1,t}$ is:

$$\begin{aligned} \epsilon_{1,t} &= z_{1,t}\sigma_{1,t} \\ z_{1,t} &\sim \mathcal{N}(0,1) \\ \sigma^2_{1,t} &= 0.0230 + 0.233\epsilon^2_{1,t-1} \end{aligned} \tag{3.18}$$

However, the α for $\epsilon_{2,t}$ is not significant, indicating the ARCH process is not suitable for the univariate negative residuals. There is no cross volatility clustering effect, and the ARCH effect only exists in the positive residual vector from the VAR model.

The final model for bivariate proportional sentiment time series $\boldsymbol{y_t}$ is:

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} 0.086 \\ -0.987 \end{bmatrix} + \begin{bmatrix} 0.310 & 0 \\ 0 & 0.228 \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \end{bmatrix} + \begin{bmatrix} 0 & -0.063 \\ 0 & 0.119 \end{bmatrix} \begin{bmatrix} y_{1,t-2} \\ y_{2,t-2} \end{bmatrix} + \begin{bmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \end{bmatrix}$$

and

$$\begin{aligned} \epsilon_{1,t} &= z_{1,t}\sigma_{1,t} \\ z_{1,t} &\sim \mathcal{N}(0,1) \\ \sigma_{1,t}^2 &= 0.0230 + 0.233\epsilon_{1,t-1}^2 \\ \epsilon_{2,t} &\sim \mathcal{SN}_{\xi}(0,1) \\ \xi &= 1.102 \end{aligned} \tag{3.19}$$

ξ denotes the rate of skewness as in Formula (3.10).

3.5 Summary

This chapter describes how the sentiment of Weibo posts can be modelled using time series models. Initially, we reviewed different methods for sentiment analyses and adopted the lexicon-based approach for further time series modelling because of the difficulties of obtaining relevant training data and complexity of manually tagging.

A full description of a general framework was provided for fitting time series, which includes model identification, estimation and diagnostic checking. Models with ARMA specification in the conditional mean, GARCH-type conditional variance, and various residual distributions could be regarded as a standard setting when individuals intend to fit a time series. Applications showed that this general framework fits our proportional positive sentiment data well. However, when it was employed in the experiments for negative sentiment time series, we could

not find a highly satisfactory model after numerous trials. We proposed other approaches for fitting the negative time series after adjusting the biggest spike with threshold clearance. The fitted models were then compared and validated using cross-validation.

In the last part, as it is reasonable to suspect an interaction between positive and negative sentiment, we attempted to fit multivariate time series models. The vector autoregressive model and the BEKK model were employed to detect the cross correlation and the conditional heteroscedasticity in the vector residual time series. The results showed that there are some cross correlations between positive and negative sentiment, as the cross-correlated coefficient at the second order is significant.

Chapter 4

Topic Modelling and Randomness Reduction

4.1 Introduction to Topic Models

With the establishment of numerous text mining methods in statistical learning, unstructured textual data can be transformed into quantitative forms and information can be derived via the discovery of patterns over the textual content. Among those methods, Topic Models is a group of statistical methods to discover a set of main "topics" from a large amount of textual data.

LDA, short for Latent Dirichlet Allocation, which is proposed as an improvement of pLSI (probabilistic Latent Semantic Indexing) (Hofmann, 1999) by Blei et al. (2003), is one of the most common models for topic analyses currently in use. In the Section 4.2.1, the development of topic models and the comparison between LDA and pLSI will be illustrated in detail.

LDA will be briefly explained in this paragraph and thoroughly illustrated in the next section. In LDA, each document in the collection is modelled as a mixture over an underlying set of topics; meanwhile, topic probabilities can determine an explicit

representation of each document. It can be regarded as a three-level hierarchical Bayesian model with latent variables and hyper-parameters. The fundamental assumption of LDA is that it assumes words in a document are exchangeable, and thus it makes it a bag-of-words model, which regards text as the multiset (bag) of its words, disregarding word order and grammar. Apart from bag-of-words models, there is syntactic analysis (Pavlidis, 1977) or part-of-speech tagging (Schmid, 1994) which takes grammar and syntax into consideration. Topic analyses usually process a large number of documents and require less syntactic accuracy than speech recognition or web search query, so the bag-of-words assumption was mainly held to capture simplicity and computational efficiency.

To estimate the parameters in LDA and make further inferences, Blei et al. (2003) proposed a Variational Expectation-Maximization algorithm for the learning process; later on, Griffiths and Steyvers (2004) suggested a collapsed Gibbs sampling algorithm for generating the posterior probabilities of the latent variables in the LDA model. LDA and its extensions are regarded as valuable tools to reduce the dimensionality from large groups of unstructured documents to certain amount of "topics". The posterior inference at the document level can be employed for contextual analysis, classification and information retrieval.

While LDA performs generally well for news articles and journals archives, topic models do not fulfil the task very well for finding topics from documents in small length (e.g. less than 140 characters for microblogs) (Wang et al., 2012). The major contribution of this chapter is creating a Randomness Reduction Algorithm applied to post-process the output of topic models, filtering out the insignificant topics and utilise topic distributions to find out the most persistent topics from the

vast amount of microblog posts.

This chapter is organised as follows. Section 4.2 reviews the comprehensive generative process for Latent Dirichlet Allocation and parameter estimation by Gibbs sampling. Section 4.3 presents the initial application of LDA on microblogs' data. The motivation and detailed algorithm of Randomness Reduction are illustrated in Section 4.4. After presenting the empirical evidence for the significance of Randomness Reduction, Section 4.7 unveils the classification and evolution of topics.

4.2 Latent Dirichlet Allocation in Depth

4.2.1 The Development of Topic Models

To process automatically and get insight from a large quantity of textual data, the initial steps could be document clustering and classification. The clustering methods which were applied in Section 2.7 is a way to find out how microblog posts are grouped and clustered through various clustering algorithms. In clustering analysis, we calculated Manhattan distances between documents (posts) from term-document matrices, and the posts were grouped into different clusters using these distances based on k-means or k-medoids algorithms. We can see all the posts are represented by points in the bivariate or trivariate plot, using principal components or multidimensional scaling. For example, in Figure 2.7.2 from Chapter 2 , we can see the clusters in the two-dimensional space and observe approximately how these posts were clustered by looking back into the posts among these

clusters. However, using clustering, it is not time-consuming and straightforward to understand the distinctions between the posts in different clusters and find the most important features from those posts in high-dimensional space. It means that it is difficult to provide the explanations or reasons for the differences between two clusters and conclude the most relevant posts. However, topic models allow the discovery of the abstract "topics" that occur in a collection of documents and the understanding of the distinctions between topics.

Latent Semantic Analysis (LSA), also known as Latent Semantic Indexing (LSI), was introduced by Landauer and Dumais (1997) as a general theory of acquiring similarity and knowledge representation without any prior linguistic or perceptual knowledge. It is based solely on mathematical matrix decomposition methods that achieves powerful inductive effects by smoothing the counts to raise the weight of most informative words (Landauer et al., 1998). Traditionally, LSA applied Singular Value Decomposition to decompose the term-document matrix into three matrices:

$$\underset{(m\times n)}{X} = \underset{(m\times m)}{U}\underset{(m\times n)}{\Sigma}\underset{(n\times n)}{V^T} \tag{4.1}$$

where U and V^T are an orthogonal matrices, and Σ is a diagonal singular value matrix. The columns of U are orthonormal eigenvectors of XX^T, the columns of V are orthonormal eigenvectors of X^TX, and Σ is a diagonal matrix containing the square roots of eigenvalues from U or V in descending order.

By selecting the k largest singular values in the matrix Σ and their corresponding singular vectors from U and V^T, a concept space X' with lower dimensions extracts

most important information and contains minimal noise:

$$\underset{(m\times n)}{X'} = \underset{(m\times k)}{U'}\,\underset{(k\times k)}{\Sigma'}\,\underset{(k\times n)}{V'^{T}} \tag{4.2}$$

It acts as an unique best approximation of the original higher-dimensional space if k is chosen, which emphasises the strongest relationships and allows to keep just the minimum information needed to define the appropriate representation of the dataset. However, the values in matrices U and V^T can be both negative and positive, which can not be interpreted straightforward as unnormalised probabilities of word and document vectors.

Later on, the realisation of LSA was extended by Non-negative Matrix Factorisation (NMF) (Lee and Seung, 2001). It is another decomposition method with a further constraint that the decomposed matrices contain only non-negative values, so it can be regarded as a way to learn unnormalised probability distributions over topics directly. NMF can be expressed as:

$$\underset{(m\times n)}{X} \approx \underset{(m\times k)}{W}\,\underset{(k\times n)}{H} \tag{4.3}$$

The NMF approximation is obtained by iterative updates, and the quality of the approximation is quantified by the cost functions. Lee and Seung (2001) pointed out two algorithms for NMF: minimise the squared Euclidean distance $\min_{W,H\geq 0}\|X - WH\|^2$) or minimise the Kullback-Leibler divergence $\min_{W,H\geq 0} D(X\|WH)$. The unnormalised probabilities are initialised at random, updated according to the iterative update rules, and local minima can be found by the convergence

properties. The detailed algorithms can be found in the paper by (Lee and Seung, 2001). Each vector of X is approximated by a linear combination of the columns of W, and weighted by the components of vectors from H. It can be regarded as the basis of NMF because we can consider each original document in our dataset as being built from a small set of hidden features and NMF generates these features.

Nonetheless, there are a few limitations for LSA. The uniqueness of the NMF solution is not guaranteed and a local optimal solution of the factorisation might be obtained. In general, LSA depends heavily on decomposition methods which can be computationally intensive and can be hard to update as new documents appear. Furthermore, there is no sparsity constraint incorporated into mathematical setup for decomposed matrices. It also lacks a well-defined probability distribution and factors have a clear probabilistic meaning in terms of mixture component distributions (Hofmann, 1999).

On the other hand, significant progress has been made on using probabilistic models to represent textual content in the past few years. In textual data, a corpus is made of a group of documents, and a document d consists of a number of words which denoted by $\mathbf{w}$. The specified document length is denoted by N. Making words unique leads to terms, and vocabulary consists of all possible terms. The initial model for textual analysis is unigram model, which employing a single multinomial distribution to generate each word in every document independently and with the assumption of exchangeability of the words (Diaconis, 1988). It can be represented as:

$$p(\mathbf{w}) = \prod_{i=1}^{N} p(w_i) \tag{4.4}$$

A mixture of unigrams model (McCallum et al., 1998) could be formed by expanding the unigram model. It brings in a distribution over topics, and we first choose a topic z from the distribution and then obtain N words independently from the conditional multinomial $p(w_i|z_k)$. k is the index of topic number and K denotes the total topic numbers here. The probability of a document becomes:

$$p(\mathbf{w}) = \sum_{k=1}^{K} \left(p(z_k) \prod_{i=1}^{N} p(w_i|z_k) \right) \tag{4.5}$$

The major drawback of unigram model is that it assumes each document is represented by only one topic, which is not reasonable for most large collection of textual data. To relax this limitation, Hofmann (1999) proposed the probabilistic Latent Semantic Indexing (pLSI) model, which is also known as probabilistic latent Semantic Analysis (pLSA) and the aspect model. In pLSI, each word in a document is still regarded as a sample from a mixture model. The mixture components from this mixture model are multinomial random variables that represent "topics". Consequently, rather than all the words from a document need to be from the same topic, different words in a document can be generated from diverse topics.

The pLSI model postulates that, given unobserved topics z, words w_i are conditionally independent from document labels d, and the joint distribution can be written as:

$$p(d, w_i) = p(d) \sum_{k=1}^{K} p(w_i|z_k) p(z_k|d) \tag{4.6}$$

The generative process for pLSI is: a document d is chosen with probablity $p(d)$, opt for a latent class z_k (topic) with probability $p(z_k|d)$, and then generate a word w_i

with probability $p(w_i|z_k)$. Through $p(z_k|d)$, which can be regarded as the mixture proportions of the topics for a specified document d, a document in pLSI model is able to include various topics.

The connection between NMF and pLSI (pLSA) has been found by (Gaussier and Goutte, 2005). Later on, it was proved by Ding et al. (2008) that any (local) maximum likelihood solution of pLSA is a solution of NMF with Kullback-Leibler divergence. We can reconstruct the formula for n documents from pLSI as:

$$p(d_n, w_i) = \sum_{k=1}^{K} p(z_k)p(w_i|z_k)p(d_n|z_k) \tag{4.7}$$

Thus, we can find out the first part $\hat{p}(z_k)\hat{p}(w_i|z_k)$ just as w_{ik} from W and second part $\hat{p}(d_n|z_k)$ as h_{kn} from H. It can be regarded that pLSA decomposes the term-document matrix as a linear combination of a set of multinomial distributions over the words (topics) where the weight vectors also follow multinomial distribution. Non-negativity constraint is imposed implicitly due to the probability representations. It has been shown that the underlying computations in NMF and pLSA are identical. However, unlike NMF where there are no additional constraints beyond non-negativity, pLSA bases and weights being multinomial distributions also have the constraint that the entries sum to 1.

It is noteworthy that d in pLSI (pLSA) is a dummy index for the list of documents in the training set. That means d is a multinomial random variable that can take as many values as the amount of documents in training set. The topic mixtures $p(z_k|d)$ are learned only for those documents the model trained on. That character leaves pLSI with no well-defined way to carry out probabilities for a document which is

out of the training set. For a formerly unobserved document, it is impossible to naturally assign probabilities.

The other shortcoming also results from using a distribution indexed by training documents. As the number of training documents goes up, the amount of parameters needing to be estimated magnifies at the same time. Blei et al. (2003) pointed out that it may cause serious problems with over-fitting.

Latent Dirichlet Allocation goes beyond probabilistic Latent Semantic Indexing (pLSI) by changing the constitution of the topic mixture weight and adding Dirichlet priors. The topic mixture weight in LDA turned to be a k-parameter hidden random variable instead of $p(z_k|d)$ in pLSI which are parameters specifically connected with training set. Girolami and Kabán (2003) showed that PLSI is a *maximum a posteriori* estimated LDA model under a uniform Dirichlet prior, so the perceived shortcomings of PLSI can be resolved within the LDA framework. The uniform distribution is actually a special case of the Dirichlet distribution when $\alpha_1 = \alpha_2 = \ldots = \alpha_k = 1$. Blei et al. (2003) clearly illustrated that finding a well-formed probabilistic model at the level of documents is the motivation to develop LDA.

4.2.2 LDA: a Generative Probabilistic Model

In LDA, each topic is regarded as a probability distribution over words, and each document as a probabilistic mixture of these topics. The exchangeability for both documents and words is a crucial assumption for LDA too, and in the literature exchangeable random variables can be represented by a mixture

distribution (De Finetti, 1990). The intra-document statistical structure and latent structures for a set of observations via the mixture distribution and generative process was discovered by Blei et al. (2003), and statistical inference was applied to illustrate the structure. As a mixture model, LDA employs a convex combination of a set of component distributions to model observations. A convex combination is defined as a linear combination of points where weighting proportion coefficients are non-negative and sum to one. LDA postulates that a word w is generated from a convex combination of topics. This can be clarified by the probability of the ith word in a given document (if the total topic number is K):

$$p(w_i) = \sum_{k=1}^{K} p(w_i|z_i = k)p(z_i = k), \quad \sum_{k=1}^{K} p(z_i = k) = 1 \tag{4.8}$$

The formula defines probabilities of words in a similar way as previous models, and we can see that the basic structure remain unchanged. This definition of $p(w_i)$ will be employed later on directly in the model derivation. In the formula, $p(w|z)$ imply which words are crucial to a topic, and $p(z)$ are the probabilities of those topics occurring within a document, which may change for different documents.

This generative process can be described as follows: first, for document m, choosing a distribution over topics $\boldsymbol{\theta}_m$ from a Dirichlet distribution, which determines $p(z)$ for that document; second, selecting a topic k from this distribution via a multinomial experiment, and then picking a word at random from that topic according to $p(w|z = k)$, which is determined by the term distribution $\boldsymbol{\phi}_k$ from another Dirichlet distribution.

Here, $\boldsymbol{\theta}_m$ is the parameter notation for $p(z|d = m)$, which is the topic mixture

proportion (weight) for each document m, indicating topic distribution for document m. $p(z)$ can be stated as a set of M multinomial distributions $\boldsymbol{\theta}$ over the K topics, and we get topic distribution $p(z|d = m) = \boldsymbol{\theta}_m$ for each document m. $\boldsymbol{\alpha}$ is the hyper-parameter for Dirichlet distribution on this mixture proportions $\boldsymbol{\theta}_m$.

$\boldsymbol{\phi}_k$ is the parameter notation for $p(w|z = k)$, which is the mixture component of topic k, indicating term distribution for topic k. $p(w|z)$ can be expressed as a set of K multinomial distributions $\boldsymbol{\phi}$ over the T terms, and we get term distribution $p(w|z = k) = \boldsymbol{\phi}_k$ for each topic k. $\boldsymbol{\beta}$ is the hyper-parameter for Dirichlet distribution on this mixture components $\boldsymbol{\phi}_k$.

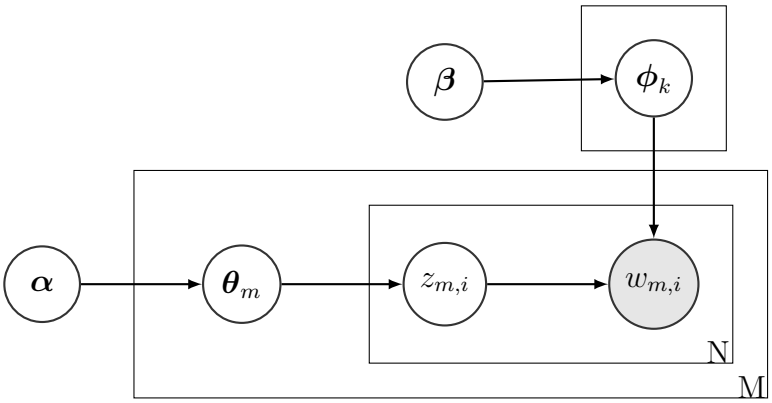

Figure 4.2.1: Bayesian network of LDA.

To make it clearer, a three-level hierarchical Bayesian network representation of LDA is presented in Figure 4.2.1. It can be clarified as:

For each topic k:

$\boldsymbol{\phi}_k \sim Dirichlet(\boldsymbol{\beta})$

For each document m:

$\boldsymbol{\theta}_m \sim Dirichlet(\boldsymbol{\alpha})$

For each word position in m:

$$\begin{aligned} topic \quad & z_{m,i} \sim Multinomial(\boldsymbol{\theta}_m) \\ word \quad & w_{m,i} \sim Multinomial(\boldsymbol{\phi}_k) \end{aligned}$$

In this way, LDA incorporates prior probability distributions on $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ into the last formula, serving as a complete generative model for documents. It can be illustrated as:

$$p(\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{z}, \mathbf{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\boldsymbol{\theta}|\boldsymbol{\alpha})p(\boldsymbol{\phi}|\boldsymbol{\beta}) \prod_{i=1}^{N} p(w_i|z_i, \boldsymbol{\phi})p(z_i|\boldsymbol{\theta}) \tag{4.9}$$

First, we can integrate over $\boldsymbol{\phi}$:

$$p(\boldsymbol{\theta}, \mathbf{z}, \mathbf{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \prod_{i=1}^{N} \int p(w_i|z_i, \boldsymbol{\phi})p(\boldsymbol{\phi}|\boldsymbol{\beta})d\boldsymbol{\phi}p(z_i|\boldsymbol{\theta}) \tag{4.10}$$

In this way, the joint distribution of a topic mixture $\boldsymbol{\theta}$ can be written as:

$$p(\boldsymbol{\theta}, \mathbf{z}, \mathbf{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \prod_{i=1}^{N} p(w_i|z_i, \boldsymbol{\beta})p(z_i|\boldsymbol{\theta}) \tag{4.11}$$

Thus, by integrating over $\boldsymbol{\theta}$, the joint probability of a sequence of words and topics has the form:

$$p(\mathbf{z}, \mathbf{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \int p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \big(\prod_{i=1}^{N} p(w_i|z_i, \boldsymbol{\beta})p(z_i|\boldsymbol{\theta}) \big) d\boldsymbol{\theta} \tag{4.12}$$

4.2.3 Learning LDA by Gibbs Sampling

In LDA, our target of inference is to sample from the posterior distribution:

$$p(\mathbf{z}|\mathbf{w}) = \frac{p(\mathbf{z},\mathbf{w})}{\sum_{\mathbf{z}} p(\mathbf{z},\mathbf{w})} = \frac{\prod_{i=1}^{N} p(z_i, w_i)}{\prod_{i=1}^{N}\sum_{k=1}^{K} p(z_i = k, w_i)} \tag{4.13}$$

We can see that it might be possible to generate the conditional distribution from the joint distribution from previous section. However, it is not achievable to compute this distribution directly, and the difficult part for evaluation is its denominator, which includes a summation over K^N terms. At this point we make use of the Gibbs sampling procedure. The Gibbs sampler is a special case of the Markov chain Monte Carlo (MCMC) algorithm for acquiring a sequence of observations which are approximated from a specified multivariate probability distribution, when direct sampling is difficult. In MCMC, a Markov chain is built to converge to the target distribution, and we take samples from that Markov chain (Gilks et al., 1996; Liu, 2008) after achieving a stationary state. We employed the Gibbs sampler because of its distinctive feature: the dimensions x_i of the distribution are sampled alternately one at a time, conditioned on the values of all other dimensions, which are denoted as $\mathbf{x}_{\neg i}$. It works by first choosing dimension i, and then sampling x_i from $p(x_i|\mathbf{x}_{\neg i})$. Note $\mathbf{x} = (x_1, x_2...x_{i-1}, x_i, x_{i+1}...)$, and $\mathbf{x}_{\neg i} = (x_1, x_2...x_{i-1}, x_{i+1}...)$.

In our case, the desired Gibbs sampler applies the full conditional $p(z_i|\mathbf{z}_{\neg i}, \mathbf{w})$ in the algorithm for the purpose of simulating $p(\mathbf{z}|\mathbf{w})$. By obtaining the joint distribution

$p(\mathbf{z}, \mathbf{w})$, we can generate $p(z_i|\mathbf{z}_{\neg i}, \mathbf{w})$ by

$$p(z_i|\mathbf{z}_{\neg i}, \mathbf{w}) = \frac{p(\mathbf{z}, \mathbf{w})}{p(\mathbf{z}_{\neg i}, \mathbf{w})} = \frac{p(\mathbf{z}, \mathbf{w})}{\sum_{z_i} p(\mathbf{z}, \mathbf{w})} \tag{4.14}$$

Since the term $\mathbf{w}$ in the joint distribution is independent of $\boldsymbol{\alpha}$ and the other term $\mathbf{z}$ is independent of $\boldsymbol{\beta}$, this joint distribution can be factored as below:

$$p(\mathbf{z}, \mathbf{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\mathbf{w}|\mathbf{z}, \boldsymbol{\beta})p(\mathbf{z}|\boldsymbol{\alpha}) \tag{4.15}$$

Then we generate the factorised two parts as follows. The first part is:

$$p(\mathbf{w}|\mathbf{z}, \boldsymbol{\beta}) = \int p(\mathbf{w}|\mathbf{z}, \Phi)p(\Phi|\boldsymbol{\beta})d\Phi \tag{4.16}$$

In this first part $p(\mathbf{w}|\mathbf{z}, \Phi)$ has a Multinomial distribution, and we can generate it by:

$$\begin{aligned} p(\mathbf{w}|\mathbf{z}, \Phi) &= \prod_{i=1}^{N} p(w_i|z_i) = \prod_{i=1}^{N} \varphi_{z_i, w_i} \\ &= \prod_{k=1}^{K} \prod_{\{i:z_i=k\}} p(w_i = t|z_i = k) = \prod_{k=1}^{K} \prod_{t=1}^{V} \varphi_{k,t}^{n_k^{(t)}} \end{aligned} \tag{4.17}$$

where n is a $K \times V$ count matrix, V is the total number of unique terms in the vocabulary, and $n_k^{(t)}$ is the number of times that assign topic k to word (term) t.

The second part $p(\Phi|\boldsymbol{\beta})$ has a Dirichlet distribution:

$$\begin{aligned} p(\Phi|\boldsymbol{\beta}) &= \prod_{k=1}^{K} p(\boldsymbol{\phi}_k|\boldsymbol{\beta}) \\ &= \prod_{k=1}^{K} \frac{1}{\mathbf{B}(\boldsymbol{\beta})} \prod_{t=1}^{V} \varphi_{k,t}^{\beta_t - 1} \end{aligned} \tag{4.18}$$

The normalising constant is the multivariate Beta function, which can be expressed in terms of the gamma function: $\mathbf{B}(\boldsymbol{\beta}) = \frac{\prod_{i=1}^{K} \Gamma(\beta_i)}{\Gamma(\sum_{i=1}^{K} \beta_i)}$.

Combining them together, we see the first part is:

$$\begin{aligned} p(\mathbf{w}|\mathbf{z}, \boldsymbol{\beta}) &= \int p(\mathbf{w}|\mathbf{z}, \Phi) p(\Phi|\boldsymbol{\beta}) d\Phi \\ &= \int \prod_{k=1}^{K} \frac{1}{\mathbf{B}(\boldsymbol{\beta})} \prod_{t=1}^{V} \varphi_{k,t}^{n_k^{(t)} + \beta_t - 1} d\boldsymbol{\phi}_k \\ &= \prod_{k=1}^{K} \frac{\mathbf{B}(\mathbf{n}_k + \boldsymbol{\beta})}{\mathbf{B}(\boldsymbol{\beta})} \end{aligned} \tag{4.19}$$

Then we consider the second part:

$$p(\mathbf{z}|\boldsymbol{\alpha}) = \int p(\mathbf{z}|\Theta) p(\Theta|\boldsymbol{\alpha}) d\Theta \tag{4.20}$$

Similarly, in the first term of the second part:

$$p(\mathbf{z}|\Theta) = \prod_{m=1}^{M} \prod_{k=1}^{K} p(z_i = k|d_i = m) = \prod_{m=1}^{M} \prod_{k=1}^{K} \vartheta_{m,t}^{n_m^{(k)}} \tag{4.21}$$

where $n_m^{(k)}$ is the number of times that topic k is assigned to words in document

m; n_m denotes the m-th row of n. The second term of the second part follows Dirichlet distribution:

$$p(\Theta|\boldsymbol{\alpha}) = \prod_{m=1}^{M} p(\boldsymbol{\theta}_m|\boldsymbol{\alpha}) = \prod_{m=1}^{M} \frac{1}{\mathbf{B}(\boldsymbol{\alpha})} \prod_{k=1}^{K} \vartheta_{m,k}^{\alpha_k - 1} \tag{4.22}$$

Finally, we get the final form of the second part corresponding to that of the first part:

$$\begin{aligned} p(\mathbf{z}|\boldsymbol{\alpha}) &= \int p(\mathbf{z}|\Theta) p(\Theta|\boldsymbol{\alpha}) d\Theta \\ &= \int \prod_{m=1}^{M} \frac{1}{\mathbf{B}(\boldsymbol{\alpha})} \prod_{k=1}^{K} \vartheta_{m,k}^{n_m^{(k)} + \alpha_k - 1} d\boldsymbol{\theta}_m \\ &= \prod_{m=1}^{M} \frac{\mathbf{B}(\mathbf{n}_m + \boldsymbol{\alpha})}{\mathbf{B}(\boldsymbol{\alpha})} \end{aligned} \tag{4.23}$$

Bring these two parts into the original formula:

$$\begin{aligned} p(\mathbf{w}, \mathbf{z}|\boldsymbol{\alpha}, \boldsymbol{\beta}) &= p(\mathbf{w}|\mathbf{z}, \boldsymbol{\beta}) p(\mathbf{z}|\boldsymbol{\alpha}) \\ &= \prod_{k=1}^{K} \frac{\mathbf{B}(\mathbf{n}_k + \boldsymbol{\beta})}{\mathbf{B}(\boldsymbol{\beta})} \prod_{m=1}^{M} \frac{\mathbf{B}(\mathbf{n}_m + \boldsymbol{\alpha})}{\mathbf{B}(\boldsymbol{\alpha})} \end{aligned} \tag{4.24}$$

After generating the extended form of the joint probability $p(\mathbf{w}, \mathbf{z})$, from Formula

(4.14), we can derive the Gibbs updating rule for LDA:

$$\begin{aligned} p(z_i|\mathbf{z}_{\neg i}, \mathbf{w}) &= \frac{p(\mathbf{z}, \mathbf{w})}{p(\mathbf{z}_{\neg i}, \mathbf{w})} = \frac{p(\mathbf{w}|\mathbf{z})}{p(\mathbf{w}_{\neg i}|\mathbf{z}_{\neg i})p(w_i)} \frac{p(\mathbf{z})}{p(\mathbf{z}_{\neg i})} \\ &\propto \frac{\mathbf{B}(\mathbf{n}_k + \boldsymbol{\beta})}{\mathbf{B}(\mathbf{n}_{k,\neg i} + \boldsymbol{\beta})} \frac{\mathbf{B}(\mathbf{n}_m + \boldsymbol{\alpha})}{\mathbf{B}(\mathbf{n}_{m,\neg i} + \boldsymbol{\alpha})} \\ &= \frac{\Gamma(n_k^{(t)} + \beta_t)\Gamma(\sum_{t=1}^{V} n_{k,\neg i}^{(t)} + \beta_t)}{\Gamma(n_{k,\neg i}^{(t)} + \beta_t)\Gamma(\sum_{t=1}^{V} n_k^{(t)} + \beta_t)} \frac{\Gamma(n_m^{(k)} + \alpha_k)\Gamma(\sum_{k=1}^{K} n_{m,\neg i}^{(k)} + \alpha_k)}{\Gamma(n_{m,\neg i}^{(k)} + \alpha_k)\Gamma(\sum_{k=1}^{K} n_m^{(k)} + \alpha_k)} \\ &= \frac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^{V} n_{k,\neg i}^{(t)} + \beta_t} \frac{n_{m,\neg i}^{(k)} + \alpha_k}{(\sum_{k=1}^{K} n_m^{(k)} + \alpha_k - 1)} \end{aligned} \tag{4.25}$$

In the end we can get

$$p(z_i|\mathbf{z}_{\neg i}, \mathbf{w}) \propto \frac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^{V} n_{k,\neg i}^{(t)} + \beta_t} (n_{m,\neg i}^{(k)} + \alpha_k) \tag{4.26}$$

where the counts $n_{.,\neg i}^{(.)}$ represent the number of the tokens that the ith term is excluded from the corresponding documents or topics.

This full conditional distribution can be employed in the Gibbs sampling algorithm 1 and the algorithm can be described as: by sequentially sampling all variables from their distribution when conditioned on the current values of all other variables and the data, we can attain the subsequent state.

This Gibbs sampler using Formula (4.26) to assign words to topics, and the counts in this formula are calculated from the subset of words proceeded so far, rather than the entire dataset. Running the chain for several iterations, each time it creates a new state by sampling each z_i from the distribution particularised by

Formula (4.26). After sufficient iterations for the chain to converge to the target distribution, we capture the current values of the z_i variables.

Finally, for any single sample we can estimate the multinomial parameter sets ϕ and θ from the state of the Markov chain, $\mathbf{z}$, and by definition we can get:

$$\phi_{k,t} = \frac{n_k^{(t)} + \beta_t}{\sum_{t=1}^{V} n_k^{(t)} + \beta_t} \tag{4.27}$$

$$\theta_{m,k} = \frac{n_m^{(k)} + \alpha_k}{\sum_{k=1}^{K} n_m^{(k)} + \alpha_k} \tag{4.28}$$

Applying Formulae (4.26),(4.27) and (4.28), the Gibbs sampler in Algorithm 1 can be run. After initialisation, burn-in and sampling, finally we can read out the value of Φ, Θ according to the updated ultimate counts.

Algorithm 1: Gibbs Sampling Algorithm for LDA.

Input: word vectors $\mathbf{w}$, start value of hyper-parameters α and β, topic number K

Data: count statistics $n_m^{(k)}$, $n_k^{(t)}$ and their sums n_m, n_k, memory of full conditional array $p(z_i|.)$

Output: topic association $\mathbf{z}$, multinomial parameters Φ and Θ, hyperparameter estimates α and β

// initialisation;

zero all count variables, $n_m^{(k)}$, n_m, $n_k^{(t)}$, n_k;

for *all documents* $m \in [1 : M]$ **do**

  for *all words* $n \in [1 : N_m]$ *in document* m **do**

    sample topic index $z_{m,n} = k \sim Mult(1/K)$ for word $w_{m,n}$;

    increment document-topic count: $n_m^{(k)}$ ++;

    increment document-topic sum: n_m ++;

    increment topic-term count: $n_k^{(t)}$ ++;

    increment topic-term sum: n_k ++;

  end

end

// Gibbs sampling over burn-in period and sampling period

while *not finish* **do**

  for *all documents* $m \in [1 : M]$ **do**

    for *all words* $n \in [1 : N_m]$ *in document* m **do**

      // for the current assignment of k **to a term** t **for word** $w_{m,n}$**:**

      decrement counts and sums: $n_m^{(k)}--$; n_m--; $n_k^{(t)}--$; n_k--;

      // multinomial sampling according to Formula (4.26)

      sampling topic index $\tilde{k} \sim p(z_i|\mathbf{z}_{\neg i}, \mathbf{w})$

      // for the new assignment of $z_{m,n}$ **to the term t for word** $w_{m,n}$**:**

      increment counts and sums: $n_m^{(\tilde{k})}$++; n_m++; $n_{\tilde{k}}^{(t)}$++; $n_{\tilde{k}}$ ++;

    end

  end

  // check convergence and read out parameters

  if *converge and L sampling iterations since last read out* **then**

    // the different parameters read outs are averaged

    read out Φ, Θ according to Formulae (4.27) and (4.28)

  end

end

4.3 Application of LDA on Microblogs' data

4.3.1 Data Pre-processing for LDA

The initial steps for data pre-processing are similar to the initial text mining for term frequency and cluster analysis. One extra step is to build a tf-idf (term frequency-inverse document frequency) matrix (Salton et al., 1975) . In text mining and semantic analysis, we employ tf-idf to modify the raw matrix counts with the purpose of removing the words with extremely high frequency and making rare words weighted more heavily than common words. For example, a word that occurs in only 5% of the documents should possibly be weighted more heavily than a word that occurs in 95% of the documents.

It can be also regarded as a statistic to reflect the importance of a word to a document in a collection or corpus. Its value goes up proportionally to the number of times a word turns up in the document, but is offset by the frequency of the word in the corpus, and thus it helps to correct the fact that some words appear more frequently in general. The first part of detailed tf-idf formula is the term frequency and the second part is the inverse document frequency:

$$\text{tf-idf}_{i,j} = \frac{N_{i,j}}{N_{*,j}} \log\left(\frac{D}{D_i}\right) \tag{4.29}$$

$N_{i,j}$: the number of times a term i occurs in a document j

$N_{*,j}$: the number of total words in document j

D : the number of documents

D_i : the number of documents in which word i appears

In our case, we reduced the number of features by only keeping terms with tf-idf > 0.1 in documents, and we prepared the final document term matrix for the LDA model.

4.3.2 Parameter Control for Functions

We employed the *topicmodels* package (Hornik and Grün, 2011; Grün et al., 2015) in R to implement the LDA model on our microblogs' data. In the settings, *Gibbs* is set as the approximate inference algorithm, and the number of Gibbs sampling draws is controlled by three parameters: *burnin*, *thin* and *iter*. They show that, for each run, the first *burnin* iterations are discarded at the beginning and then every *thin* iteration is returned for *iter* iterations. In our case, *burnin* is set to be 5000, *thin* to be 100 and *iter* as 1000 after many experiments. It means we used a burn-in of 5000 followed by 1000 draws with a thinning of 100 and all draws are returned. Specifically, we set the number of omitted Gibbs iterations at the beginning to make sure we only conduct sampling after the Markov Chain reaches a steady state. It is often practical to leave an appropriate lag of iterations between subsequent read-outs to obtain decorrelated states of the Markov chain. This interval is often named "thinning interval" or sampling lag.

We need to specify values for the parameters of the prior distributions for estimation using Gibbs sampling. Griffiths and Steyvers (2004) suggest starting values of $50/k$ for α and 0.1 for β. Furthermore, for fitting the LDA model to a given document-term matrix, the topic number needs to be fixed a-priori. Models with

several different numbers of topics are fitted and the perplexity, which is equivalent to the geometric mean per-word likelihood, is employed as a measurement for evaluating the models on held-out data. The perplexity assesses how well a probability model predicts a sample, thus it can be a way to compare probability models. For Gibbs sampling, the perplexity can be represented as

$$\text{Perplexity}(\mathbf{w}) = \exp\left(-\frac{\log(p(\mathbf{w}))}{\sum_{d=1}^{D}\sum_{t=1}^{V} n^{(td)}}\right) \tag{4.30}$$

$n^{(td)}$ denotes how often the tth term occurred in the dth document.

In literature, Goldenberg et al. (2010) carried out a survey and comparison for three general tools: cross-validation, nonparametric mixture priors, and marginal likelihood. They pointed out that choosing the number of clusters in a mixture model is a complicated statistical issue and there is no single best solution.

Cross-validation is the most widely used method (Hornik and Grün, 2011) and we applied a 5-fold cross-validation for topic number selection. The original sample is randomly partitioned into 5 equal sized subsamples. Among these 5 subsamples, a single subsample is retained as the validation data for testing the model, and the remaining 4 subsamples are used as training data. From the previous formula of perplexity, we can see that if we keep the word amount the same, the larger the log-likelihood, the lower the perplexity. Therefore, the perplexity of models with different topic numbers is calculated and we choose the model with the lowest perplexity. The optimal topic number of one month's data is around 40 and the optimal topic number of one week's data is around 30 by measuring perplexity. As we carry out an algorithm after model fitting to filter out the insignificant topics,

the original set-up of topic numbers becomes less important to some extent.

4.4 Randomness Reduction

4.4.1 Motivation and the Literature

Gibbs Sampling is a randomised algorithm, therefore it may generate diverse results each run. In addition, there is no guarantee that all of the topics generated by LDA represent coherent subjects. It is probable that part of them will be meaningless or even "junk". To choose and retain more stable and well-explained topics for further topic evolution analysis, we intend to reduce the randomness and select the most "significant" topics for every single period of time.

First, it is essential to provide an unconventional definition for the "significance" in our particular case. In literature, topic coherence is the most wide-spread measurement to evaluate topic quality after topic modelling (Chang et al., 2009), and the output of topic models can be improved significantly if one can figure out methods to identify automatically coherent and incoherent topics. In the traditional approaches, distances between topic words under a specific topic were calculated via various methods, and the underlying assumption is that all the terms in a specific topic should be associated with a common theme. That is to say, the closer the top n topic words, the higher the topics' coherence. For example, a topic with "cat" "kitten" "dog" "puppy" "cute" will have higher coherence score than a topic with "cat" "weather" "election" "beauty" "boring", as the words in the former topic are "closer" than the latter ones.

In early researches, some extrinsic methods (Chang et al., 2009) were applied to measure models' performance for specific tasks, e.g. accuracy for information retrieval (Wei and Croft, 2006) and perplexity for predictive likelihood in held-out documents (Wallach et al., 2009). However, these approaches do not provide information about the interpretability of topics for humans.

Newman et al. (2010) proposed a method for automatically computing topic coherence by estimating if all or the most of its top-10 words are related, which was proved to be highly correlated with human evaluation. Word relatedness is predicted using the Pointwise Mutual Information (PMI) (Church et al., 1989). The PMI of each pair for the top ten words was calculated and estimated from the entire corpus of over two million English Wikipedia articles as an external reference corpus (approximate 1 billion words). Topic coherence models based on WordNet similarity and search engine-based similarity using google title matches and log hits matches provided less consistent results, by computing the correlation with human judgements.

Later on, Mimno et al. (2011) showed that available co-document frequency of words in the training corpus can be used to automatically evaluate semantic coherence without using human annotators or reference collections outside the training data. The logic of this method is the co-occurrence of words within documents in the corpus can indicate semantic relatedness. Topic coherence in their model is defined as the sum of the log ratio between co-document frequency and the document frequency for the N most probable words in a topic.

Turney et al. (2010) and Erk (2012) suggested distributional approaches are often implemented in vector space models. Turney et al. (2010) provided a

survey on past work with vector space models according to the structure of the matrix (term-document, word-context, or pair-pattern). Under the distributional hypothesis by Harris (1954), words with similar meanings tend to occur in a similar context. Aletras and Stevenson (2013) explored distributional semantic similarity methods to measure the coherence of words generated by a topic model. Each topic word is represented as a bag of highly co-occurring context words from the semantic space which is created using Wikipedia as a reference corpus, and the words are weighted using either PMI or a normalised version of PMI (NPMI). Results showed that measures on the fuller vector space are comparable to the state-of-the-art proposed by Newman et al. (2010), while performance consistently improves using a reduced vector space.

The way we define the "significance" of topics is different from traditional topic coherence measurements due to the particularity of our data set, i.e. microblog posts. In our case, every single microblog post is regarded as a document and we intended to measure the existence of topics among a huge amount of microblog posts. A post may contain apparently incoherent topic words, but if we link all of them together, it may result in a sensible topic. For instance, if we take a look at an empirical example in Figure 4.5.1, which will be explained fully later, we can find out the top topic words are "companies" "land" "evade" "value added tax" "including" "well-known" "a huge amount", which are not seemingly with a high score for topic coherence if we measure the similarity of the top-n most probable words. Our objective is now to find out, based on learned topic words, whether we can figure out the most wide-spread microblog posts and track the topic trends or not, rather than pursuing the coherence of the topics.

To discover the most relevant topics that are particularly informative about the content of posts, we intended to find out the most frequently appearing terms from the top n words in topics among different sample sets and to see if they are more specific to the topics than other words. Having conducted a comprehensive search, we could hardly find any well-established methods to deal with these issues.

Initially, by intuition, we considered aggregating and averaging the term probabilities from term-topic distributions and then setting a threshold and choosing the most frequent terms. For example, if we have two sets of topic words, the probabilities of the top n words from one sampled topic-term distribution are: exercises 0.16, workout 0.124, well-being 0.051, swim 0.045, boxing 0.023, gym 0.012, badminton 0.011, life 0.009, air 0.007, etc.; and for a topic from another sampled term distribution which is initialised using a different random seed, we may get: exercises 0.164, workout 0.12, swim 0.05, well-being 0.041, boxing 0.025, gym 0.014, badminton 0.009, life 0.008, air 0.005, etc. The average term probability of this topic would then be: exercises 0.162, workout 0.122, swim 0.0475, well-being 0.046, boxing 0.024, gym 0.013, badminton 0.01, life 0.0085, air 0.006, etc. If we set the threshold for the smallest probability as 0.01, the five most important words for this topic would be selected and we may infer the topic as "sport".

In the literature, stability selection (Meinshausen and Bühlmann, 2010) is a method designed to improve the performance of a selection procedure by aggregating the results of the selection of subsamples of the data. The key concept for stability selection is that by subsampling the data set (sample size is typically $n/2$) without replacement and conducting variable selection or structure estimation on each subsample, the aggregated result, which is in the form of probabilities, can provide

a control on the false discoveries rate. This selection performs well in the finite sample setting and for the high-dimensional data. Compared to previous studies, Meinshausen and Bühlmann (2010) put the subsampling and selecting procedure into a more formal framework and provided evidences of empirical and theoretical advantages. However, in our case, the topics generated from Gibbs sampling cannot be easily averaged over iterations or aggregated directly due to lack of identifiability: the *ith* Topic is theoretically not constrained to be similar to the *ith* Topic in another sample.

This problem is also known as the label switching problem in mixture models. The posterior mean of $\hat{\phi}$ and $\hat{\theta}$ is used as the final estimate, but the sequence of the conditional interaction groups between runs may not be the same (Callaghan, 2013). The problem used to be dealt through the Relabeling algorithm (Stephens, 2000), which labels estimates for each single run. The Relabeling algorithm employs the Kullback-Liebler (KL) divergence, quantifying the distance of the estimates between runs to a specific reference Q distribution, which is the true parameter in a simulation study or the estimate from the first run if a real dataset is being investigated. The order of the $\hat{\phi}$ estimate changes so that each $\hat{\phi}_k$ is ordered to the same index as the Q_k based on the minimum KL distance, and then the $\hat{\theta}$ can be ordered accordingly.

4.4.2 The Algorithm

The algorithm which we proposed to address this label switching problem is Randomness Reduction. It can be briefly stated as selecting the significant topics

with high common occurrences from results generated from Gibbs sampling. It is more straightforward and computationally easier than previous approaches. Furthermore, it has been shown that, besides solving the unidentifibility problem, as an effective by-product, the Randomness Reduction algorithm can also help to filter out "insignificance" topics and only keep "significance" ones.

The Randomness Reduction algorithm is laid out in Algorithm 2. Empirical examples at both the word and topic levels will be given later to clarify the process. In the algorithm, the most basic and elementary function is $Intersection$, which matches each word pair from two different topics and finds the words in common from those two topics as an intersection. The function $RandomnessReduction$ updates the intersections, finds the best intersections with the maximum length, and confirms the topic matchings. The crucial idea of our algorithm is after matching and comparing initial sets of topics from different random seeds (from parallel chains started at different initial values) and get the intersection, we use the intersection for further comparisons, instead of using the whole sets all the time. At each step of the Randomness Reduction, the current best intersection is generated from former intersections and the topics at this step, rather than looking ahead and comparing all the possibilities in the future steps, which can guarantee the strict optimal intersection.

Assuming we get n set of topics as the results from n random seeds, each set includes K topics. If we consider the full matching and comparison algorithm which compares all the possible combinations for n different sets (inspect all the possible pairs for n dimensions), the computational complexity increases dramatically with dimensions: around K^n combinations need to be matched and compared. However,

Algorithm 2: Randomness Reduction Algorithm.

Input: N: total number of topic sets; $Topics_1, \ldots, Topic_N$ ($Topics_i$: topics from set i); T: minimum words number (threshold) for each topic
Output: $BestIntersection$: significant topics sets

```
for i ∈ [1 : N − 1] do
    if i == 1 then
        BestIntersection_i = RandomnessReduction(Topics_i, Topics_{i+1})
    else
        BestIntersection_i = RandomnessReduction(BestIntersection_{i−1}, Topics_{i+1})
    end
end
```

```
function RandomnessReduction(BestIntersection, Topics)
    minLength = T
    newBestIntersection = null
    foreach topicA ∈ bestIntersection do
        foreach topicB ∈ Topics do
            WordsInCommon = Intersection(topicA, topicB)
            if length(WordsInCommon) >= minLength then
                minLength = length(WordsInCommon)
                newBestIntersection = WordsInCommon
            end
        end
    end
    return (newBestIntersection)
```

```
function Intersection(topicA,topicB)
    WordsInCommon = null
    foreach wordA ∈ topicA do
        foreach wordB ∈ topicB do
            if wordA == wordB then
                WordsInCommon = WordsInCommon ∪ wordA
            end
        end
    end
    return (WordsInCommon)
```

if we continue the comparison using intersections, the total number of pairs that we need to match and compare become less than $K^2(n-1)$. For example, if we have 10 sets of topics and 40 topics for each set and we compare all the possible pairs and finally get exactly the best matched intersections, we need to have $40^{10} = 1.04 \times 10^{16}$ calculations. Instead, using our algorithm, the total number of calculations would only be less than $40^2 \times 9 = 14,400$. Our algorithm shares a similar intuition with greedy algorithms, and it is a simplified matching and comparison process with relatively low computational complexity.

A threshold (T, minimum words number for each topic) is set in our algorithm to ensure the number of common words in an intersection must be no less than T. If we set a higher T, there would be fewer topics left; if we set a lower T, it is hard to conclude topics from the remaining topic words. This threshold filtered out the intersections (topics) with words less than T, and we could figure out the most significant topics over all the topics.

In the Algorithm Box 2 and the running empirical example in Figure 4.4.1, the Randomness Reduction algorithm is illustrated within a group of random seeds. After having completed the Randomness Reduction within a group (e.g. for each 10 random seeds), we match the topics between groups using a similar algorithm. Using the "best intersections within a group" from each group, we do the Randomness Reduction to find out the "best intersections between groups". For instance, if we have 5 groups and each group has ten sets (in total 50 sets from 50 different random seeds), first we apply Randomness Reduction on ten sets for each group and get five "best intersections within a group", then use Randomness Reduction on the five "best intersections within a group" to get the final "best

intersections between groups".

To examine and validate our algorithm, we change the sequence of different sets "within a group" and "between groups", and start from different random seeds to see if they lead to similar outcomes. Results show that the words from final best intersections stayed almost the same, which means the remaining topics were much the same regardless of the order of matching and comparisons between different random seeds. Although this algorithm is not exactly as precise as calculating all possible combinations, it generates satisfactory results and is computationally much easier. Further evidence for the effectiveness of the Randomness Reduction algorithm will be given in the next section.

Empirical Examples at both the word level and the topic level can be found in the next section. They describe how the algorithm proceed in detail and clarify each step of the algorithm using real data from Weibo.

4.4.3 Empirical Examples for Randomness Reduction

In our empirical studies, the topics generated and initialised by a single random seed can be regarded as a set, and a group is made to have 10 sets. The number of topics for each set (random seed) K is fixed at 40 as discussed in Section 4.3.2. For each topic, the top 20 terms (words) are recorded for the Randomness Reduction algorithm. The threshold (T) of the minimum number of common words is set to 8 after numerous trials on our dataset in order to keep a balance between the number of remaining topics and the informative of topic words. $topic_{i,j}$ represents the jth topic from ith set (seed), e.g. topic 10 from the 1st set is $topic_{1,10}$. To take

samples from topic models, we generate n groups of m random numbers ($m \times n$ in total) by a random permutation of the elements between 1 to 10^9 as random seeds.

To clarify the algorithm, a running empirical example at the word level for the best intersection within a group can be found in Figure 4.4.1. It is important to note that for each seed there are 40 topics, but we only list the matched topic, not the other 39 topics. The resulting best intersection is one of the final best intersections within a group. We can see from the first two columns that the Topic 4 from Seed 1 matches the Topic 15 from Seed 2, and all the top-ranked 20 topic words stay the same. Then we use the 3rd column, which is the intersection from Seed 1 and Seed 2 (1st Intersection), to compare with the words from Seed 3 in the 4th column, and Topic 36 is the best match. Except one word (marked in grey), all other words match exactly. The 2nd Intersection comes from the matches between the 1st Intersection in the 3rd column and the topic words from Seed 3 in the 4th column. Then we use the 2nd Intersection to match topic words from Seed 4 to get the 3rd Intersection, etc. In general, the nth Intersection comes from the (n-1)th Intersection and the words from Seed n+1. All the words marked in grey disappear at that step. In this empirical example, the number of remaining topic words in the final intersection is 14, which is much higher than the threshold ($T = 8$).

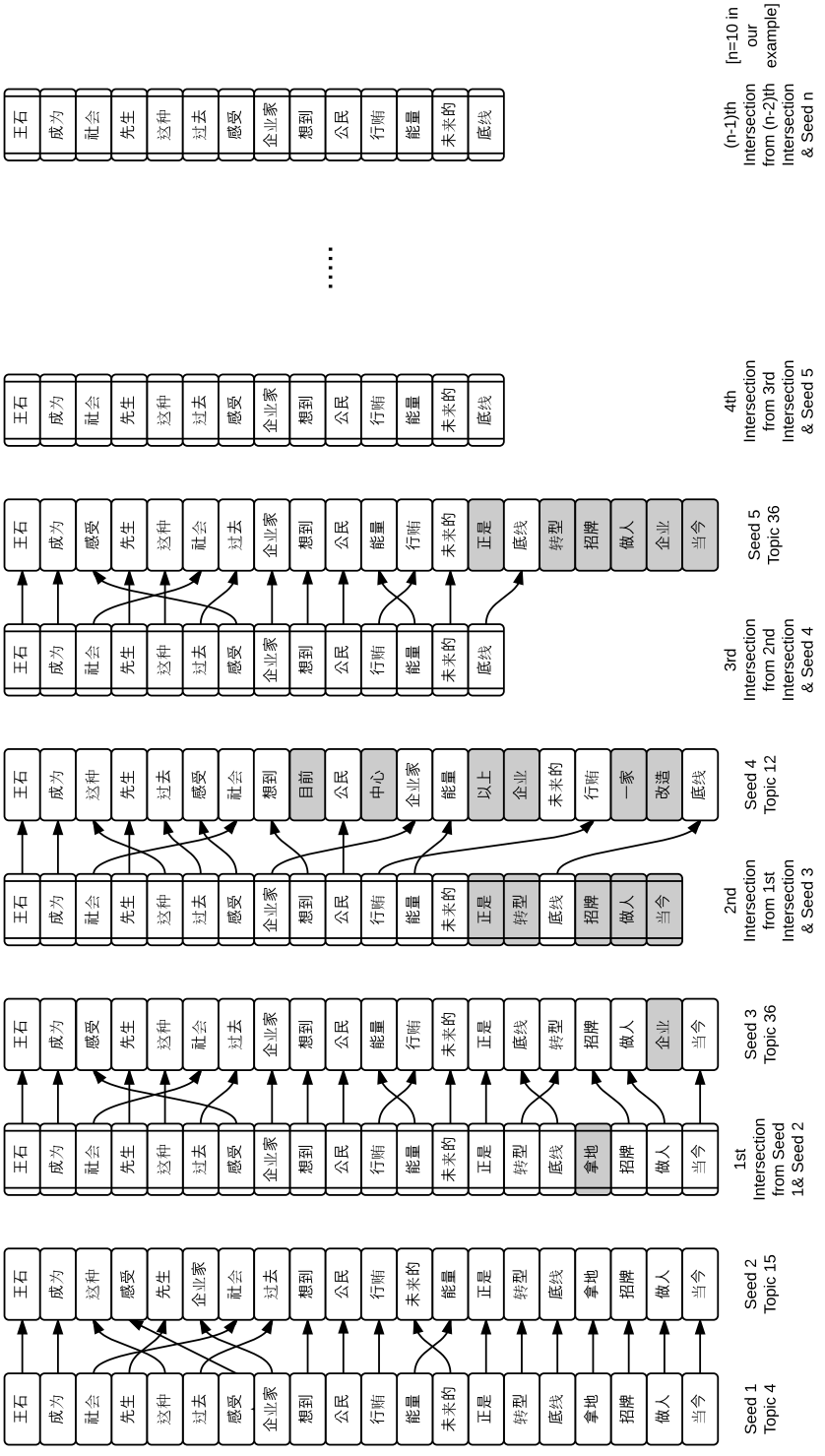

Figure 4.4.1: Randomness Reduction empirical example at the word level.

This empirical example at the word level shows word matchings and comparisons for Randomness Reduction, but it does not include all the topics for a random seed and does not have an overview at the topic level. As a result, the empirical example at the topic level is introduced here. It is described below and the corresponding coloured chart representation for the initial steps can be found in Table 4.1.

We begin from $i = 1$ and match the words from the $topic_{1,1}$ with the words from all the topics from the 2nd set ($topic_{2,j}, j \in [1 : 40]$). For instance $topic_{1,1}$ has 4 words in common with $topic_{2,1}$; it has 2 words in common with $topic_{2,2}$; it has 8 words in common with $topic_{2,3}$; it has 10 words in common with $topic_{2,4}$;... and it has 4 words in common from $topic_{2,10}$. After comparison, we found out that the best matched topic (with the largest number of common words) for $topic_{1,1}$ from the 2nd set of topics is $topic_{2,4}$. They have 10 common words ($wordsInCommon$) and the number is above our threshold T. The common words for this best match are recorded as a $BestIntersection$, which is a subset of the original topic words. Then we do the same for the remaining nine topics (2nd, 3rd, etc.) from the 1st set $topic_{1,2}$ against all the topics in the 2nd set. For instance, we can see from Table 4.1 that $topic_{1,1}$ matches $topic_{2,4}$, $topic_{1,3}$ matches $topic_{2,6}$, and $topic_{1,7}$ matches $topic_{2,2}$.

After finding all the $BestIntersection$s (with the largest number of words in common) from the comparisons between all the 1st set and topics from the 2nd set, we regard the words from $BestIntersection$s as updated topics and employ these $BestIntersection$s (updated topics) to match all the topics from the 3rd set $topic_{3,j}$, instead of using all the original words from the 1st set and the 2nd set to match the 3rd set. Next, we use these further improved $newBestIntersection$s from

these three sets to match the topics in the 4th set $topic_{4,j}$, etc... Using this method, the final intersections are generated by matching the words from the intersections which are drawn gradually from set 1 to set 9 with all the topics in the set 10. In this way, we can figure out the most significant topics out of forty for ten different sets (from 10 random seeds) within a group.

Table 4.1 indicates three examples in different colors at the topic level for our algorithm, and we can see the route of matched pairs via the same color. For instance, the blue color in this table, $topic_{1,1}$ matches $topic_{2,4}$ (with the largest number of words in common), and then the words from their intersections match $topic_{3,6}$, and the further intersections match $topic_{4,3}$, ...until the final intersection generated from $topic_{10,5}$. The green and red colors are discontinued because the numbers of common words in their intersections drop below the threshold when matching 6th and 9th set respectively. After getting the most significant topics (final intersections) within this group (10 seeds), as described before, we can find out significant topics between several groups using Randomness Reduction again. For many topics, the corresponding further intersections would drop below T and thus discontinue after several steps. That means the number of remaining final intersections will be much less than the original topic numbers, and only "significant" intersections (topics) with more than T common words can survive. This is a valuable by-product, as it acts as an automatic filter machine and only keeps the most frequently appearing topics.

Table 4.1: Randomness Reduction empirical example for the best final intersections at the topic level.

Set (Seed)	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	final intersections
Topics	1	1	1	1	1	1	1	1	1	1	
	2	2	2	2	2	2	2	2	2	2	
	3	3	3	3	3	3	3	3	3	3	
	4	4	4	4	4	4	4	4	4	4	
	5	5	5	5	5	5	5	5	5	5	
	6	6	6	6	6	6	6	6	6	6	
	7	7	7	7	7	7	7	7	7	7	
	8	8	8	8	8	8	8	8	8	8	
	9	9	9	9	9	9	9	9	9	9	
	...	...	...	...	...	...	...	...	...	...	
	40	40	40	40	40	40	40	40	40	40	

4.5 Case Studies for the Significance of Randomness Reduction

To demonstrate the effectiveness of Randomness Reduction, we conducted several case studies to check whether this procedure made topics clearer and lessened the insignificant topics. As illustrated in Section 4.4.1, the traditional topic coherence measurement is not appropriate for our dataset and we must find out a different way to evaluate the significance of topic-word groups.

The method we applied to check the significance of topic words is to search all the topic words which remained after reduction online using Weibo's search function or using other search engines (Google and Baidu). Weibo's search function is provided to find out specific posts within a set period with a cluster of keywords. The group of topic words would be defined as significant (detailed categories will be provided later) if we can find a specific popular post from the searches or if we can discriminate topics from these words distinctly.

We have three categories for finding topics: Y means we could find the specified topic from the topic words exactly; N means we are not able to find the specified topic from the topic words; NE (Not Exactly) means we could just suspect potential topics from the topic words (inconclusive). For instance, from Table 4.2, the total amount of grouped topic words before Randomness Reduction is 40, among which there are 8 Y, 25 N, and 7 NE, accounting for 20%, 17.5%, and 62.5% respectively. That means from all topic words groups, we could find 20 percent of groups with specified topics, and for 17.5 percent of groups in which we could only guess the topics. The remaining 62.5 percent is the group for which we could not find

corresponding topics.

Figure 4.5.1 and 4.5.2 show an example of "significant" topics from LDA after Randomness Reduction. The remaining topic words are eight out of twenty, and indicating a news about Vanke (well-known real estate company) evaded a huge amount of land value added tax. The related posts could be found out by using search function on Weibo and six examples of related posts can be found in the figures as well. Except simply forwarding the news, some examples include comments and discussions on the news. This topic can also be found in the following months, and several example posts are listed in Figures C.1.1 and C.1.2 in the Appendix as well.

From Table 4.2 to Table 4.8, we investigated the results for seven months as seven different cases, both before and after Randomness Reduction to check the significance. We can see from the summary in Table 4.9 that the overall percentage of Y is boosted from around 21.4% to approximately 75%, which is a huge growth. If we include NE as part of Y, the topics which we could recognise also increase from 42.1% to 90.6%.

Through investigating the topics case by case, we find in most cases those meaningless topic words are filtered out and only significant words are kept. For instance, in one case, using the eight words which remained after reduction to search, we are able to find the specified post directly. But if we applied all the 20 topic words or unfiltered top 10 from the list to search online, the result turns out to be not found or ambiguous. In another case, the reduction process deleted redundant words significantly and only maintained useful words, e.g. "Bank of China" and "hundreds of billions of dollars" (but "China" and "billions of dollars"

Example One:
Topic words:
"公司" "土地" "拖欠" "增值税" "包括" "律师" "知名" "大量"
("companies" "land" "evade" "value added tax" "including" "lawyer" "well-known" "a huge amount")

Related posts in that month:
Eg.1@火星人袁泽敏:【土地增值税漏洞推高房价 专家称房企应回应质疑】日前，律师李劲松对 29 家上市的房企提出质疑，称包括万科、SOHO 在内知名的公司大量拖欠应该上缴的土地增值税金，总金额至少高达 640 亿元。那么 29 家上市房... http://t.cn/zRLZNjU
(@huoxingrenyuanzemin: Lawyer Jingsong LI challenged 29 listed real estate companies, accusing Vanke and other well-known companies of evading on land value added tax, with the total amount of at least 64 billion Yuan)

Eg.2@老姜也不辣耶: 企业受益，国家和民众受损，土地增值税缘何征缴不力？央视讲万科 SOHO 中国等房企拖欠 3.8 万亿土地增值税，请问国家征管部门干啥去了？
(@laojiangyebulaye: It benefited those companies, but deprived nation and people. How could they evade those land value added tax? CCTV said Vanke and SOHO evaded land value added tax of 3.8 trillion. What did State Administration departments do?)

Eg.3@黎华 916:【万科富力等 45 家房企被曝拖欠巨额土地增值税】11 月 25 日，《每周质量报告》报道称，调查发现多家知名房地产公司欠缴土地增值税，且数目惊人。2005 年至 2012 年 8 年间，房地产开发企业应交而未交的土地增值税总额超过 3.8 万亿元。包括 SOHO 中国、富力、万科、招商地产等知名房企均榜上有名。
(@Lihua916: [Vanke, R&F and other 45 well-known housing companies were reported evaded on land value added tax] On November 25, "Weekly Quality Report" reported that the investigation found that several well-known real estate companies evaded a huge amount of land value added tax. From 2005 to 2012, real estate development enterprises evaded payable land value added tax by total more than 3.8 trillion yuan, including SOHO China, R&F, Vanke, China Merchants Property and other well-known housing companies on the list.)

Eg.4@yemao0315: 央视说万科 SOHO 中国等房企拖欠 3.8 万亿土地增值税。不管这个事情到底是谁对谁错，已经再次证实房价居高不下的源头其实是国家，光一种土地增值税，还是拖欠的就有 3.8 万亿，其他税费有多少？建议广大老百姓都别买房，看他们找谁收税，辛辛苦苦赚点钱不容易啊。
(@yemao0315: CCTV reported SOHO China Vanke and other well-known housing companies evade land value added tax for 3.8 trillion. Regardless of whether this matter in the end is right or wrong, it has been confirmed once again the source of high prices is actually from the government: tax evasion only for land value added tax is already 3.8 trillion, so how much for other taxes? I Recommend that the majority of people should not buy a house, then government has no chance to collect extra taxes. It is hard to earn money.)

Eg.5@马江洲：调查发现多家知名房地产公司欠缴土地增值税，且数目惊人。报道称，2005 年至 2012 年 8 年间，房地产开发企业应交而未交的土地增值总额超过 3.8 万亿元。其中包括万科、雅居乐、金地、华远、龙湖、绿城、莱蒙、明发、花样年等等！这些企业，在南京又拖欠多少呢？ http://t.cn/8kAasiE
(@Majiangzhou: [45 Housing companies evaded 3.8 trillion land value added tax] The investigation found that several well-known real estate companies evaded the land value added tax by 3.8 trillion, a huge amount. Reported that from 2005 to 2012 eight years, real estate development enterprises evaded payable land value added tax by total more than 3.8 trillion yuan, including Vanke, Agile, Golden, Huayuan, Lake, Greentown, Lemmon, Mingfa, Fantasia, and so on! How much do these enterprises owe in Nanjing? http://t.cn/8kAasiE)

Eg.6@理财周刊:【央视报道房企欠税 3.8 万亿】据央视报道，长期关注房地产的北京执业律师李劲松根据关数据，对没有征收到位的土地增值税做出分析测算，结果发现，全国房地产开发企业应缴未

Figure 4.5.1: Example one: topic words, orginal posts and follow-up posts (Part1).

缴的土地增值税总额超过 3.8 万亿，万科、富力、招商、金地、SOHO 等知名企业均存在拖欠。
(@FinanceWeekly: [CCTV reported that housing companies evaded 3.8 trillion land value added tax] According to the report from CCTV, Jinsong Li, a Beijing real estate lawyer, who focused on real estate industry for a long-term, analysed and measured land value-added tax evasion. He found real estate development enterprises evaded payable land value added tax by total more than 3.8 trillion yuan. Vanke, R & F, China Merchants, Gold, SOHO and other well-known enterprises are present in arrears.)

Follow-up posts in the following months:
@liao 三元闲士:【央视：万科 SOHO 中国等房企拖欠 3.8 万亿土地增值税】据央视《每周质量报告》报道，长期关注我国房地产业发展的北京职业律师、注册会计师、注册税务师李劲松，，此前根据国家统计局、财政部、以及国家税务总局公布的房地产相关数据，对没有征收到.... http://t.cn/8kG0Xa0（分享自 @凤凰财经）
(@liaosanyuanxianshi: According to CCTV "Weekly Quality Report", by long-term focusing on China's real estate development, Beijing professional lawyer, certified public accountant, registered tax consultant, Jinsong LI pointed out SOHO China and Vanke may owe 3.8 trillion land value added tax) via CCTV (share from @ Phoenix Finance)

@任志强: 各公司分别已经公告的情况直接证明了央视的荒谬。 //@新浪房产:【王石：万科不存在欠缴土地增值税的情况】
(@Zhiqiang Ren (CEO of Huayuan Estate): the announcements from all companies have already directly proved the ridiculousness of CCTV. //@ Real Estate News: [Wang: Vanke is not a case of the presence of non-payment of land value added tax])

@成都商业地产官博:【万科等房企声援任志强】24 日，央视报道 SOHO 中国、万科在内的 45 家房地产企业，均有土地增值税拖欠行为，拖欠总额超过 3.8 万亿元。华远地产董事长任志强发博反驳，万科在内的多家企业声援任志强，“任志强说的很清楚，房地产项目要达到清算条件时才需要清算并交纳土地增值税”。http://t.cn/8kLPXC7
(@Chengdushangyedichanguanbo: [Housing companies such as Vanke support Ren] 24th, CCTV reported SOHO, Vanke, including 45 real estate companies, evaded land value added tax by more than 3.8 trillion yuan. Zhiqiang Ren, chairman of Huayuan Real Estate, refuted. A number of housing companies, including Vanke, supported Ren: "What Ren said is clear: only when real estate projects meet the requirement of settlement of land appreciation tax do the developer need to make the corresponding settlement and payment." http://t.cn/8kLPXC7)

@CTOCE:针对 24 日央视关于 45 家房企拖欠 3.8 万亿土地增值税的报道，昨日，万科等对于华远地产董事长任志强对此事的回应，央视报道提及的爆料人—北京执业律师李劲松昨日又发表“任志强：专业人士叫你补习税务常识”的长博文，对任志强的回应进行逐条反驳， http://t.cn/8kLIslP
(@CTOCE: For the report on 24th from CCTV on 45 housing companies evaded 3.8 trillion land value added tax, yesterday, Zhiqiang Ren(chairman of Huayuan Real Estate)'s response on the matter, Beijing lawyers Li Jinsong (who broke the news in CCTV) yesterday published a long post "Ren: tax professionals ask you to study tax knowledge" to refute Ren's responses one by one. http://t.cn/8kLIslP)

Figure 4.5.2: Example one: topic words, orginal posts and follow-up posts (Part2).

are eliminated).

Table 4.2: Improvement by Randomness Reduction for the first month (3rd May to 2nd June). Y means we could find the specified topic from the topic words exactly; N means we are not able to find the specified topic from the topic words; NE (Not Exactly) means we could just suspect potential topics from the topic words (inconclusive)

Y/N/NE(not exactly)	Count	Percentage
Y	8	20%
N	25	62.5%
NE(not exactly)	7	17.5%
Total	40	100%
After Randomness Reduction	Count	Percentage
Y	3	100%
N	0	0
NE(not exactly)	0	0
Total	3	100%

Table 4.3: Improvement by Randomness Reduction for the second month (2nd Jun to 1st Jul).

Y/N/NE(not exactly)	Count	Percentage
Y	9	22.5%
N	24	60%
NE(not exactly)	7	17.5%
Total	40	100%
After Randomness Reduction	Count	Percentage
Y	4	66.6%
N	1	16.7%
NE(not exactly)	1	16.7%
Total	6	100%

Table 4.4: Improvement by Randomness Reduction for the third month (1st Jul to 1st Aug).

Y/N/NE(not exactly)	Count	Percentage
Y	9	22.5%
N	27	67.5%
NE(not exactly)	4	10%
Total	40	100%
After Randomness Reduction	Count	Percentage
Y	3	75%
N	0	0
NE(not exactly)	1	25%
Total	4	100%

Table 4.5: Improvement by Randomness Reduction for the fourth month (1st Aug to 1st Sep).

Y/N/NE(not exactly)	Count	Percentage
Y	8	20%
N	20	50%
NE(not exactly)	12	30%
Total	40	100%
After Randomness Reduction	Count	Percentage
Y	4	80%
N	0	0
NE(not exactly)	1	20%
Total	5	100%

Table 4.6: Improvement by Randomness Reduction for the fifth month (1st Sep to 1st Oct).

Y/N/NE(not exactly)	Count	Percentage
Y	7	17.5%
N	23	57.5%
NE(not exactly)	10	25%
Total	40	100%
After Randomness Reduction	Count	Percentage
Y	3	75%
N	0	0%
NE(not exactly)	1	25%
Total	4	100%

Table 4.7: Improvement by Randomness Reduction for the sixth month (1st Oct to 1st Nov).

Y/N/NE(not exactly)	Count	Percentage
Y	10	25%
N	21	52.5%
NE(not exactly)	9	22.5%
Total	40	100%
After Randomness Reduction	Count	Percentage
Y	4	66.7%
N	2	33.3%
NE(not exactly)	0	0
Total	6	100%

Table 4.8: Improvement by Randomness Reduction for the seventh month (1st Oct to 1st Nov).

Y/N/NE(not exactly)	Count	Percentage
Y	9	22.5%
N	22	55%
NE(not exactly)	9	22.5%
Total	40	100%
After Randomness Reduction	Count	Percentage
Y	3	75%
N	0	0
NE(not exactly)	1	25%
Total	4	100%

Table 4.9: Summary of results for Randomness Reduction.

Y/N/NE	1st	2nd	3rd	4th	5th	6th	7th	Total No.	Overall Percentage
Y	8	9	9	8	7	10	9	60	21.4%
N	25	24	27	20	23	21	22	162	57.9%
NE	7	7	4	12	10	9	9	58	20.7%
Total	40	40	40	40	40	40	40	280	100%
After RR	1st	2nd	3rd	4th	5th	6th	7th	Total No.	Overall Percentage
Y	3	4	3	4	3	4	3	24	75%
N	0	1	0	0	0	2	0	3	9.4%
NE	0	1	1	1	1	0	1	5	15.6%
Total	3	6	4	5	4	6	4	32	100%

4.6 Significance of Randomness Reduction on Twitter Datasets

In the previous section, the significance of the Randomness Reduction algorithm has been verified by the self-collected Weibo data. To further examine the effectiveness of Randomness Reduction on the sets of topic words, we employed the algorithm on the topic words generated from Twitter.

The pre-processing steps for English tweets (posts) are similar to the Chinese ones, we cleaned up the dataset by excluding punctuations, stopwords, numbers and urls, and built up the document-term matrix. We also employed tf-idf, which was fully explained in Section 4.3.1 to remove the words with extremely high frequency and make rare words weighted more heavily than common words.

Using topic modelling initialised by multiple random seeds, we generated different sets of topic words from tweets, and then searched them back using Google to see if we can find specific topics from those words. Afterwards, we applied Randomness Reduction to the topic words from different sets, got processed topic words, and searched back again.

In the following subsections, two specific Twitter datasets were employed to conduct topic modelling and Randomness Reduction, and they provided further evidence for the significance of the Randomness Reduction. Several examples for topic words are listed and the percentages of meaningful topics are concluded before and after the Randomness Reduction. We can notice a difference between Weibo posts and Twitter tweets from the topics generated from Twitter and Weibo: there are many

promotions and advertisements on Weibo, but they seldom appear on Twitter. Topics from Twitter are mainly related to popular news and general discussions about the keywords companies. Except these two sets, we also inspected a dataset with hashtag #Amzn which is the nasdaq stock ticker for company Amazon. However, the topics generated from those tweets via topic modelling are generally meaningless either before and after Randomness Reduction: most of them are stock market related and with plenty of tickers from other companies; and almost no content about the company and its products.

4.6.1 Randomness Reduction on Tweets about Apple

The first Twitter dataset is about company Apple. It is with size 170,000 and was collected using the hashtag #Apple on the 29th April 2016 (TwitterInc., 2016).

Some examples of topic words after applying the Randomness Reduction are listed below:

- “icahn” “carl” “billionaire” “investor” “shares” “stake” “says” “china” “dumps” “stock” “sold” “entire”
- “apps” “carekit” “first” “the” “apples” “four” “health” “hit”
- “single” “hot” “check” “changer” “itunes” “cia” “ciadm” “monde”

The first set of topic words is related to a news about the Billionaire investor Carl Icahn selling his entire stake in Apple, and the second set is related to the first four apps with Apple CareKit, which is a software platform designed to help developers

improve personal health and which hit the App Store on that day. However, for the third set, we can hardly find out any related news, posts, topics or coherent meanings.

In summary, the comparative results are listed in the Table 4.10. Before applying Randomness Reduction, there is only 36.7% chance that we could find specific topics from the topic words, and the percentages for unable to find and quite hard to find are 46.7% and 16.7% respectively. However, after adopting Randomness Reduction, we can find 64.6% topic words direct to a clear topic, only 25% left for totally unknown and 10% for ambiguous ones. The reduced number of topics also helps us to narrow down the potential topics and do further topic evolution analysis. Randomness Reduction proves to be effective again on this Twitter dataset.

Table 4.10: Summary of results for Randomness Reduction on Twitter data set about Apple

Y/N/NE	1st	2nd	3rd	4th	5th	Total No.	Overall Percentage
Y	8	12	10	10	15	55	36.7%
N	16	14	14	13	13	70	46.7%
NE	6	4	6	7	2	25	16.7%
Total	30	30	30	30	30	150	100%
After RR	1st	2nd	3rd	4th	5th	Total No.	Overall Percentage
Y	6	7	5	5	8	31	64.6%
N	4	2	2	3	1	12	25%
NE	1	1	1	2	0	5	10.4%
Total	11	10	8	10	9	48	100%

4.6.2 Randomness Reduction on Tweets about US Airlines

The second Twitter dataset with size 14,640 is about multiple US airlines: Virgin America, United, Southwest, Delta, US Airways and American Air. Individuals tweets about those airline companies in February 2015 via @ the company names, e.g. @Delta.

The nature of the generated topics from this data set is slightly different from the previous ones, as the tweets are about multiple companies and via @, not hashtag #. The topics generated are more general about the whole airline industry, not only about a specific company. Furthermore, some of the tweets can be regarded as a channel for passengers speaking to the airlines. These features can be observed from the generated topics, and four sets of topic words from the 1st group after the Randomness Reduction can be seen as an example:

- "cancelled" "flight" "tomorrow" "morning" "dfw" "rebooked" "rebook" "dallas" "flighted"
- "call" "phone" "back" "someone" "line" "system" "called" "hung"
- "bag" "still" "luggage" "baggage" "bags" "lost" "checked" "claim" "found"
- "service" "customer" "rude" "terrible" "rep" "poor" "disappointed" "horrible"

The first, third and fourth topics are marked as significant, and the second is marked as not very clear. We can see the first set is related to flight cancellation and rebooking; the third is about luggage claim, and the fourth is mainly negative

comments/complaints about customer service. The topic of the second set is relatively ambiguous: it might be related to phone system of the airline.

The overall result from this dataset is presented in Table 4.11, and it shows that although there are less topics remained after applying the Randomness Reduction as we set topic numbers as 20 for this dataset, all the remaining topics are marked as "Y" or "NE". It means we filtered out all the insignificant topics via Randomness Reduction and only kept those meaningful ones from all the sets of topic words. The Randomness Reduction boosts the well-constructed topic rate from 33% to 76.2%, which evidences again the significance of our algorithm.

Table 4.11: Summary of results for Randomness Reduction on Twitter dataset about US Airlines

Y/N/NE	1st	2nd	3rd	4th	5th	Total No.	Overall Percentage
Y	7	6	8	6	6	33	33%
N	7	7	9	9	6	38	38%
NE	6	7	3	5	8	29	29%
Total	20	20	20	20	20	100	100%
After RR	1st	2nd	3rd	4th	5th	Total No.	Overall Percentage
Y	3	3	4	4	2	16	76.2%
N	0	0	0	0	0	0	0%
NE	1	1	2	0	1	5	5%
Total	4	4	6	4	3	21	100%

4.7 Topic Classification and Evolution

To describe the changes of topics over time, we would like to introduce topic evolution analysis. If we are able to classify our various topics into several categories, it would be easy to specify changes. After investigation, we categorised the remaining topics about company Vanke into seven categories.

First, via topic words online searching, those topics that we are not able to find the exact topics for are tagged as "NOT FOUND" (same as N and NE in Table 4.9). Here we count NE (not exactly) into the "NOT FOUND" group, which limits the topics in the "FOUND" group (same as Y in Table 4.9) which are all well-defined topics with explicit context.

For the "FOUND" type, there are four specified categories: "Popular post from Wang (chairman of the board in Vanke)", "News", "Advertisement", and "Promotion"; and the other two categories are general "Popular post" and "Others". "Others" means we can find out the specified posts but it is difficult to allocate them to any of the four categories above; however, they are still distinguished to some extent, not general "Popular post". For instance, in our case studies, only two topics are in the "Others" group: one is "Discussion" and the other one is "Popular competitor CEO's post".

Table 4.12 concludes the topic categories with overlapped periods (see Table 4.13 later), so it includes larger topic numbers than the previous section. It shows that after Randomness Reduction we are able to discover specific topics for more than four-fifths (85%) of groups of topic words. The largest category in "Found" type is general "Popular post" (28.6%) and then in the second place is "News" (19.6%).

"Popular post from Wang", who is the chairman of the board, surprisingly counted for 16.1% among all the topic word groups. It reveals that the impact of social celebrities on Weibo in China is remarkable.

Table 4.12: Summary of topic categories.

Categories	Label	Amount	Percentage
Popular post from Wang	A	9	16.1%
News	B	11	19.6%
Advertisement	C	6	10.7%
Promotion	D	4	7.1%
Popular post	E	16	28.6%
Others(Discussion, Popular competitor CEO's post)	F	2	3.6%
NOT FOUND	G	8	14.3%
Total		56	100%

We further labelled different topics (A1, A2, A3, etc.) in different categories, and the detailed contexts of the labelled topics can be found from the topics list later or in the Appendices. Table 4.13 concludes topic evolution over time with overlapping periods for the labelled topics. We use different shades of red to represent "Popular post from Wang", different shades of yellow to represent "News", different shades of green to represent "Advertisement" and "Promotion", and different shades of blue to represent other "Popular post" for visualisation. For instance, there are four different "Popular post from Wang" throughout the whole period, and they are labelled as A1, A2, A3 and A4. We can see topics A1, A2 and A4 continued for two overlapping periods, e.g. Topic A1 appeared during both 2 Jun - 1 Jul and 15 Jun - 15 Jul. Different persistent topics are marked by different colors in Table 4.13, from which we can observe many topics lasted for exactly two overlapped periods. There are only a few topics maintained more than two overlapped periods. Several non-coloured non-persistent topics can be found from the table as well, e.g. E1,

E2, B2, etc.

After turning the overlapping periods into non-overlapping periods in Table 4.14 (from the beginning of months) and Table 4.15 (from the middle of months), the persistent topics are reduced to a fairly small proportion. Only two topics (B3 and B5 in Table 4.15) or three (A3, B1/7 and C1 in Table 4.15) persisted during the whole period.

Table 4.13: Monthly topic evolutions with overlapped periods. Same persistent topics marked by the same color. Different shades of red to represent "Popular post from Wang", different shades of yellow to represent "News", different shades of green to represent "Advertisement" and "Promotion", and different shades of blue to represent other "Popular post".

Time							
3 May - 2 Jun	E1	D1	B1				
15 May - 15 Jun	E2	D1					
2 Jun - 1 Jul	B2	A1	G	C1	G	D2	
15 Jun - 15 Jul	B3	A1	E3				
1 Jul - 1 Aug	B3	C2	B4	C1			
15 Jul - 15 Aug	B3	C2	G	C1	A2	D3	
1 Aug - 1 Sep	G	E4	F1	C1	A2		
15 Aug - 15 Sep	F2	E4	B5				
1 Sep - 1 Oct	A3	E5	B5				
15 Sep - 15 Oct	A3	G	B5	A4			
1 Oct - 1 Nov	E6	E7	G	A4	G	E8	
15 Oct - 15 Nov	E6	E7	E9	E10	G	B6	E11
1 Nov - 1 Dec	A3	E12	B7	E10			

Table 4.14: Monthly topic evolutions without overlapped periods (Part1).

Time							
3 May - 2 Jun	E1	D1	B1				
2 Jun - 1 Jul	B2	A1	G	C1	G	D2	
1 Jul - 1 Aug	B3	C2	B4	C1			
1 Aug - 1 Sep	G	E4	F1	C1	A2		
1 Sep - 1 Oct	A3	E5	B5				
1 Oct - 1 Nov	E6	E7	G	A4	G	E8	
1 Nov - 1 Dec	A3	E12	B7	E10			

Table 4.15: Monthly topic evolutions without overlapped periods (Part2).

Time							
15 May - 15 Jun	E2	D1					
15 Jun - 15 Jul	B3	A1	E3				
15 Jul - 15 Aug	B3	C2	G	C1	A2	D3	
15 Aug - 15 Sep	F2	E4	B5				
15 Sep - 15 Oct	A3	G	B5	A4			
15 Oct - 15 Nov	E6	E7	E9	E10	G	B6	E11

We consider shortening the length of data for topic modelling from one month to one week and intend to find out if there would be more persistent topics for shorter time periods. The possibility here is that a topic may appear across one and a half months, and our original one month's model may only capture it once, but if we run topic models on one week's data, it may appear six times for each week during this one and a half month period. The results of weekly topic evolutions can be found in Table 4.17 to Table 4.21, and the yellow cells represent persistent topics.

Table 4.16: Summary of topic categories by month and by week.

		by Month		by Week	
Categories	Label	Amount	Percentage	Amount	Percentage
Popular post from Wang	A/a	9	16.1%	15	12.8%
News	B/b	11	19.6%	39	33.33%
Advertisement	C/c	6	10.7%	18	15.38%
Promotion	D/d	4	7.1%	9	6.84%
Popular post	E/e	16	28.6%	18	15.38%
Others (Discussions, etc.)	F/f	2	3.6%	7	15.38%
NOT FOUND	G/g	8	14.3%	12	10.26%
Total		56	1	117	1%

Table 4.16 compares the difference between topics generated from one month's and one week's data. The results are based on the same dataset and the time span is between 3 May and 1 Dec 2013; the only difference is the topic modelling is carried out on different windows of data: from one month changed to one week. The total topic numbers are surprisingly doubled for one week's data, which indicates by shortening the time periods, many new topics are detected by topic models. It could be the case that some topics may appear to be hot topics for only two weeks. Our original model for one month's period could not capture them, but they are recognised by topic modelling on one week's data. However, due to many suddenly appearing new topics, the proportion of persistent topics is still very low. After

investigation, we found only 30 percent are long-lived among all the topics, which makes the further topic evolution analysis difficult. However, from Table 4.16, we could conclude several interesting findings: first, the percentage of "Popular post from Wang" decreased from 16.1% to 12.8% for shorten periods, indicating that the posts of chairman of the board seem to last for a long time; second, "News" and "Discussion" had much larger percentages for one week's period, and they might be regarded as hot topics more often in short periods, and only stayed for a short time (one or two weeks); third, the Randomness Reduction performed even better for short period (we could not find corresponding popular posts for only 1/10 groups of topic words).

Table 4.17: Weekly topic evolutions (part1).

3May10May	1(b)	2(b)	3(d)															
10May17May			3(d)	4(c)	5(e)													
17May24May			3(d)			6(d)	7(b)	8(b)	9(b)									
24May31May										10(f)	11(b)	12(e)						
31May7Jun													13(e)	14(b)	15(c)	16(a)		
7Jun14Jun																16(a)	17(e)	18(b)
14Jun21Jun																		
21Jun28Jun				4(c)														
28Jun5Jul				4(c)														
5Jul12Jul				4(c)														
12Jul19Jul																		
19Jul26Jul				4(c)														
26Jul2Aug				4(c)														
2Aug9Aug																		
9Aug16Aug																		
16Aug23Aug				4(c)														
23Aug30Aug				4(c)														
30Aug6Sep																		
6Sep13Sep																		
13Sep20Sep																		
20Sep27Sep																		
27Sep4Oct				4(c)														
4Oct11Oct				4(c)														
11Oct18Oct																		
18Oct25Oct																		
25Oct1Nov																		
1Nov8Nov																		
8Nov15Nov																		
15Nov22Nov									9(b)									
22Nov29Nov		2(b)*4																
29Nov6Dec		2(b)																

Table 4.18: Weekly topic evolutions (part2).

3May10May																		
10May17May																		
17May24May																		
24May31May																		
31May7Jun																		
7Jun14Jun	19(a)																	
14Jun21Jun		20(e)	21(a)	22(d)	23(c)	24(a)												
21Jun28Jun							25(b)	26(b)										
28Jun5Jul									27(b)	28(d)								
5Jul12Jul											29(e)	30(b)	31(f)					
12Jul19Jul														32(d)				
19Jul26Jul															33(b)			
26Jul2Aug																34(b)	35(c)	
2Aug9Aug																		36(c)
9Aug16Aug																		
16Aug23Aug																		
23Aug30Aug																		
30Aug6Sep																		
6Sep13Sep																		
13Sep20Sep																		
20Sep27Sep																		
27Sep4Oct																		
4Oct11Oct																		
11Oct18Oct																		
18Oct25Oct																		
25Oct1Nov																		
1Nov8Nov																		
8Nov15Nov																		
15Nov22Nov																		
22Nov29Nov																		
29Nov6Dec																		

Table 4.19: Weekly topic evolutions (part3).

3May10May																		
10May17May																		
17May24May																		
24May31May																		
31May7Jun																		
7Jun14Jun																		
14Jun21Jun																		
21Jun28Jun																		
28Jun5Jul																		
5Jul12Jul																		
12Jul19Jul																		
19Jul26Jul																		
26Jul2Aug																		
2Aug9Aug	37(a)																	
9Aug16Aug	37(a)	38(b)	39(f)															
16Aug23Aug				40(a)	41(a)	42(e)												
23Aug30Aug						42(e)	43(b)											
30Aug6Sep								44(b)	45(a)	46(f)								
6Sep13Sep											47(b)	48(f)	49(b)	50(b)	51(b)			
13Sep20Sep															51(b)	52(d)		
20Sep27Sep																	53(e)	54(c)
27Sep4Oct															51(b)			
4Oct11Oct																		
11Oct18Oct																		
18Oct25Oct																		
25Oct1Nov																		
1Nov8Nov																		
8Nov15Nov															51(b)			
15Nov22Nov																		
22Nov29Nov																		
29Nov6Dec																		

Table 4.20: Weekly topic evolutions (part4).

3May10May																		
10May17May																		
17May24May																		
24May31May																		
31May7Jun																		
7Jun14Jun																		
14Jun21Jun																		
21Jun28Jun																		
28Jun5Jul																		
5Jul12Jul																		
12Jul19Jul																		
19Jul26Jul																		
26Jul2Aug																		
2Aug9Aug																		
9Aug16Aug																		
16Aug23Aug																		
23Aug30Aug																		
30Aug6Sep																		
6Sep13Sep																		
13Sep20Sep																		
20Sep27Sep	55(a)	56(e)	57(e)															
27Sep4Oct				58(a)														
4Oct11Oct																		
11Oct18Oct					59(a)	60(e)	61(a)	62(b)	63(e)									
18Oct25Oct										64(e)	65(f)	66(b)	67(e)					
25Oct1Nov														68(f)	69(c)			
1Nov8Nov															69(c)	70(b)	71(b)	
8Nov15Nov															69(c)			72(b)
15Nov22Nov				58(a)														
22Nov29Nov																		
29Nov6Dec																		

Table 4.21: Weekly topic evolutions (part5).

3May10May									
10May17May									
17May24May									
24May31May									
31May7Jun									
7Jun14Jun									
14Jun21Jun									
21Jun28Jun									
28Jun5Jul									
5Jul12Jul									
12Jul19Jul									
19Jul26Jul									
26Jul2Aug									
2Aug9Aug									
9Aug16Aug									
16Aug23Aug									
23Aug30Aug									
30Aug6Sep									
6Sep13Sep									
13Sep20Sep									
20Sep27Sep									
27Sep4Oct									
4Oct11Oct									
11Oct18Oct									
18Oct25Oct									
25Oct1Nov									
1Nov8Nov									
8Nov15Nov	73(e)	74(e)	75(b)						
15Nov22Nov				76(b)					
22Nov29Nov									
29Nov6Dec							77(b)	78(e)	

Persistent topics in monthly topic evolutions:

A1: Popular post from Wang: Vanke outdoor sports can be sorted by: 1 long-distance running, 2 outdoor cycling, 3 soccer, 4 swimming, 5 basketball, 6 badminton, 7 tennis, 8 rowing, 9 table tennis, 10 climbing. Exercise is a good lifestyle choice.

A2: Popular post from Wang: "No bribe" is the bottom line of my life, citizenship, and entrepreneurship. 30 years later, with the unstable transformation in society, actually I did not expect "no bribery" to become a rarity, and I did not expect to "no bribery" to become a symbol of Vanke's unique reputation.

A3: Popular post from Wang: top and secondary cities' housing prices followed the same trend of the late 1980s Japan's bubble economy of high land prices and high house prices. But Japan's bubble burst. That's a warning alert! Vanke should: never be a 'land king', adhere to the mainstream housing, improve quality, and maintain cash flow.

A4: Popular post: (Wang is studying Jewish Studies and business ethics research in Cambridge and posted his admiration for Cambridge. But someone replied: Go and live in your favourite countries UK, USA and Germany. Do not make a lot of money in China but criticise China a lot.) Wang replied: Vanke, which is my business, is ranked ninth in private enterprises, and ranked second for taxes (2012 RMB 210 billion) . I'm still working hard, and I go abroad for further studies. Do not ask "what the government can do for me?" But ask "What can I do for the community?".

B1: News: Lawyer Jingsong LI challenged 29 listed real estate companies, accusing

Vanke and other companies of defaulting on land value-added tax, with the total amount of at least 64 billion Yuan.

B7 (related to B1): News: According to CCTV "Weekly Quality Report", by long-term focusing on China's real estate development, Jinsong LI, who is a Beijing professional lawyer, certified public accountant and registered tax consultant, pointed out SOHO China and Vanke may owe 3.8 trillion LAT (land value added tax) via CCTV (share from @ Phoenix Finance).

E12 (related to B1): Popular post: @ REN (CEO of Huayuan Estate): the announcements from all companies have already directly proved the ridiculousness of CCTV. @ Real Estate News: [Wang: Vanke is not a case of the presence of non-payment of land tax].

B3: News: In the first half of this year, the performance of the real estate has attracted much attention. According to CRIC latest release: "the first half of 2013 China's real estate business sales TOP50 rankings", Vanke and Hengda are top 1 in sales value and area sold area respectively. Sales for Lvdi, Zhonghai, Baoli were similar to last year.

B5: News: By the closing on September 3rd, Tencent's share price closed at 379.8 Hong Kong dollars, Tencent's market value exceeds 700 billion Hong Kong dollars. The amount is equivalent to 5 times Vanke.

C1: Advertisement: Free pick-up service for showings: Vanke provides a convenient, free shuttle for real estate showings (Vanke Egrets County).

C2: Advertisement: Unprecedented good source of housing(Vanke Zitai): two metro stations nearby. Vanke gives good-quality and tailored design service.

D1: Promotion: Please join "I love my family relay race": re-post to join, win gifts (a promotion held by Vanke official account: if the followers re-post this promotion, there would be a chance to win gifts).

E4: Popular post: The rise of real estate demolished historical sites: Huayuan Estate (CEO Zhiqiang REN) removed 45 other large courtyards; and SOHO (CEO Shiyi PAN) demolished historical alleys. Huarun Estate demolished the former residence of LIANG Sicheng (Chinese architect). CITIC(Zhongxin) demolished Xuannan area. Vanke's (Chairman Shi WANG) also made high buildings surrounding Nanjing Mochou Lake.

E6: Popular post: (about Vanke campus recruitment) Getting married and finding a job sometimes depend on fate: Feng LIU, Vanke HR Manager in Hangzhou: My current wife (former girlfriend) and I once returned to school. We were very happy, and I proposed on that night successfully. Although we studied hard at university, we still need luck to get a good degree and find a job. It's important to have "chemistry" between you and your company, just like love.

E7: Popular post: The vast majority of high-rise buildings were built in the last decade intensively. Using the same stone material and construction process, their life expectancy would be roughly the same. Imagine after a few years, these high buildings could deteriorate roughly in the same period. What would it be like? I have sought the advice from a vice president of Vanke, and his answer was: "do not consider the consequence" is one of China's current development models.

E10: Popular post: In 2012, the state put more effort on regulating and controlling real estate and construction industry, a lot of real estate and construction industries

were eliminated. The remaining enterprises became better regulated. For example, in the survey, East Lake, China Vanke and other companies strengthened their HR management. Therefore, HR effectiveness and competence in all aspects has been increasing.

4.8 Summary

In this chapter, we have presented a comprehensive literature review for the development of topic models. Latent semantic analysis (LSA), which can be realised by Singular Value Decomposition (SVD) or Non-negative Matrix Factorisation (NMF), is introduced in detail for acquiring similarity and knowledge representation from textual content.

Latent Dirichlet Allocation (LDA) was developed from and went beyond the integration of LSA and probabilistic models by changing the constitution of the topic mixture weight and adding Dirichlet priors. We applied Gibbs sampling to learn topics from LDA for algorithm implementation.

It has been shown that there is no guarantee that all of the topics generated by LDA represent coherent subjects. The rest of the chapter chiefly focuses on inventing an algorithm to choose and retain more stable and well-explained topics. The full description of the Randomness Reduction algorithm, together with empirical examples at both word and topic levels, clarified the process thoroughly. Case studies and the evaluations for the significance of the algorithm have been carried out. Results indicate that the Randomness Reduction is a promising approach for filtering out insignificant topics. In the end, topic classification and evolution

analyses were conducted to understand the dynamics of topics and describe the changes of topics over time.

Chapter 5

Conclusion, Discussion and Future Direction

This chapter summarises the main results of the preceding chapters, provides discussions on the importance of empirical findings, and outlines some preliminary ideas for future research.

The key motivation for this research, as stated in the introduction, is to investigate the statistical learning approaches to extract the comprehensive information from a large amount of micro-blog posts. This thesis has attempted to consolidate and extend earlier research in micro-blog data mining, sentiment time series modelling and topic models post-processing. Our research can also be used as an awareness indicator of Weibo data and a successful guide to applying multiple statistical learning methods on Micro-blog's data.

Several approaches for exploratory data mining on Weibo data have been provided in Chapter 2. After reviewing previous Twitter and Weibo data analysis literature, we explored the pattern of posting time and its relationship with share price on self-collected Weibo data. Term frequency charts and cluster analysis have been introduced to obtain an initial understanding about what individuals are posting

about.

In the term frequency charts of Chapter 2, we can find several common words for closely linked objects (companies), and network analysis with directed paths can be a potential research subject. For instance, when individuals talked about Suning, they also talked about Guomei; but when they talked about Guomei, they did not mention Suning. Both of them also included Jingdong (their competitors), online shopping, electronic appliances, etc. According to the term frequency charts, a network graph can be developed and the relationship can be studied between different nodes.

In Chapter 3, the primary focus has been on sentiment time series modelling. By extending the Box-Jenkins method, we built up a general framework for time series fitting and employed it on Weibo's proportional positive and negative sentiment. The fitted models were diagnosed using residual analysis and compared by cross-validation. Construction of a multivariate sentiment time series model also helped to find out cross correlations between positive and negative sentiments.

In the literature, we can hardly find any previous extensive studies for modelling sentiment time series. All the detailed procedures with instructions involved with our sentiment time series modelling are readily applicable to other time series. In our analysis, we adjusted and eliminated the biggest spike of negative time series, but the behaviour of these huge spikes can be investigated and modelled separately. Noting the significance of bivariate sentiment time series modelling, further research can also attempt to build multivariate models for all seven sentiments to find out their potential correlations.

In addition, to using the informational content of the sentiment time series, we also attempted to build up a joint model between sentiments and stock index. A multivariate regression model having standardised proportional positive and negative sentiments as regressors and log returns of the stock price as the dependent variable has been built. In the model without any lags, we found the coefficients of the regression for both negative and positive sentiments are negative, which is hard to interpret and potentially indicating that on the same day the extreme emotions have negative effects on the return. When we added one-day lag, the effect of positive sentiment became positive. This may imply that when the amount of yesterday's posts containing positive sentiments increases or negative sentiments decreases, we can see a raise in today's return. However, all the multivariate regressions with or without lags are not quite significant (p-values between 0.1 and 0.3) and with very low adjusted R^2 value.

Comparing with findings from recent similar researches by Giachanou and Crestani (2016), Zimbra et al. (2016) and Giachanou and Crestani (2016), different transformations and dynamic regression models can be further applied to detect the predictive power of sentiment indexes; an overall Impact Factor determined by the number of followers can be used as a multiplier on sentiment to include the effects of popularity; and different approaches can be used to measure sentiment apart from lexicon based methods, such as employing the dynamic architecture for Artificial Neural Networks suggested by Zimbra et al. (2016).

Furthermore, we may attempt some other methods for the fitting of proportional negative sentiment. It is possible that the dynamics of peak points for the negative sentiment time series are different from the other points. The threshold

autoregressive (TAR) model (Tsay, 1989) is an applicable model which assumes the process may behave differently when the values of a variable exceed a threshold value. We employed a two-regime TAR model (Tong, 1990) to fit the demeaned proportional negative sentiment (Chan et al., 2012). The estimated threshold is 0.14 (the 90 percentile of all data) from a minimum AIC fitting with thresholds searched from the 10 percentile to the 90 percentile of all data. It separates the data into lower regime with 196 data points and upper regime with 21 data points. Result shows that the AR(1) components for both lower regime and upper regime are significant and the estimated AR coefficient for upper regime are quite different from lower regime's one, with values 0.5 and 0.2 respectively.

In addition, the time series plot in Figure 3.3.12 may show a potential change point around the 80th day: the volatility after the 80th day stayed at a much higher level than before. If we can identify the change point via various change point detection method, e.g. Wild Binary Segmentation (Fryzlewicz et al., 2014), it might be more appropriate to fit different models before and after the change point. The only difficulty might be we have not enough data point for change point detection - in this case, it might be applicable to look into hourly data instead and further find a best fitted model for hourly sentiment.

The major contribution for Chapter 4 is to improve topic models post-processing by filtering out insignificant topics and making the output of topic models more comprehensible. Based on the results after the Randomness Reduction, we can generate the evolution of topics and overview the topic dynamics.

Recent research regarding assessing topics in tweets is still with mostly conventional metrics: measuring semantic similarity or conducting statistical analysis to

calculate the distance between term distributions (Fang et al., 2016b), which are not particularly useful for the context of micro-blogs. Applying Word Embedding can be an alternative to evaluate the coherence of topics from Twitter (Fang et al., 2016a). We defined a search-back rule to find out corresponding news or popular posts for significant topics, which is more relevant to micro-blog research.

Collectively our analyses have endeavoured to "bridge" the topic modelling on micro-blog data and the topic words post-processing algorithm named Randomness Reduction. There are plenty of topic analyses on Twitter, but the researchers seldom used any topic post-possessing methods to filter out the redundant topics before classifying the topics. Due to this, most topic analyses for micro-blogs only cover very general topic categories, e.g. politics, economics, or sports (Cho et al., 2016). We address the issue of insignificant topics by narrowing down the topic numbers via the Randomness Reduction.

It is of interest to calibrate the Randomness Reduction algorithm and see how it works for results generated from other large amounts of textual content. We have further applied the algorithm on Twitter datasets related to company Apple and multiple US Airlines. Results for Randomness Reduction on both Weibo and Twitter data were very significant: the algorithm increased the rate of significant topics from 21.4% to 75% (see Section 4.5), from 36.7% to 64.6% (see Section 4.6.1) and from 33% to 76.2% (See Section 4.6.2) respectively. There is considerable scope for further adopting the Randomness Reduction to topic words generated from other collections of documents, e.g. news, mails, papers, etc. An R package can possibly be built for other users to apply the Randomness Reduction for post-processing the results from topic models and filtering out insignificant topics.

Appendix

Appendix A

Appendix for Chapter 2

A.1 Capturing Data via API

Sina Open platform provides several official Software Development Kits (SDKs) for API access. They are typically sets of software development tools that allow for the creation of applications for some software packages, software frameworks, hardware platforms, operating systems, or similar platforms. Applying SDKs via API is a common way to capture data from micro-blogging platforms. The SDKs which Sina Website provides official support for are Java SDK, PHP SDK, Flash SDK, Javascript SDK (JS SDK), C++, Android SDK, and WP7 SDK.

There is an unofficial SDK for R, named Rweibo, which can be applied as an R package. It was explored on R-Forge and could be found at

`http://r-forge.r-project.org/projects/rweibo/`

The key method for Sina Weibo's authority mechanism is OAuth. In order to use API, the first step is to register an application. After filling in several forms related to a user's account, an App Key and an App Secret will be given. With these request tokens, a developer will be able to be redirected to an authorization page. By logging in with the account name and password, an access token will

be called back. Using this access token, the authorized data are available for data analysis. The detailed information can be found at Sina's OAuth page:

`http://open.weibo.com/wiki/Oauth`

This connection for authorisation, designed for Rweibo, is realised by RCurl, which is a package allowing the user to compose general HTTP requests, providing convenient functions to fetch URLs, getting post forms, and processing the results returned by the web server.

Sina Weibo updated its authority mechanism from OAuth1.0 to OAuth2.0 on 15th of October 2012. The new version (V2) interface, compared with older version (V1), is more functional, more efficient, more standardized, and more controllable. Adding more interfaces, such as the short link interface, the location service interface, and the application interface, better meets the needs of applications developers. Meanwhile, to control the access of information, implement and ensure information security and prevent itself from losing exclusive information, Sina Weibo closed its search interface. It means now it is not possible to get a large amount of data from a simple search function via Sina API, which is still available for Twitter API.

In conclusion, data mining using search function via Sina API appeared to be impossible. As a result, it is much more difficult for Weibo to get data than Twitter. This is the main difference for data collection between Weibo data mining and Twitter data mining.

A.2 Details for Web Crawling

Web Crawling or Spidering is another possible technique for capturing Weibo posts. Many sites, especially search engines, use spidering as a means of providing up-to-date data. Web crawlers are mainly used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages to provide fast searches. Crawlers can also be used for automating maintenance tasks on a website, such as checking links or validating HTML code.

The basic idea of this script is to imitate a search process: first log on with username and password, type in keyword, open search result page, flip the pages automatically, parse the content of pages, extract the information of posts and finally write them into csv file. The incremental crawler process is realised by comparing the time of older posts and new posts.

A Python Web Crawling tool was employed for several weeks to obtain data, but errors occurred too frequently due to Sina's supervisory control. After several weeks' trials, this method had to be abandoned.

A.3 Details for Web Parsing

The author of Rweibo developed the web search interface *web.search.content* to parse web content via XML without the establishment of OAuth2 objects. The advantage is, it does not require API access, but the disadvantage is, it is not stable. In the updated Rweibo 0.1-6 version in April 2013, the author added more properties for *web.search.content*, defining search range using "since" and

"sinceID", downloading incremental posts using "Combinewith" , getting more entries of "the number of times this post is forwarded" and "the number of comments received". All data downloaded for our research are in csv format.

Before April 2013, the method applied to collect data was mainly to manually download them once or twice every day at a certain time. The first chosen company was $Huawei$, and data were downloaded manually (twice a day) for about four months from January 2013 to April 2013. But later on, it was found that the stock market listed $Huawei$ was not the company we were downloading data from. Therefore, we needed to start from the beginning: around 40 days' (from 18th March to 26th April) data with keyword $Suning$, $Guomei$, $Vanke$ and $Haier$ were collected. To ensure no posts were missed out, for the first three keywords, 40 pages posts were downloaded twice a day, in the late morning and late at night. For $Haier$, as the amount of daily posts is around five hundred, data were collected only once a day. Then these posts were combined and deduplicated using R codes to make them continuous.

The creation of the incremental feature by "Combinewith" is a great breakthrough. Therefore, we do not need to manually combine and deduplicate after data collection and it enables the running loop for continuous data. The only thing that we need to be aware of is that the incremental number of pages must be less than 40. This means we must run this loop by a certain interval to make sure the amount of posts created is less than 800 (40*20) within this time interval. Empirically, the data can be downloaded using the same IP every 30 minutes (1800 seconds) for 40 pages due to the limit set by Sina. In the first several months, gaps occurred several times as the Windows system crashed and showed Blue Screen.

Later on, we use the Linux system instead of Windows, which is much more stable for data downloading, and gaps seldom occurred after the change.

A.4 Additional Results for the Analysis of the Time of the Posts

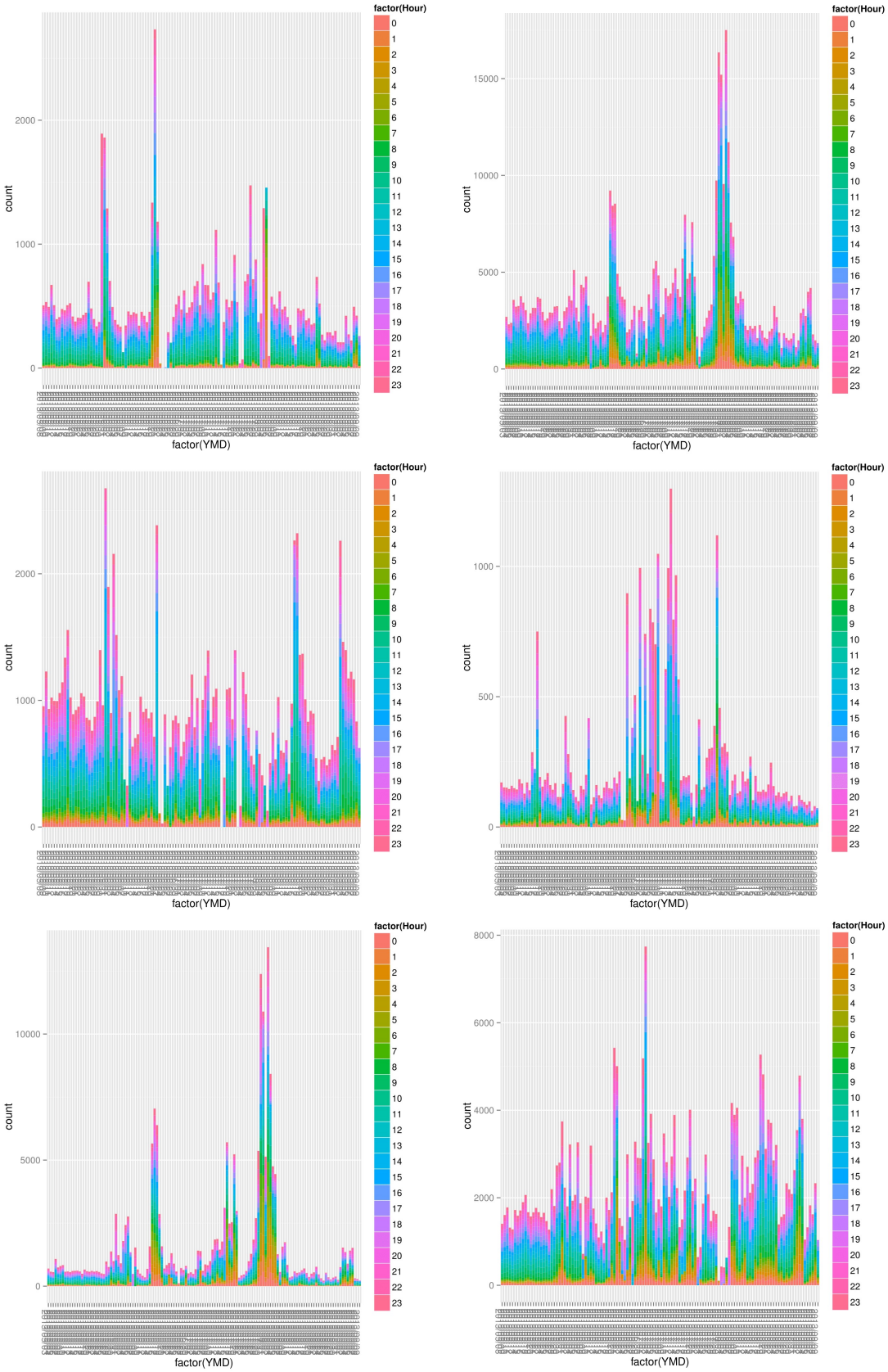

Figure A.4.1: Daily post amount, coloured by time (Multiple companies: Biguiyuan, Biyadi, Maotai, Donghang, Guomei, and Suning).

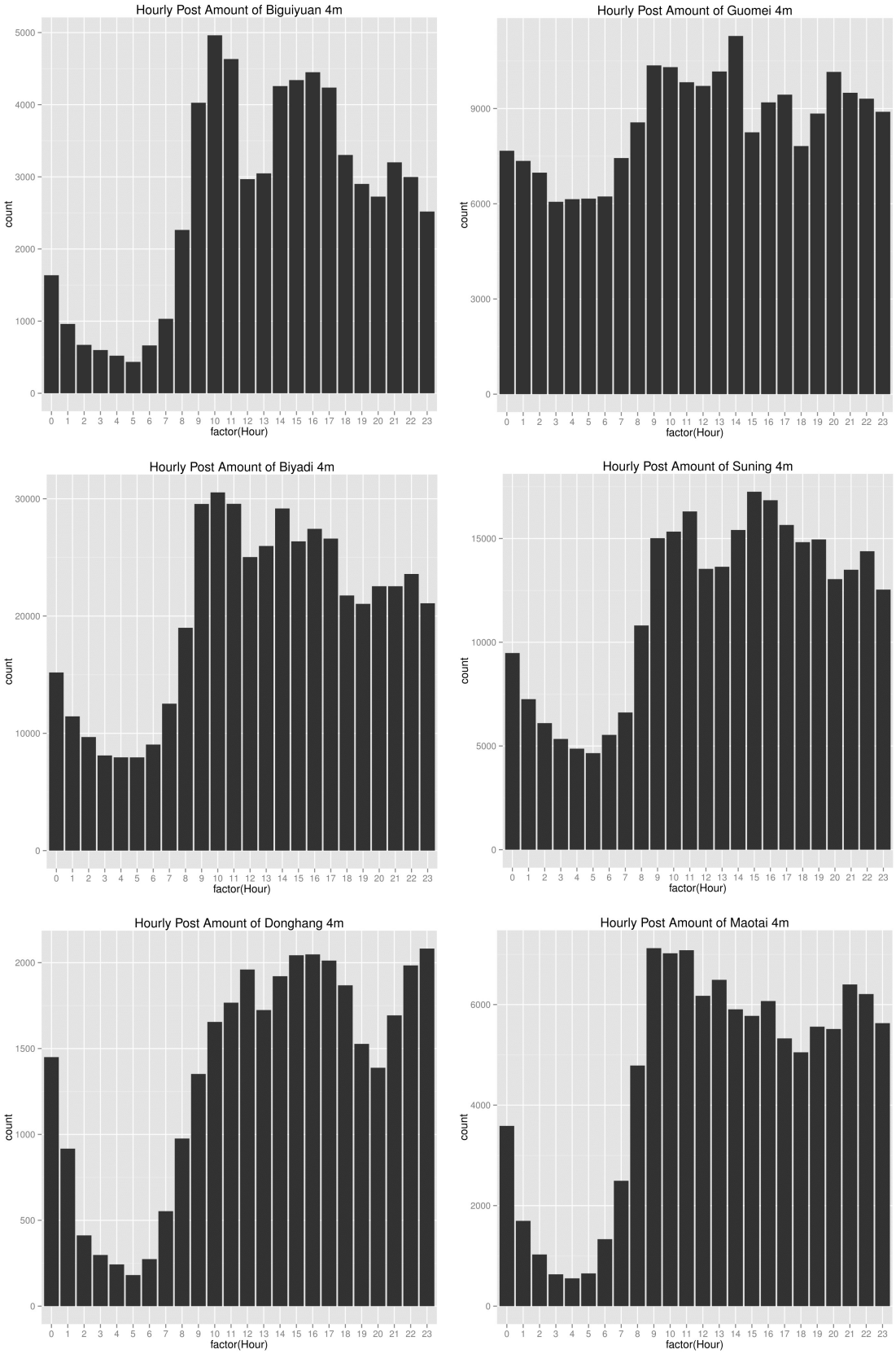

Figure A.4.2: Hourly post amount (Multiple companies: Biguiyuan, Biyadi, Maotai, Donghang, Guomei, and Suning).

A.5 Result of Linear Regression for Vanke

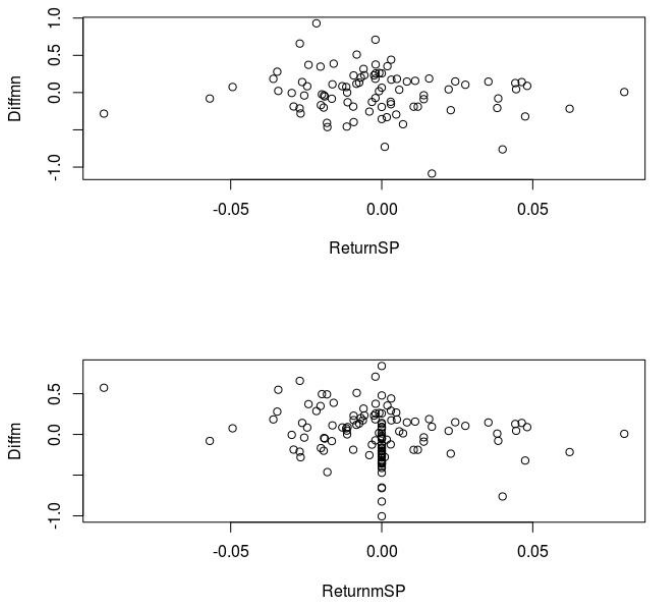

Figure A.5.1: Dot plot for log returns of share price and log returns of post amount of Vanke. Group A: R_t vs D'_t; Group B: R'_t vs D_t.

```
> lrn<- lm(ReturnSP~0+ Diffmn)
lrn<- lm(ReturnSP~0+ Diffmn)
> summary(lrn)
summary(lrn)

Call:
lm(formula = ReturnSP ~ 0 + Diffmn)

Residuals:
      Min        1Q    Median        3Q       Max
-0.094740 -0.017331 -0.002922  0.007425  0.080373

Coefficients:
        Estimate Std. Error t value Pr(>|t|)
Diffmn -0.009771   0.009014  -1.084    0.281

Residual standard error: 0.02605 on 88 degrees of freedom
Multiple R-squared:  0.01318,   Adjusted R-squared:  0.001963
F-statistic: 1.175 on 1 and 88 DF,  p-value: 0.2813
```

Figure A.5.2: Result of linear regression for Vanke (Group A: R_t and D'_t).

The fitted models can be expressed as:

```
> lr<- lm(ReturnmSP~0+ Diffm)
lr<- lm(ReturnmSP~0+ Diffm)
> summary(lr)
summary(lr)

Call:
lm(formula = ReturnmSP ~ 0 + Diffm)

Residuals:
      Min        1Q    Median        3Q       Max
-0.083461 -0.010770 -0.002991  0.006928  0.080413

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
Diffm -0.014856   0.006376   -2.33   0.0214 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02154 on 125 degrees of freedom
Multiple R-squared:  0.04162,   Adjusted R-squared:  0.03396
F-statistic: 5.429 on 1 and 125 DF,  p-value: 0.02141
```

Figure A.5.3: Result of linear regression for Vanke (Group B: R'_t and D_t).

Group A: $R_t = -0.01D'_t + \epsilon_t$

Group B: $R'_t = -0.015D_t + \epsilon_t$

The coefficient seems to be not significant and the R-squared score is low. Group B's results give a coefficient of -0.015 and the p-value is 0.0214, which shows that the coefficient of the variable becomes significant. But this significance might be due to the repetition of Friday's share price. The residuals of these two regressions appear to be stationary from the test.

A.6 Additional Results for Intensity of the Posts vs Share Price

A.6.1 Time Series Figures for Company Guomei

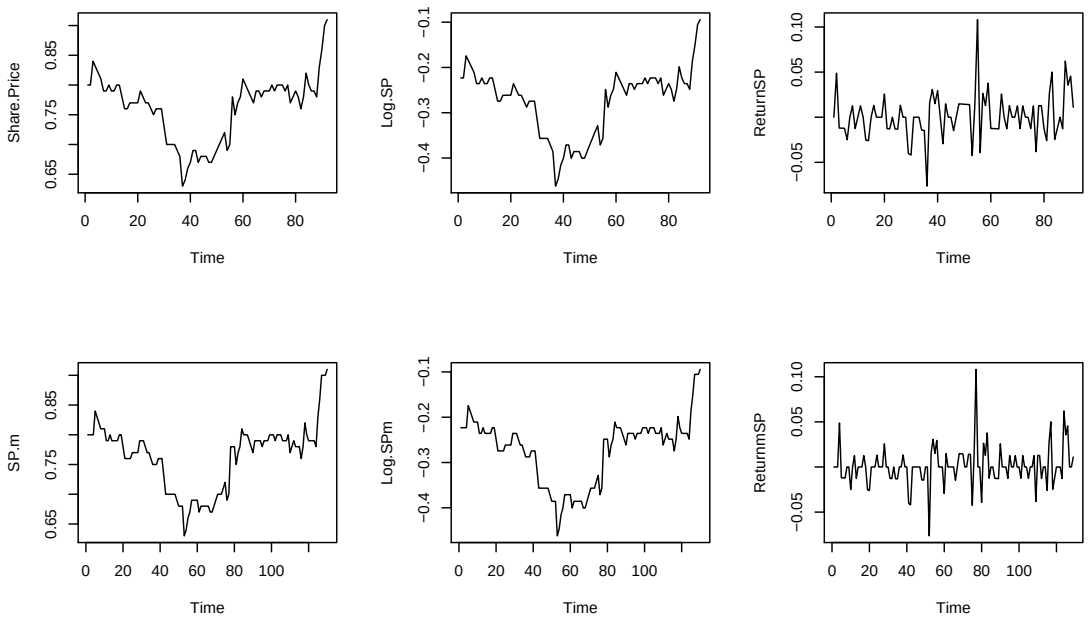

Figure A.6.1: Time series for share price of Guomei (First row, from left to right: P_t, $\log(P_t)$, R_t. Second row, from left to right: P_t', $\log(P_t')$, R_t').

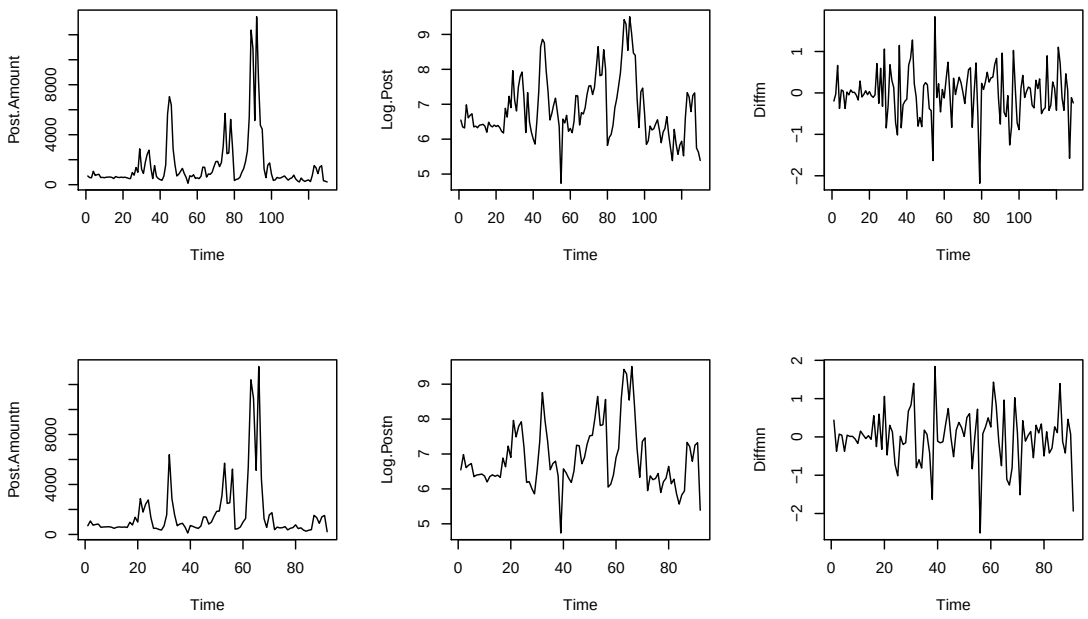

Figure A.6.2: time series for post amount of Guomei (First row, from left to right: A_t, $\log(A_t)$, D_t. Second row, from left to right: A'_t, $\log(A'_t)$, D'_t).

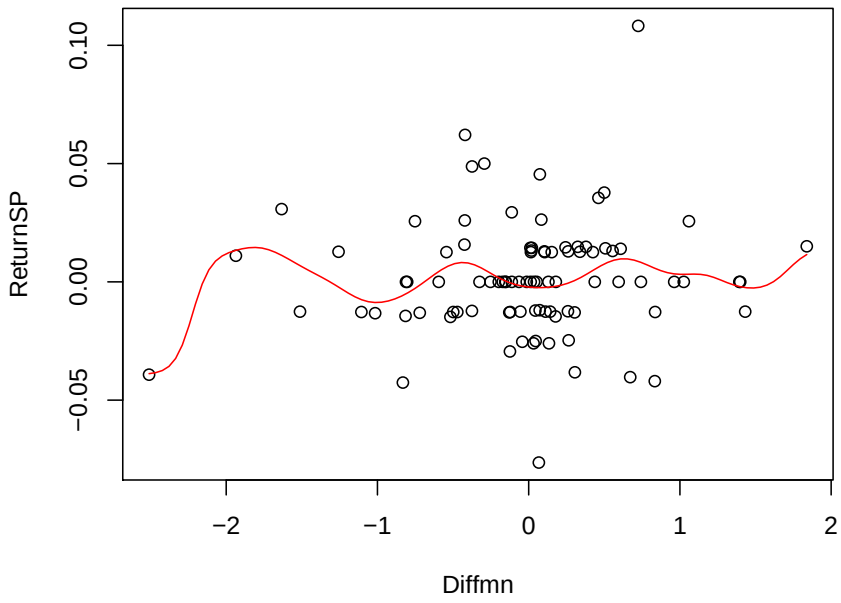

Figure A.6.3: Regression estimate using local polynomials with Bandwith 0.5 for Guomei (R_t and D'_t).

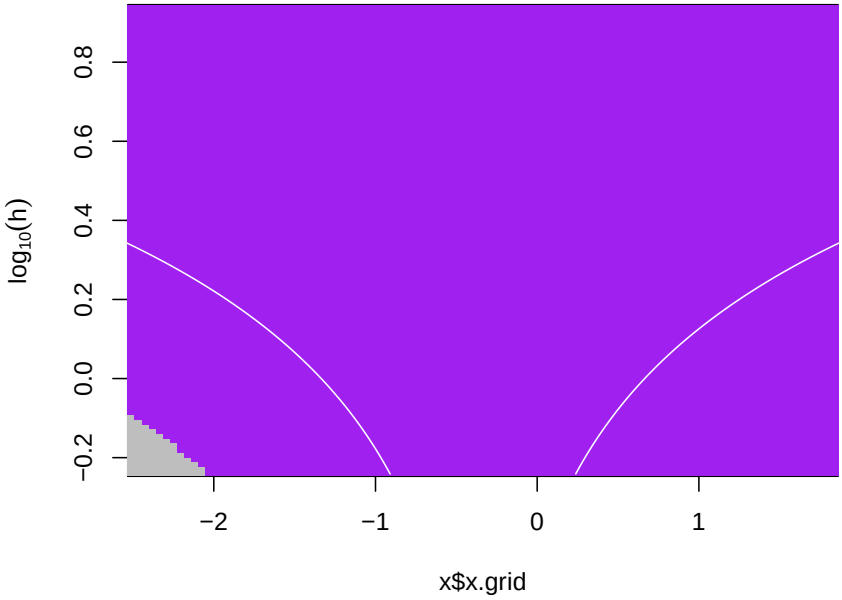

Figure A.6.4: SiZer plot for the first derivative for Guomei (R_t and D'_t).

A.6.2 Time Series Figures for Company Donghang

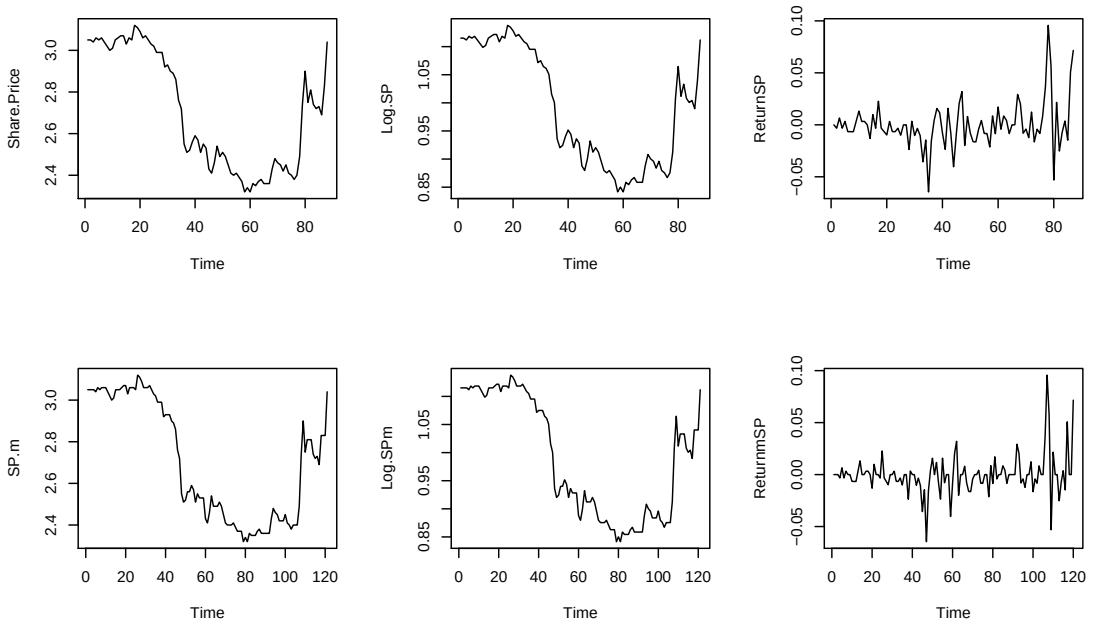

Figure A.6.5: Time series for share price of Donghang (First row, from left to right: P_t, $\log(P_t)$, R_t. Second row, from left to right: P'_t, $\log(P'_t)$, R'_t).

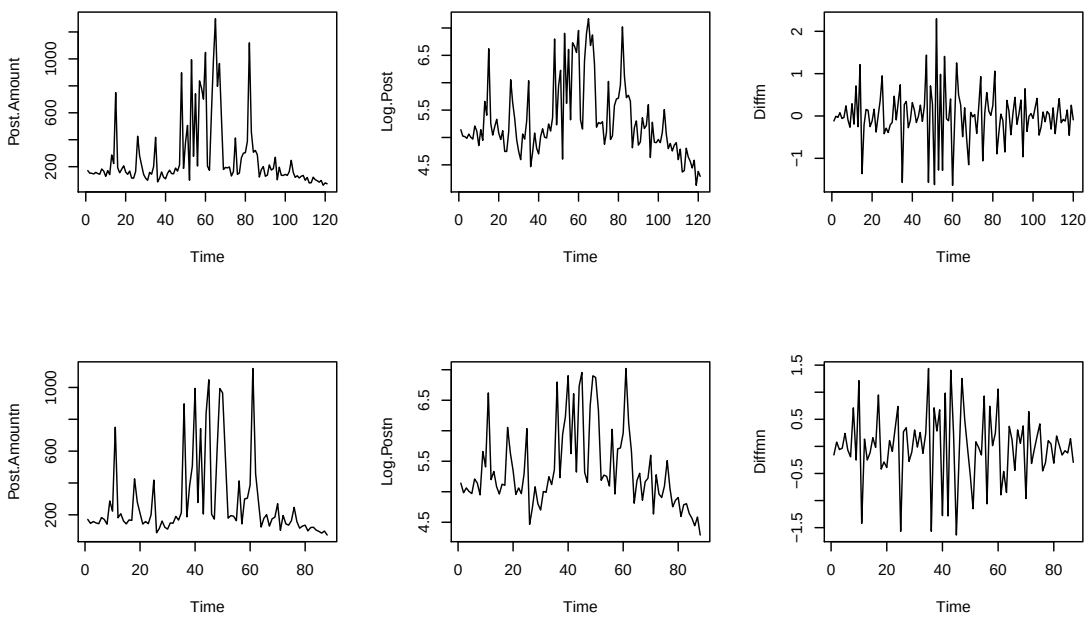

Figure A.6.6: Time series for post amount of Donghang (First row, from left to right: A_t, $\log(A_t)$, D_t. Second row, from left to right: A'_t, $\log(A'_t)$, D'_t).

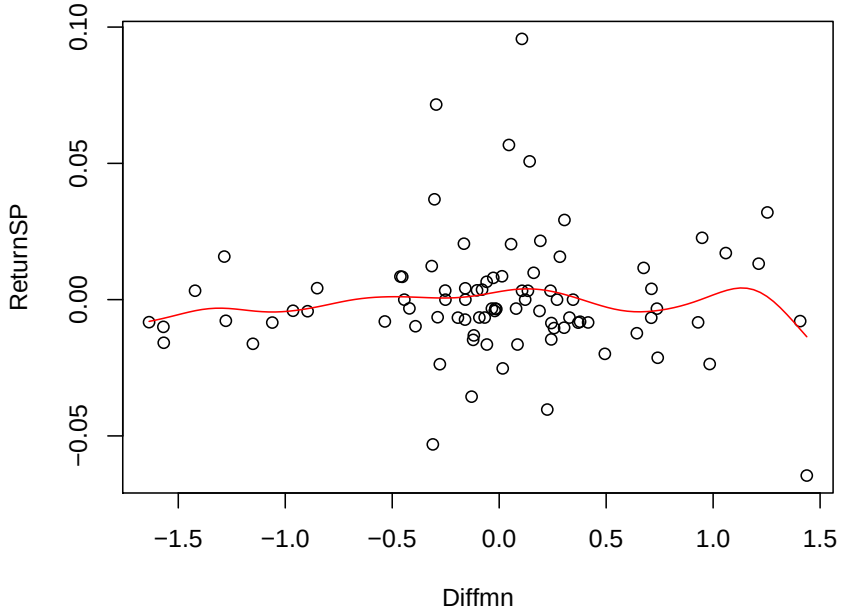

Figure A.6.7: Regression estimate using local polynomials with Bandwith 0.5 for Donghang (R_t and D'_t).

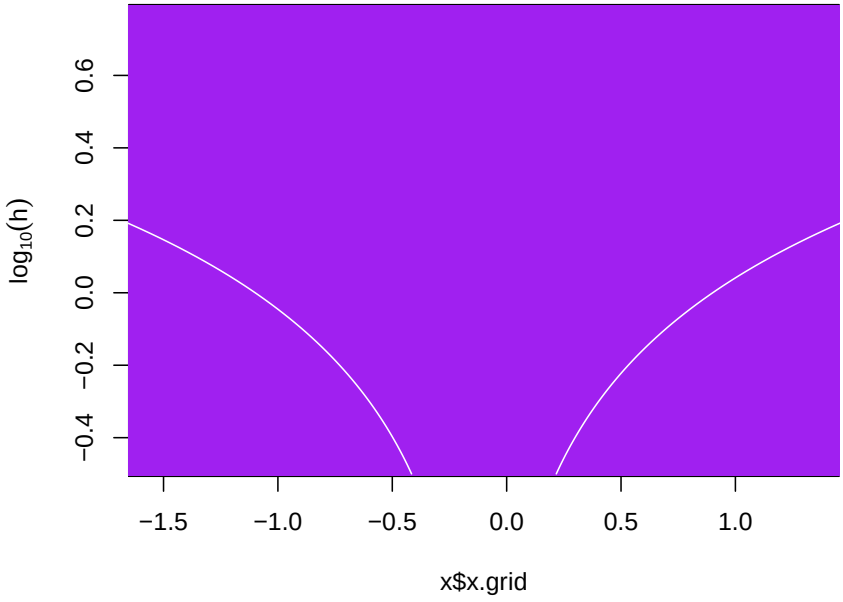

Figure A.6.8: SiZer plot for the first derivative for Donghang (R_t and D'_t).

A.6.3 Time Series Figures for Company Biyadi

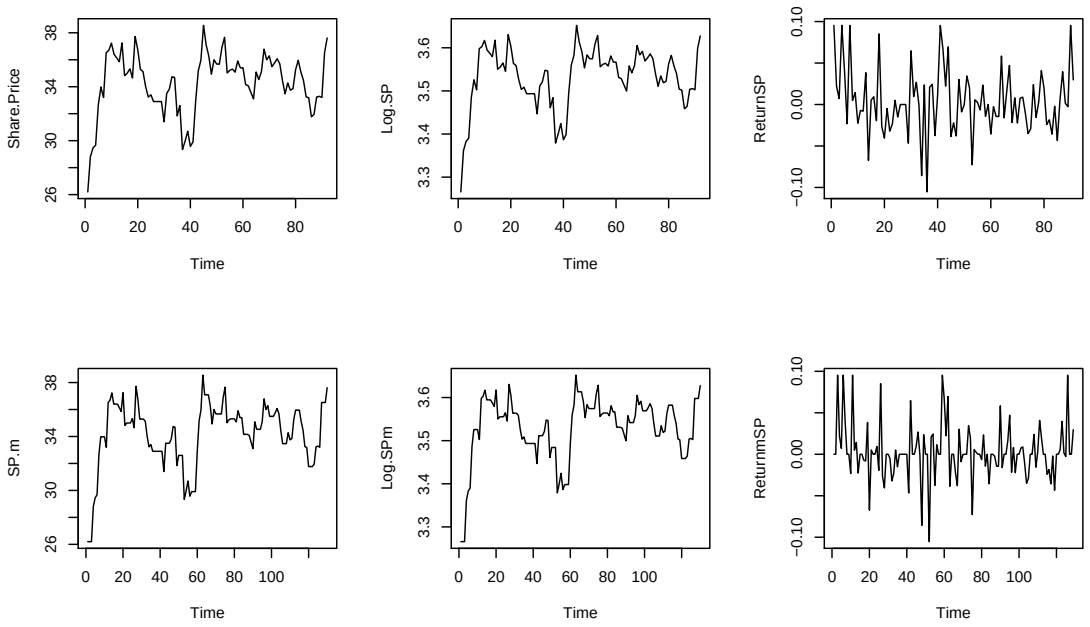

Figure A.6.9: Time series for share price of Biyadi (First row, from left to right: P_t, $\log(P_t)$, R_t. Second row, from left to right: P'_t, $\log(P'_t)$, R'_t).

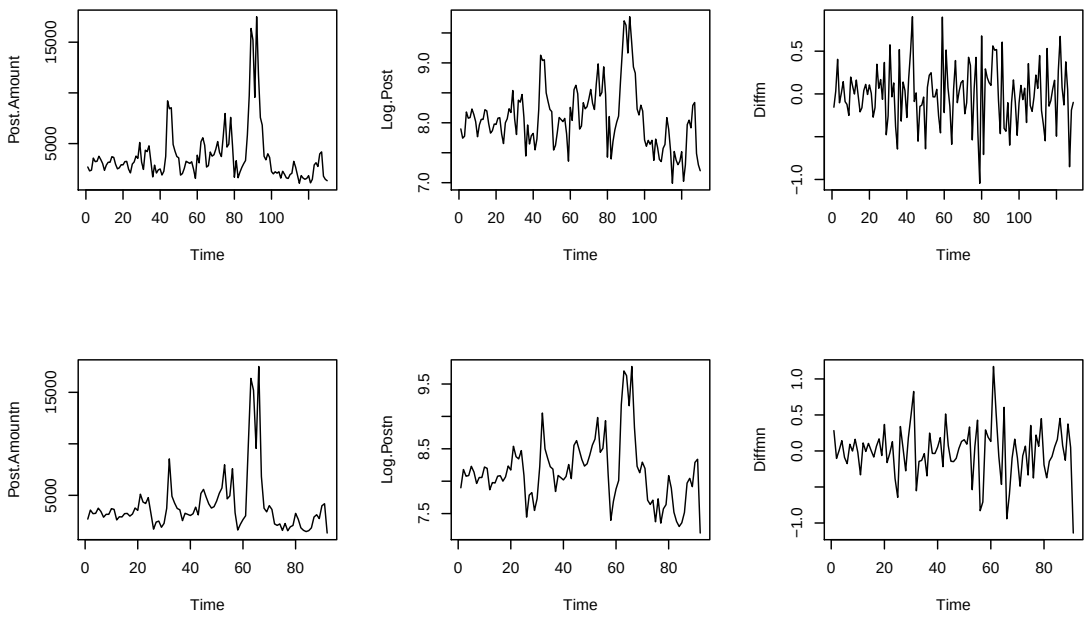

Figure A.6.10: Time series for post amount of Biyadi (First Row: A_t, $\log(A_t)$, D_t; Second Row: A'_t, $\log(A'_t)$, D'_t).

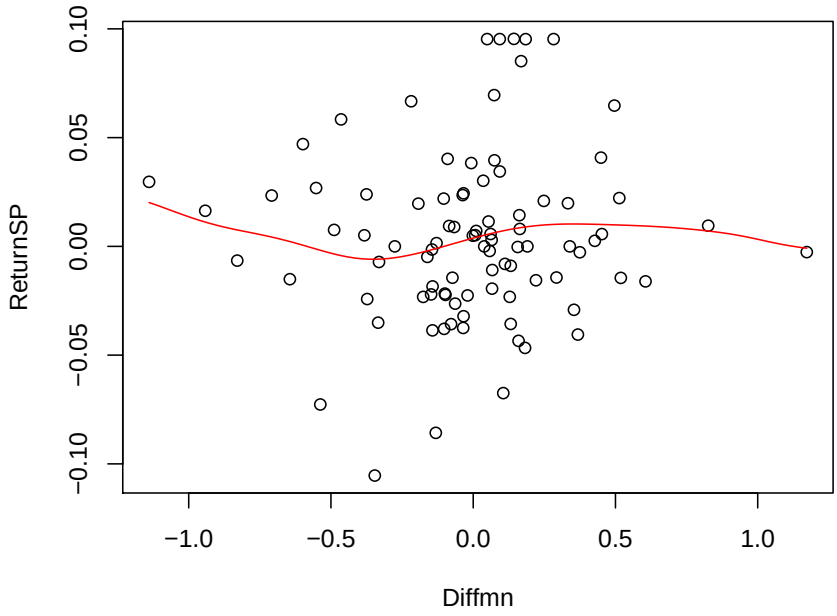

Figure A.6.11: Regression estimate using local polynomials with Bandwith 0.5 for Biyadi (R_t and D'_t).

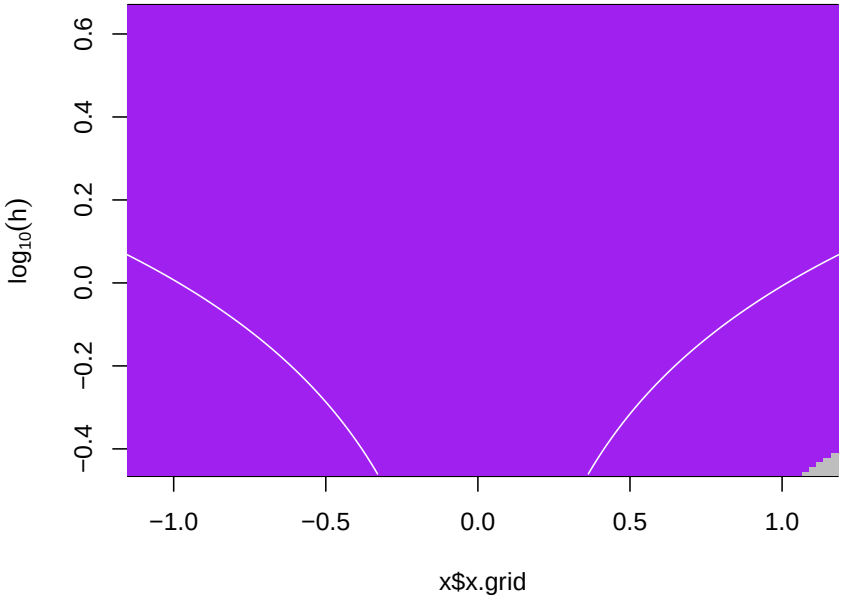

Figure A.6.12: SiZer plot for the first derivative for Biyadi (R_t and D'_t).

A.7 Word Cloud

Figure A.7.1: Word cloud for Biyadi.

A.8 Additional Figures for Cluster Analysis

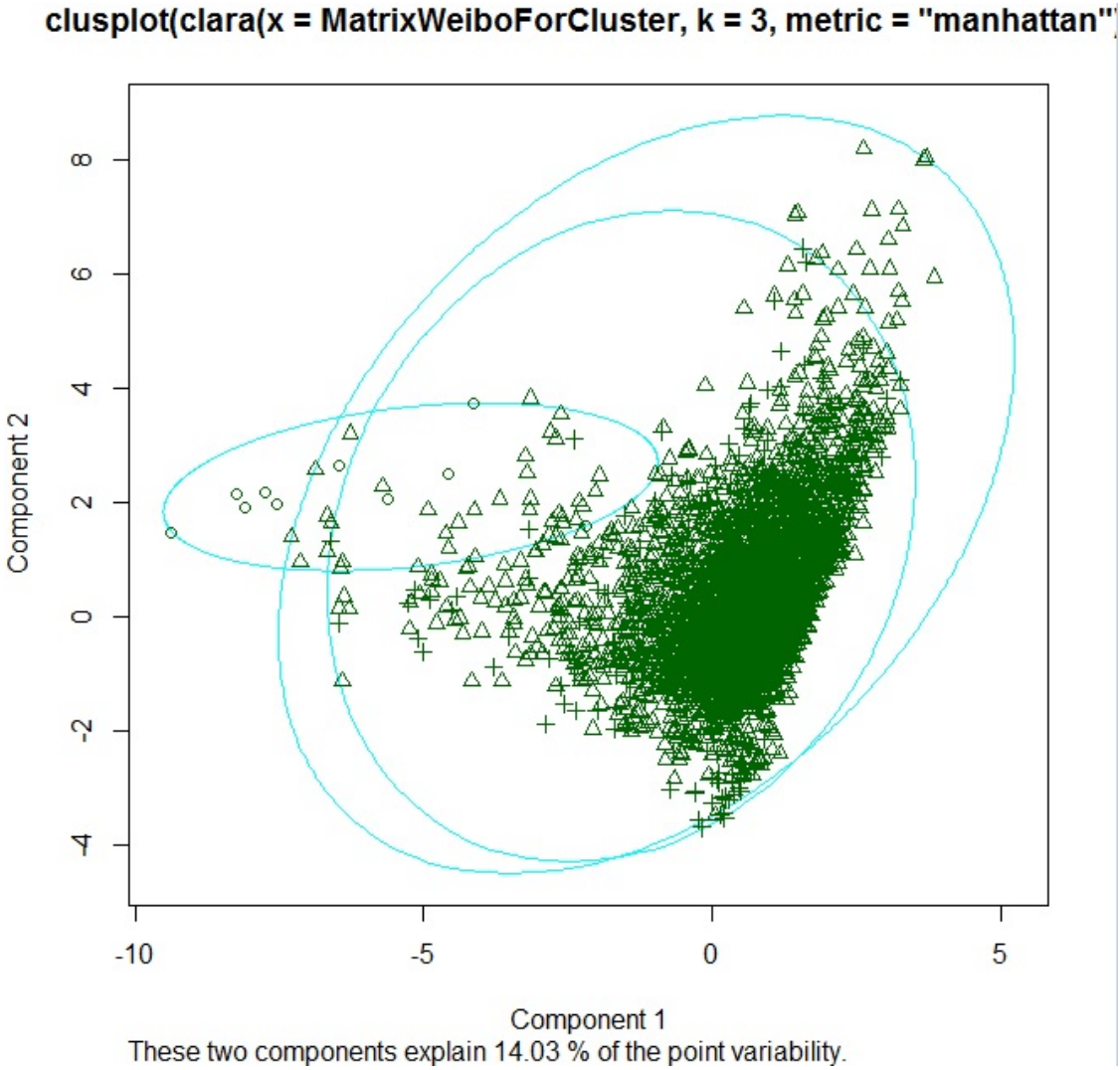

Figure A.8.1: Cluster plot from CLARA. Visualised by two leading principal components. Number of clusters k = 3. Sparsity = 0.95. Contains 49 terms.

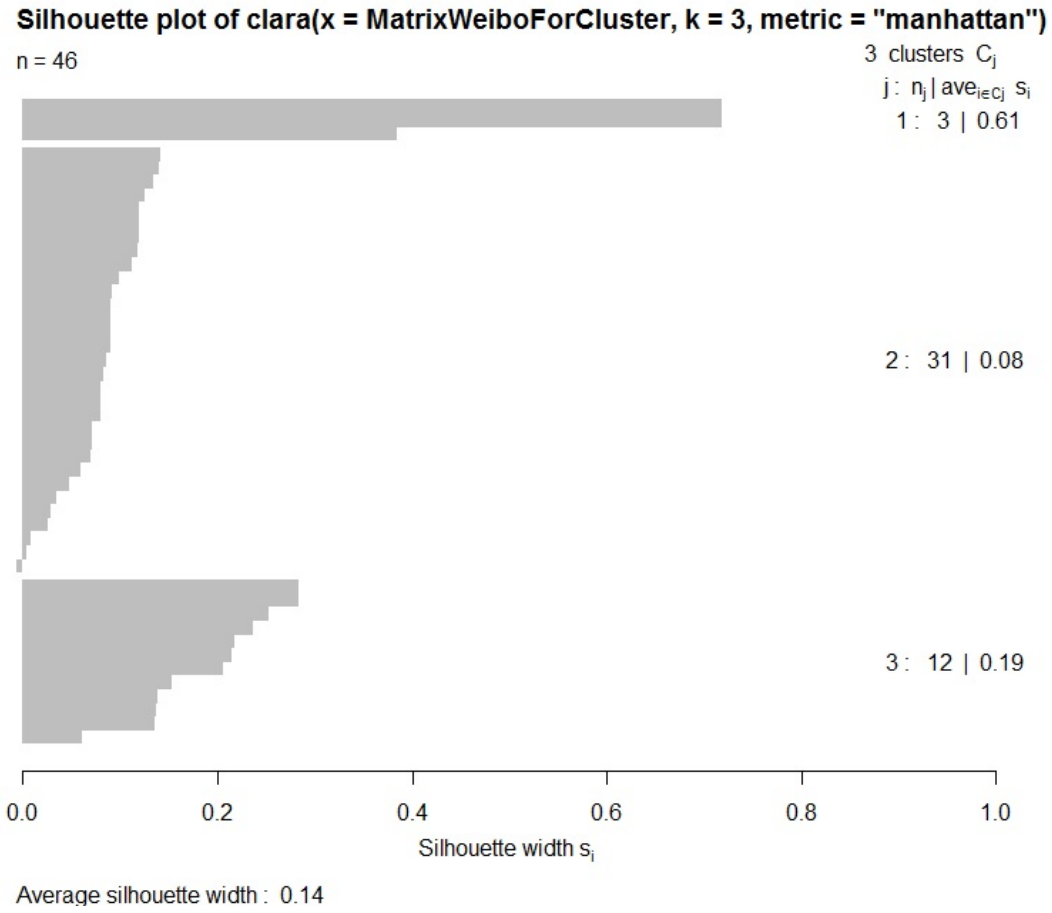

Figure A.8.2: Silhouette plot for CLARA. Sparsity = 0.95.

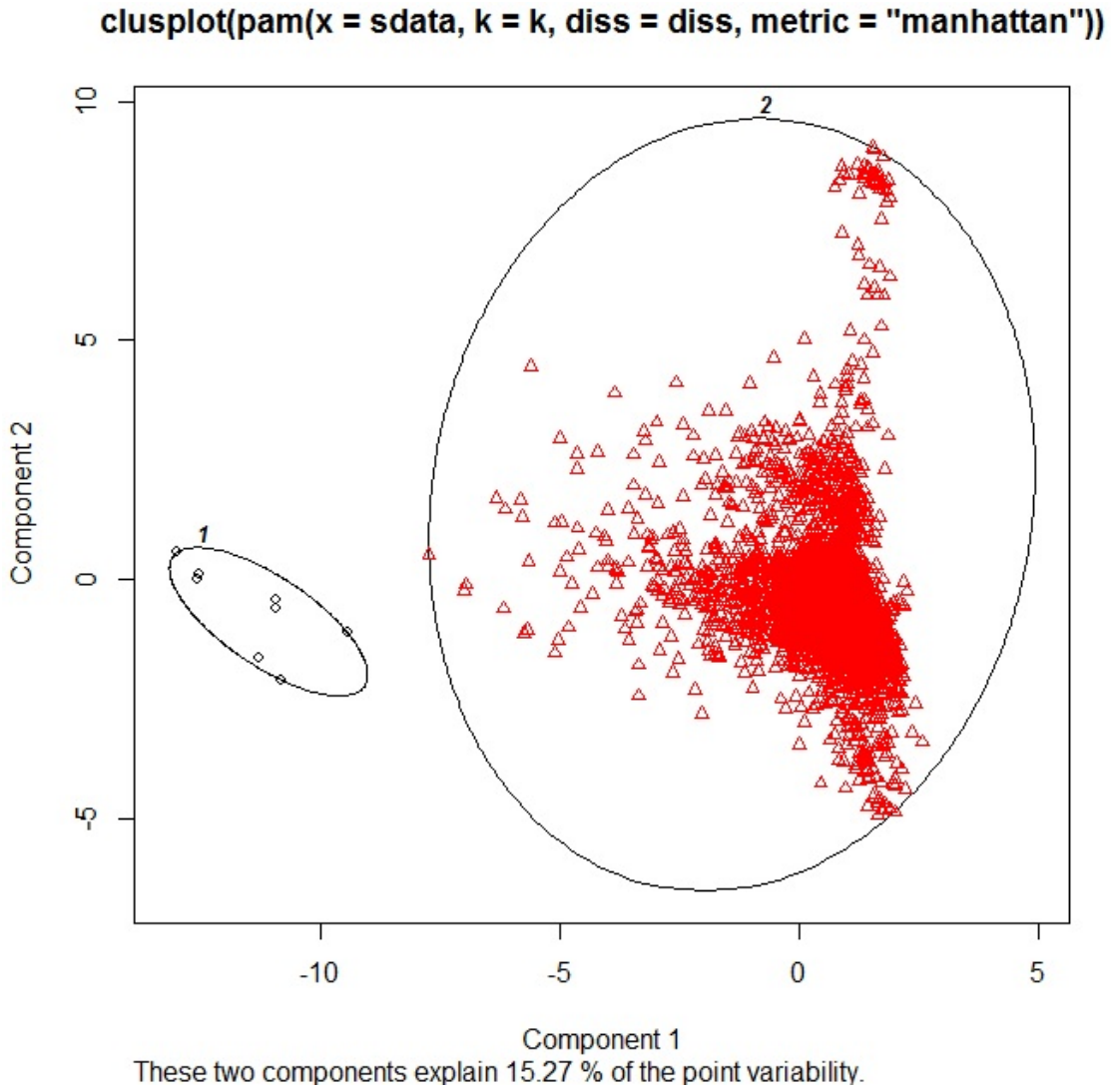

Figure A.8.3: Cluster plot from PAM. Visualised by two leading principal components. Number of clusters k = 2. Sparsity = 0.96. Contains 70 terms.

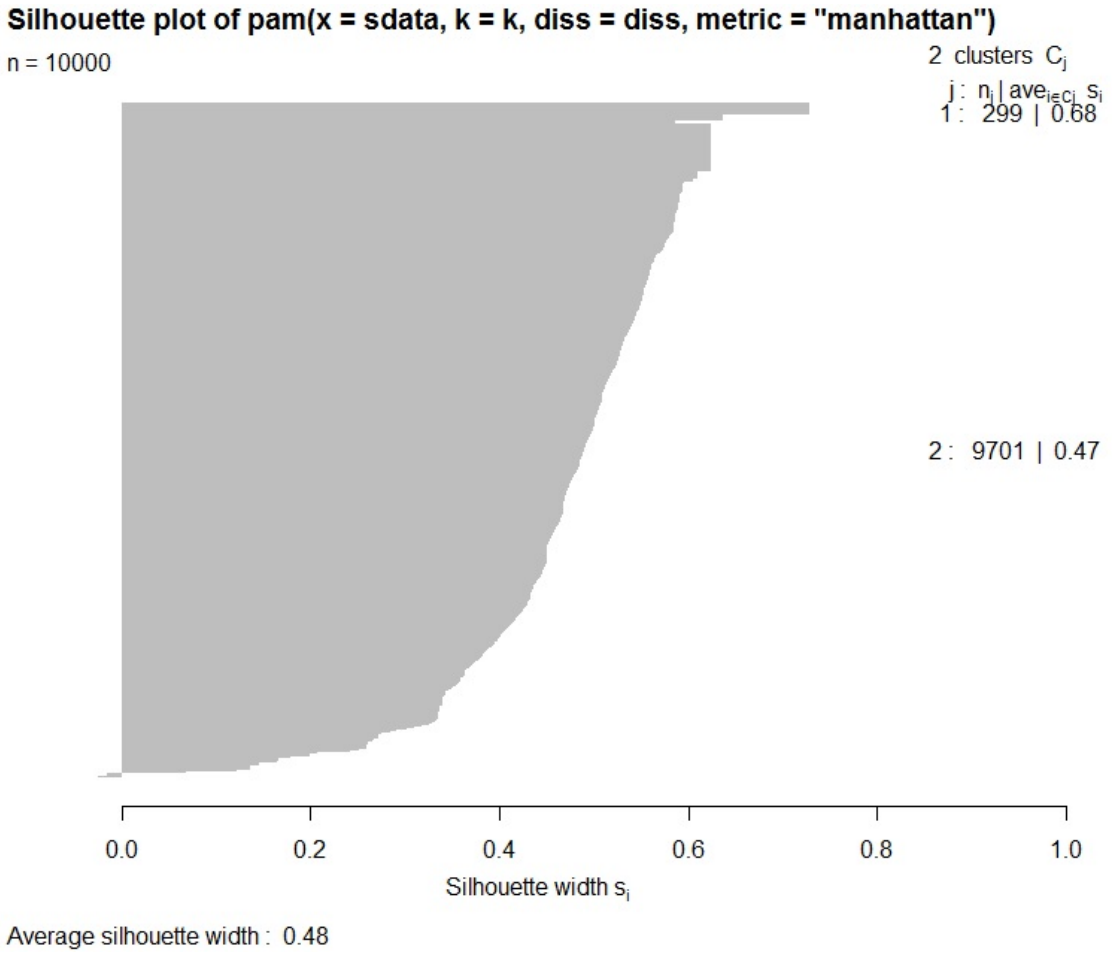

Figure A.8.4: Silhouette plot for PAM. Sparsity = 0.96.

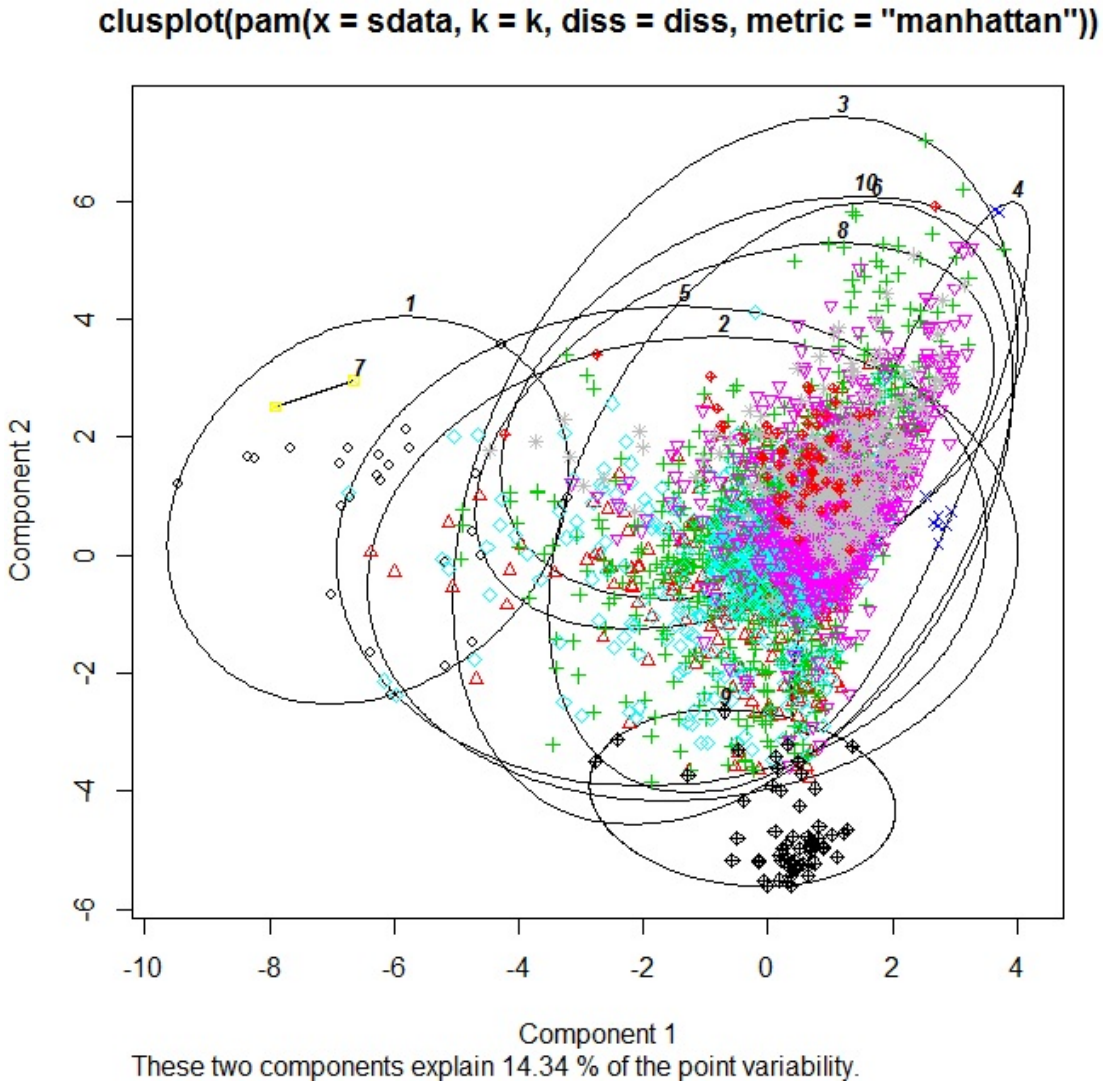

Figure A.8.5: Cluster plot from PAM. Visualised by two leading principal components. Number of clusters k = 10. Sparsity = 0.95. Contains 49 terms.

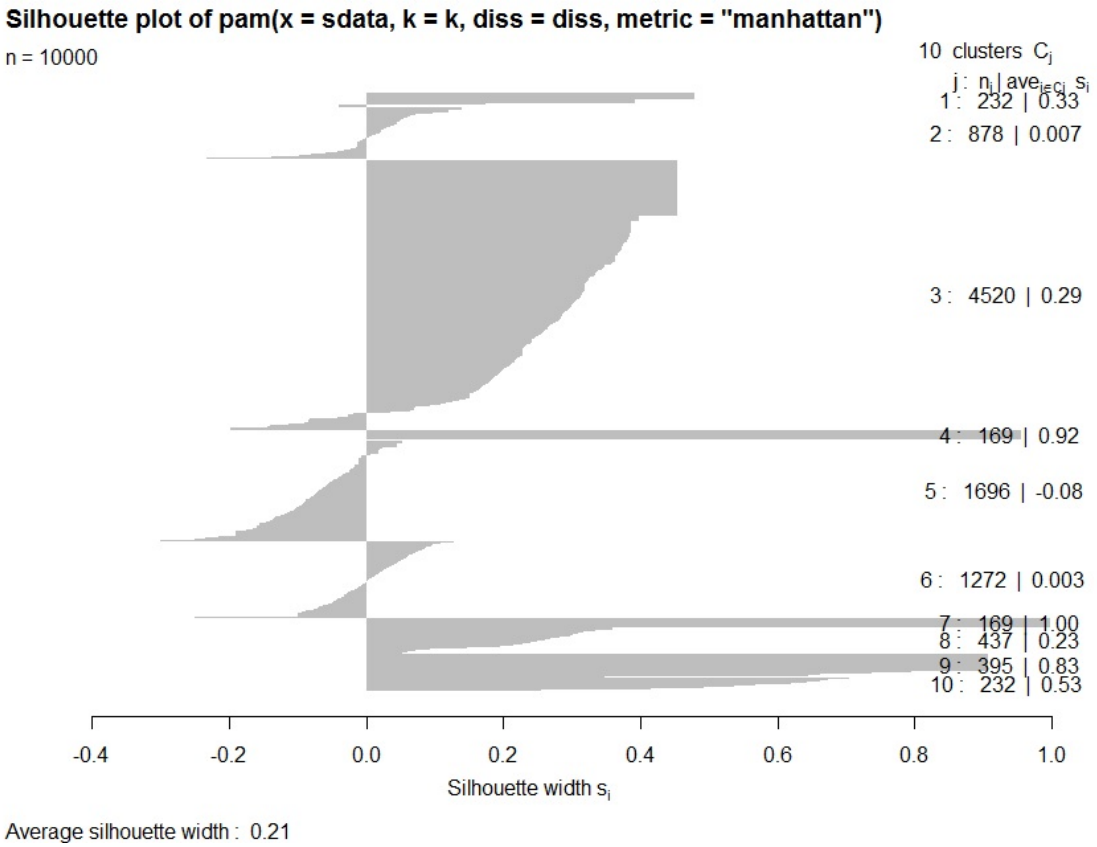

Figure A.8.6: Silhouette plot for PAM. Sparsity = 0.95.

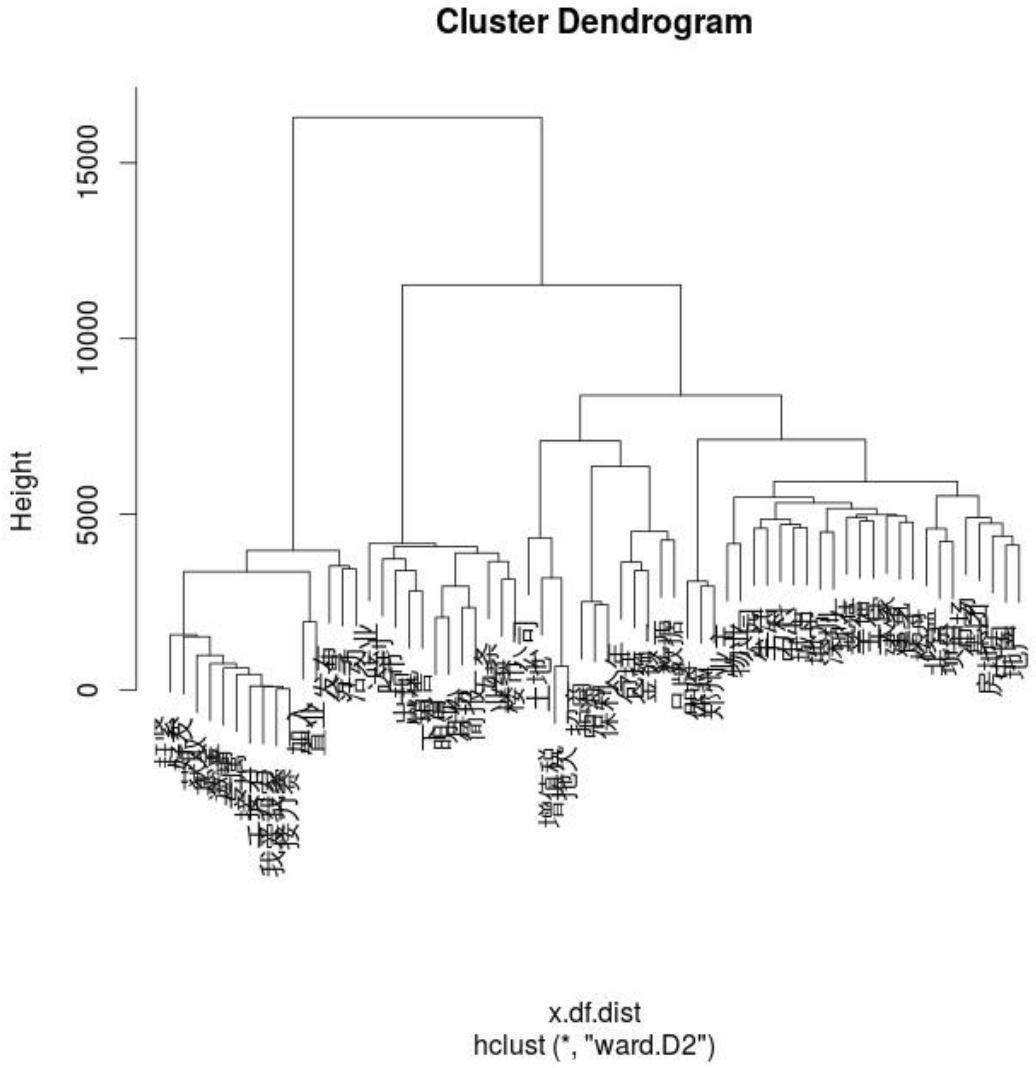

Figure A.8.7: Agglomerative hierarchical clustering. Sparsity = 0.97.

Appendix B

Appendix for Chapter 3

B.1 Additional Figures for Positive/Negative Polarity

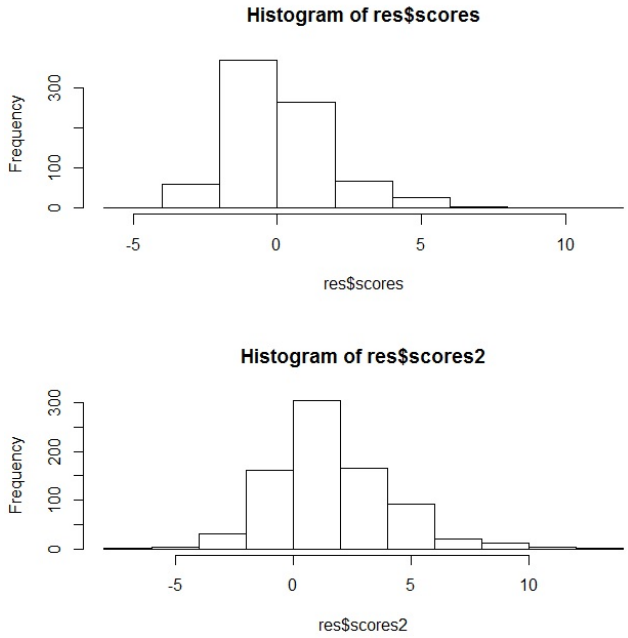

Figure B.1.1: Histograms of polarities (comparison between Chinese Emotional Words Ontology (top) and Hownet Chinese Message Structure Base (bottom)).

```
> skewness(res$scores)
[1] 0.983169
> agostino.test(res$scores, alternative = c("two.sided"))

        D'Agostino skewness test

data:  res$scores
skew = 0.9832, z = 35.7704, p-value < 2.2e-16
alternative hypothesis: data have a skewness
```

Figure B.1.2: Skewness test of positive and negative sentiments.

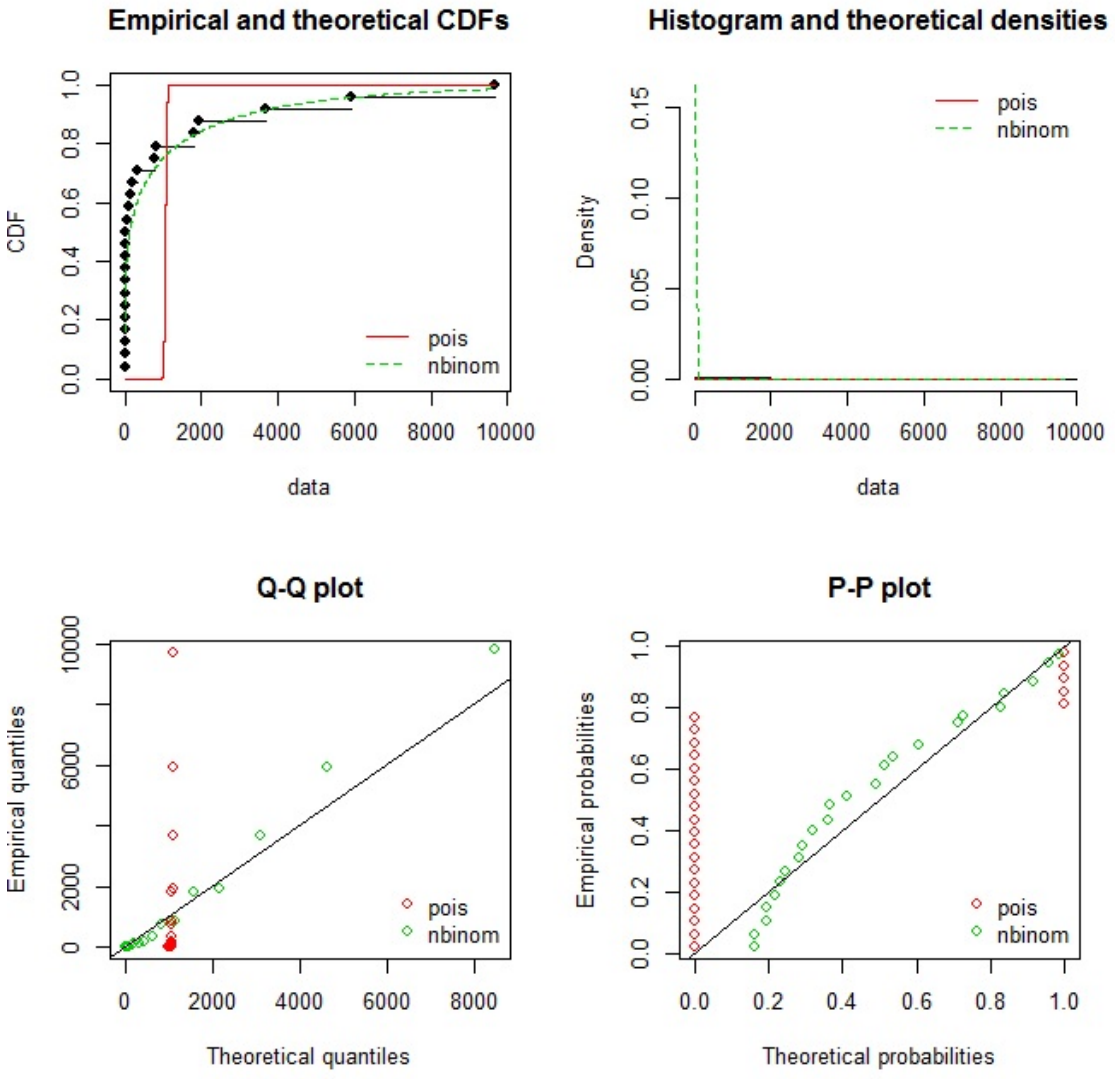

Figure B.1.3: Poisson and Negative Binomial distribution fitting.

```
Chi-squared statistic:  Inf 5.297544
Degree of freedom of the Chi-squared distribution:  4 3
Chi-squared p-value:  0 0.1512618
   the p-value may be wrong with some theoretical counts < 5
Chi-squared table:
        obscounts     theo pois theo nbinom
<= 1            4  0.000000e+00    4.735946
<= 8            4  0.000000e+00    2.016561
<= 29           4  0.000000e+00    2.028308
<= 181          4 5.041599e-245    4.078941
<= 1803         4  2.400000e+01    7.038344
> 1803          4  0.000000e+00    4.101901

Goodness-of-fit criteria
                                       pois   nbinom
Aikake's Information Criterion 71875.90 317.4523
Bayesian Information Criterion 71877.08 319.8084
```

Figure B.1.4: Information of Poisson and Negative Binomial distribution fitting.

```
> significance

        Pearson's product-moment correlation

data:  res$posn and res$negn
t = 10.081, df = 25533, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.05073748 0.07517117
sample estimates:
       cor
0.06296376

> significancem

        Pearson's product-moment correlation

data:  posnm and negnm
t = 9.3649, df = 2794, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.1382789 0.2101640
sample estimates:
      cor
0.1744539
```

Figure B.1.5: Correlation test of positive and negative sentiments.

temp2								
精彩	犀利	展现	正面	配合	奔腾	顶礼膜拜	确实	忠诚
133	133	133	133	134	135	135	136	137
领先	必须	全面	装饰	惊喜	重要	黄金	先进	值得
138	139	143	146	154	154	159	161	163
广阔	高端	相信	含蓄	青睐	升级	焕然一新	突破	获得
175	177	181	190	199	206	209	239	244
节能	增长	感谢	真正	适时	哈哈	专家	美女	大道
254	257	259	260	262	301	304	344	357
成功	最高	喜欢	创新	支持	吉利	大礼	最新	安全
364	374	381	392	464	472	512	542	654
不错	丰富	时尚	全新	自主				
663	671	724	1265	2006				

Figure B.1.6: Top 50 positive sentiment words.

敲诈	迟到	骄傲	亏待	窃取	波折	查抄	失去	犯罪
27	28	28	28	28	29	29	29	30
困境	手段	危机	野蛮	超载	随便	无知	拖累	鄙视
30	30	30	30	31	31	31	32	33
麻烦	破产	抢劫	入侵	逃逸	不够	抄袭	报废	假冒
34	34	34	34	35	36	36	38	40
恶心	忽悠	无耻	不可	侵入	不满	抛弃	殴打	涉嫌
41	41	48	52	55	59	60	69	71
狂人	坐井观天	不行	严重	不好	刺激	胡扯	疯狂	遭遇
75	79	81	81	89	91	91	99	99
惩治	污染	事故	犹豫	不足				
109	129	212	289	319				

Figure B.1.7: Top 50 negative sentiment words.

B.2 Additional Figures for Seven Dimensions Sentiments

```
Goodscore
   1    2    3    4    5    6    7    8    9   10   11   12   13
6392 4646 2166 1275  718  421  140   84   53   10    5    2    2
> table(Happinessscore)
Happinessscore
   1    2    3    4    5    6    7    8
4097 1055  363  113   26   35    2    1
> table(Surprisescore)
Surprisescore
  1   2   3
363  11   2
> table(Angerscore)
Angerscore
  1   2   3
506  63   2
> table(Fearscore)
Fearscore
  1   2   3   4   5
961  91   7   2   1
> table(Sadnessscore)
Sadnessscore
   1    2    3    4    5    6
1901  154   52   11    3    1
> table(Disgustscore)
Disgustscore
   1    2    3    4    5    6    7    8   10
4406 1033  472  143   84   31    6    5    1
```

Figure B.2.1: Scores of seven dimension sentiments.

	Good	Happiness	Surprise	Anger	Fear	Sadness	Disgust
Good	NA	0.161	0.027	-0.033	0.018	0.068	0.110
P-value1	NA	0.000	0.000	0.000	0.004	0.000	0.000
Happiness	0.161	NA	0.012	-0.035	0.028	-0.003	0.132
P-value2	0.000	NA	0.056	0.000	0.000	0.687	0.000
Surprise	0.027	0.012	NA	-0.005	0.017	0.004	0.033
P-value3	0.000	0.056	NA	0.433	0.007	0.498	0.000
Anger	-0.033	-0.035	-0.005	NA	0.014	-0.008	0.117
P-value4	0.000	0.000	0.433	NA	0.023	0.181	0.000
Fear	0.018	0.028	0.017	0.014	NA	0.012	0.072
P-value5	0.004	0.000	0.007	0.023	NA	0.062	0.000
Sadness	0.068	-0.003	0.004	-0.008	0.012	NA	0.100
P-value6	0.000	0.687	0.498	0.181	0.062	NA	0.000
Disgust	0.110	0.132	0.033	0.117	0.072	0.100	NA
P-value7	0.000	0.000	0.000	0.000	0.000	0.000	NA

Figure B.2.2: Correlation and p-values between different sentiments.

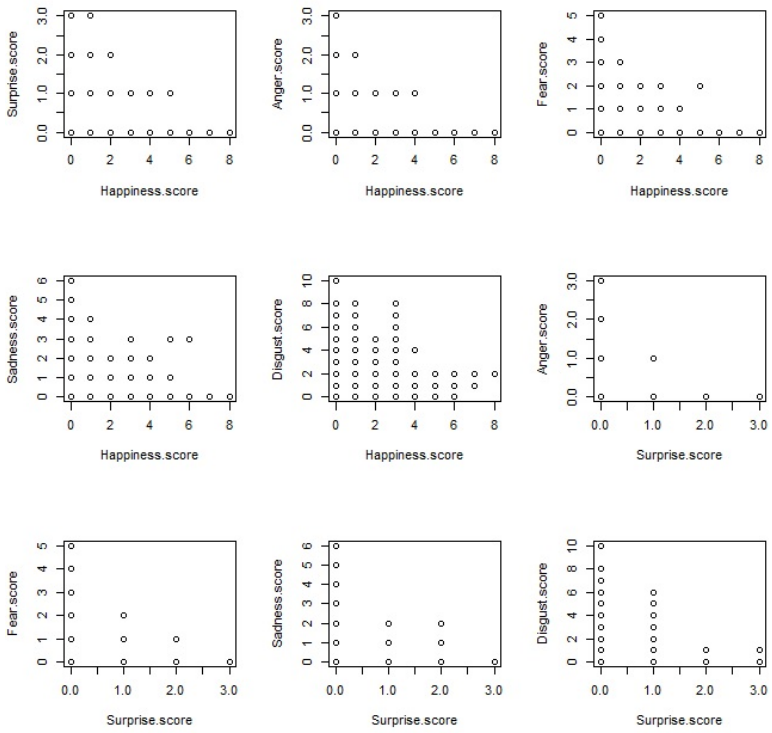

Figure B.2.3: Plots of Happiness, Surprise score vs other remaining scores.

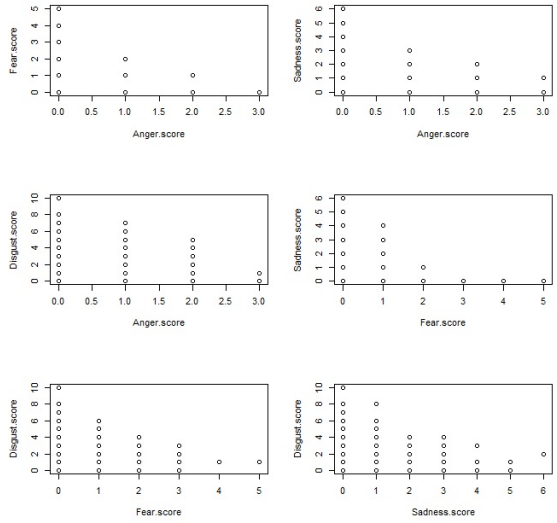

Figure B.2.4: Plots of Anger, Fear, Sadness score vs other remaining scores.

B.3 Details for Fitting Proportional Negative Sentiment Time Series

B.3.1 R Package rugarch for Fitting Proportional Negative Sentiment

R package *rugarch* offers a comprehensive set of methods for modelling the univariate GARCH process, including fitting, filtering, simulation, and diagnostic tools for plots and various tests.

The Model fitting procedure can be divided into two functions: $ugarchspec$ for specifying a model which we intended to fit, and $ugarchfit$ for conducting the fitting and optimisation. The first function includes three parts of specifications: mean model, variance model and distribution model. In this way, a model can be illustrated by the dynamics of the conditional mean, conditional variance, and the distribution to which the residuals belong, which determines any additional parameters.

In the conditional mean model specification, it is possible to add ARCH-in-mean effects (Engle et al., 1987) and m external regressors into the general ARFIMAX model. It can be formally defined as:

$$\Phi(L)(1-L)^d(y_t-\mu_t)=\Theta(L)\epsilon_t \tag{B.1}$$

with the left hand side AR specification and the right hand side MA specification. (L) is the lag operator, $(1-L)^d$ is the long memory fractional process with $0 <$

$d < 1$, and μ_t can be further defined as

$$\mu_t = \mu + \sum_{i=1}^{m-n} \delta_i x_{i,t} + \sum_{i=m-n+1}^{m} \delta_i x_{i,t} \sigma_t + \xi \sigma_t^k \tag{B.2}$$

where $x_{i,t}$ are the possible external regressors, with amount m and the last n of m can be multiplied by the conditional standard deviation σ_t if required. The last term is the ARCH-in-mean choice: k can be 1 for conditional standard deviation and 2 for conditional variance.

For the conditional variance, various extensions of the GARCH model can be chosen: fGARCH, sGARCH, eGARCH, gjrGARCH, apARCH, iGARCH and csGARCH (Ghalanos, 2012). The fGARCH model (Hentschel, 1995) subsumes plenty of submodels as a complement of the ordinary GARCH settings. In our model fitting, a model with Asymmetric Power ARCH conditional variance (apARCH) stands out and the detailed model is introduced in Formula (3.11). A range of distributions including Normal, Student-t, Generalized Error, and their skew variants can be chosen for the residuals in the *ugarchspec* function.

After model specifications, the model fitting is executed by the function $ugarchfit()$. In this procedure, different solvers can be selected.

B.3.2 Additional Figures for Proportional Negative Sentiment Time Series

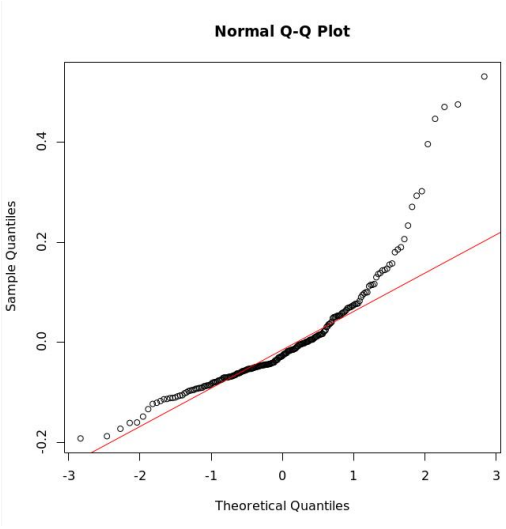

Figure B.3.1: QQ plot against normal ARMA for proportional negative sentiment (adjusting the huge spike between Day 206 and Day 208 by threshold clearance).

Appendix C

Appendix for Chapter 4

C.1 Additional Topic Words and Original Posts

Example Two:
Topic Words:
"世界" "需要" "成功" "王石" "精神" "总经理" "别人" "高峰" "辞去" "动摇" "行囊" "踏遍"
("world" "need" "success" "Shi WANG" "spirit" "general manager" "others" "peak" "resign" "change" "adventure bag" "climb")

Related Posts:
Eg1. @傅老幺: 厉害!!//王石说：我的成功是别人不再需要我: 1999 年，王石辞去万科总经理一职，背上行囊，踏遍世界各大高峰，但是，王石对万科的精神引领，从未动摇过。
(@Fulaoyao: That's amazing!!//Wang said, My success is, in the end, that others do not need me any more. In 1999, Wang resigned from the position of general manager of Vanke, took his adventure backpack and climbed the world's major mountain peaks. Although he had left, his leadership spirit in Vanke never changed)

Eg2.@蜜斯朵拉:【王石说——我的成功是别人不再需要我】1999 年，王石辞去万科总经理一职，背上行囊，踏遍世界各大高峰，但是，王石对万科的精神引领，从未动摇过。本书源于时代纪录对王石三年的贴身跟踪拍摄，从大量的跟拍记录中摘取最能体现王石对于万科、特别是对万科的未来思考的言论。http://t.cn/zj1GLiP
(@Misiduola: 【Wang said, My success is, in the end, that others do not need me any more】In 1999, Wang resigned from the position of general manager of Vanke, took his adventure backpack and climbed the world's major mountain peaks. Although he had left, his leadership spirit in Vanke never changed. Wang's life was filmed and extracts from this used to make this book, which reflected his opinions about Vanke's future. http://t.cn/zj1GLiP)

Eg3.@世界之大为何会相遇: 王石说，我的成功是别人不再需要我。1999 年，王石辞去万科总经理一职，背上行囊，踏遍世界各大高峰， 我在:|周屋路
(@Shijiezhidaweihehuixiangyu: Wang said, My success is, in the end, that others do not need me any more. In 1999, Wang resigned from the position of general manager of Vanke, took his adventure backpack and climbed the world's major mountain peaks,..I'm at Zhouwu Road)

Concluded topic from the posts:
The story of Shi WANG from his autobiography: Wang said, My success is, in the end, that others do not need me any more. In 1999, Wang resigned from the position of general manager of Vanke, took his adventure backpack and climbed the world's major mountain peaks. Although he had left, his leadership spirit in Vanke never changed.
[popular post]

Figure C.1.1: Example Two: topic words and original posts.

Example Three:
Topic Words:
"分享" "活动" "发布" "赶紧" "加入" "置业" "好友" "邀请" "基金" "获取" "专属" "我爱我家" "千万" "接力" "接力赛"
("share" "activity" "release" "hurry" "join" "real estate" "friends" "invite" "funds" "receive" "exclusive" "I love my family" "millions of" "relay" "relay race")

Related Posts:
Eg1. @chenhao_64411_388: @BraveGoblin #我爱我家接力赛#接力中！只需@好友发布微博获取专属链接，邀请好友点击专属链接加入活动，分享约千万万科置业基金！专属链接：http://t.cn/zTRu3ZV 赶紧来点击吧！
(@chenhao_64411_388: @BraveGoblin #I love my family relay race# RELAYING! Just '@' your friends and repost the weibo to receive an exclusive link, then invite you friends to click on the exclusive link to join the activity: sharing millions of Vanke real estate funds! Exclusive link: http://t.cn/zTRu3ZV. Hurry to click!)

Eg2. @日月不暗: #我爱我家接力赛#接力中！只需@好友发布微博获取专属链接，邀请好友点击专属链接加入活动，分享约千万万科置业基金！ @中国股市短线 @谈股论经 28 @阳光黑股推票 @短线狙击营
(@Riyuebuan: #I love my family relay race# RELAYING! Just '@' your friends and repost the weibo to receive an exclusive link, then invite you friends to click on the exclusive link to join the activity: sharing millions of Vanke real estate funds! @Zhongguoguoshiduanxian @Tangulunjin28 @Yangguangheigutuipiao @Duanxianjujiying)

Eg3. @aero069: 响应沈公子号召，绝非植入广告，特此声明。#我爱我家接力赛#接力中！邀请好友点击专属链接加入活动，分享约千万万科置业基金！专属链接：http://t.cn/zHMvcC4
(@aero069: I declare this post is in response to the call of Mr. Sheng, not an embedded advertisement. #I love my family relay race# RELAYING! Just '@' your friends and repost the weibo to receive an exclusive link, then invite you friends to click on the exclusive link to join the activity: sharing millions of Vanke real estate funds! Exclusive link: http://t.cn/zHMvcC4)

Concluded topic from the posts:
#I love my family relay race# RELAYING! Just '@' your friends and repost the weibo to receive an exclusive link, then invite you friends to click on the exclusive link to join the activity: sharing millions of Vanke real estate funds! Exclusive link: http://t.cn/zTRu3ZV. Hurry to click!
[a promotion held by Vanke official account: if the follower repost this promotion, there is a chance to win money]

Figure C.1.2: Example Three: topic words and original posts.

C.2 Non-persistent Topics in Monthly Topic Evolutions

Non-persistent topics in monthly topic evolutions:

B2: News: Wang said in Shanghai, China's real estate market bubble does exist, and the bubble is not insignificant; however, it is unlikely that the bubble would burst in the short term.

B4: News: a 'real estate King' has emerged in Shanghai as well, Vanke had a successful bid for an area in Shanghai Zhangjiang Hi-Tech Park in Pudong District, costing 4.87 billion Yuan. After seven days, Vanke regained the titile "Shanghai real estate King" based on the value of whole year's bids.

B6: Market news: The real estate financing is in full swing, the approval cycle lasts about 5 months (600048 Poly real estate, 000002 Vanke A).

D2: Promotion: Vanke organises Summer Camps to give your child a exciting summer vacation. Remember to forward, comment, and be interactive. If the forwarding amount of your post reaches the top 30, you can get family fun vacation package valued 9000 Yuan.

D3: Promotion: Follow, Repost and Vote for favourite restaurant in Vanke Life Plaza, Win mini iPad.

E1: Popular post: Downloading and recommendation of Real Estate Advertisement Selection.

E2: Popular post: The story of Shi Wang. From his autobiography: Wang said: My

success is, in the end, others do not need me any more. In 1999, Wang resigned from the post of general manager of Vanke, and travelled to the world's major peaks, but the leadership spirit from Wang for Vanke never wavered.

E3: Popular post: (A popular topic/post from Xi'an (place-name) Vanke) about Happiness of Environmental Performance Standards / IAQ Indoor Air Quality / residential sources of pollutants / product model house testing analysis / detection equipment.

E5: Popular post: President of Vanke Yuliang's speech on the relationship between company size and its regulations Yu Liang, president of Vanke recetly said, "When a company reaches a certain size, institutional problems are more important than talent. If a small company with the opportunity to compete against mid-sized companies relies on talent, so big companies will undoubtedly rely on the systems and processes. Vanke is currently considered to be a medium-sized companie, but it may become a large company with value of 300 billion. We should now start to increase the construction of systems and processes as the foundation of the future.

E8: Popular post: @ Wang: Vanke urban communities are exploring the problem of ageing. In the first phase of exploration, our community received the involvement and guidance from related departments of government// @Vanke Weekly: launched a trial in four cities: Vanke R + F Health Centre for Seniors.

E9: Popular post: The turnover on Single's Day (11th Nov) of Tmall reached 35.019 billion Yuan via Alipay transactions. The total amount is roughly equivalent to annual sales of one middle-ranked top 500 enterprise. For instance, in the real estate industry: Yuanyang real estate sales target this year is 35 billion. Hengda's

full-year sales will be over 100 billion this year. The leading company Vanke, last year sales of had over 140 billion.

E11: Popular post: Vanke chairman Wang's speech: a question was raised: What are the opportunities? Wang replied, opportunities like a thief, come quietly, when you let it go, you lose a lot. The best way to seize the opportunities is that you work every day with passion, regard each of the tasks as opportunities, allow you to be infected by the opportunity which is reluctant to go. Someone asked again:"Is it possible to infect a thief?". "The impossible becomes possible by creating the opportunity," Wang answered.

F1: Discussion between Real Estate CEOs followed a piece of news: LIU's reminder: Entrepreneurs in 2015 should know how to manage their cash flow.

F2: Popular competitor Huayuan Estate CEO's post: REN: a land auction gained net added value over four billion yuan. Vanke, China's largest real estate companies, did almost this amount of profits for half year. Government always gained more.

Bibliography

H. Achrekar, A. Gandhe, R. Lazarus, S. Yu, and B. Liu. Twitter improves seasonal influenza prediction. *Fifth Annual International Conference on Health Informatics*, 2 2012.

N. Aletras and M. Stevenson. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers*, pages 13–22, 2013.

S. Asur and B. A. Huberman. Predicting the future with social media. *arXiv preprint arXiv:1003.5699*, 2010.

F. Black. {Stuedies of stock price volatility changes}. 1976.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

A. Blum, A. Kalai, and J. Langford. Beating the hold-out: Bounds for k-fold and progressive cross-validation. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 203–208. ACM, 1999.

J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327, 1986.

BIBLIOGRAPHY

K. Boudt, B. Peterson, and C. Croux. Estimation and decomposition of downside risk for portfolios with non-normal returns. *DTEW-KBI_0730*, pages 1–30, 2007.

G. E. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 211–252, 1964.

G. E. Box and D. A. Pierce. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American Statistical Association*, 65(332):1509–1526, 1970.

G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

P. J. Brockwell and R. A. Davis. *Introduction to time series and forecasting*. Springer Science & Business Media, 2006.

L. Callaghan. *An evaluation of latent Dirichlet allocation in the context of plant-pollinator networks*. PhD thesis, 2013.

G. M. Caporale, C. Ntantamis, T. Pantelidis, and N. Pittis. The bds test as a test for the adequacy of a garch (1, 1) specification: A monte carlo study. *Journal of Financial Econometrics*, 3(2):282–309, 2005.

K. Chan, B. Ripley, M. K. Chan, and J. Cryer. Package tsa. *Opgehaald van http://www. stat. uiowa. edu/˜ kchan/TSA. htm*, 2012.

J. Chang, S. Gerrish, C. Wang, J. L. Boyd-Graber, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, pages 288–296, 2009.

BIBLIOGRAPHY

C. Chatfield. The analysis of time series, 2004.

P. Chaudhuri and J. Marron. Sizer for exploration of structures in curves. *Journal of the American Statistical Association*, 94(447):807–823, 1999.

J. Chen. *The Construction of Chinese emotional words ontology and its applications [D]*. PhD thesis, Dalian University of Technology, 2009.

China-Resonance. Sina commands 56% of chinas microblog market. *Retrieved September*, 14:2012, 2011.

S. W. Cho, M. Cha, and K. Sohn. Topic category analysis on twitter via cross-media strategy. *Multimedia Tools and Applications*, 75(20):12879–12899, 2016.

M. Chrzanowski and D. Levick. Using twitter to predict voting behavior. 2012.

J. Chung and E. Mustafaraj. Can collective sentiment expressed on twitter predict political elections. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence. San Francisco, CA, USA*, 2011.

K. Church, W. Gale, P. Hanks, and D. Hindle. Parsing, word associations and typical predicate-argument relations. In *Proceedings of the workshop on Speech and Natural Language*, pages 75–81. Association for Computational Linguistics, 1989.

CNKI. China knowledge resource integrated database, 2016. URL `http://www.cnki.net/`.

R. B. D'Agostino. Transformation to normality of the null distribution of g1. *Biometrika*, pages 679–681, 1970.

BIBLIOGRAPHY

B. De Finetti. Theory of probability, volume i, 1990.

D. Defays. An efficient algorithm for a complete link method. *The Computer Journal*, 20(4):364–366, 1977.

P. Diaconis. Recent progress on de finettis notions of exchangeability. *Bayesian statistics*, 3:111–125, 1988.

D. A. Dickey and W. A. Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association*, 74 (366a):427–431, 1979.

C. Ding, T. Li, and W. Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis*, 52(8):3913–3927, 2008.

T. Ding, V. Fang, and D. Zuo. Stock market prediction based on time series data and market sentiment. 2011.

Z. Ding, C. W. Granger, and R. F. Engle. A long memory property of stock market returns and a new model. *Journal of empirical finance*, 1(1):83–106, 1993.

Z. Dong and Q. Dong. *HowNet and the Computation of Meaning*. World Scientific Publishing Co., Inc., 2006.

P. Ekman. Facial expression and emotion. *American Psychologist*, 48:384–384, 1993.

R. F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the Econometric Society*, pages 987–1007, 1982.

BIBLIOGRAPHY

R. F. Engle and K. F. Kroner. Multivariate simultaneous generalized arch. *Econometric theory*, 11(01):122–150, 1995.

R. F. Engle, D. M. Lilien, and R. P. Robins. Estimating time varying risk premia in the term structure: the arch-m model. *Econometrica: Journal of the Econometric Society*, pages 391–407, 1987.

K. Erk. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653, 2012.

J. Fan, N. E. Heckman, and M. P. Wand. Local polynomial kernel regression for generalized linear models and quasi-likelihood functions. *Journal of the American Statistical Association*, 90(429):141–150, 1995.

A. Fang, C. Macdonald, I. Ounis, and P. Habel. Using word embedding to evaluate the coherence of topics from twitter data. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1057–1060. ACM, 2016a.

C. Fang, A.and Macdonald, I. Ounis, and P. Habel. Topics in tweets: A user study of topic coherence metrics for twitter data. In *European Conference on Information Retrieval*, pages 492–504. Springer, 2016b.

I. Feinerer. Introduction to the tm package text mining in r, 2014.

C. Fellbaum. *WordNet*. Wiley Online Library, 1998.

C. Fernández and M. F. Steel. On bayesian modeling of fat tails and skewness. *Journal of the American Statistical Association*, 93(441):359–371, 1998.

J. T. S. Ferreira and M. F. Steel. A constructive representation of univariate skewed distributions. *Journal of the American Statistical Association*, 2012.

P. Fryzlewicz et al. Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6):2243–2281, 2014.

E. Gaussier and C. Goutte. Relation between plsa and nmf and implications. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 601–602. ACM, 2005.

D. Gayo-Avello. I wanted to predict elections with twitter and all i got was this lousy paper a balanced survey on election prediction using twitter data, 2012a.

D. Gayo-Avello. No, you cannot predict elections with twitter. *Internet Computing, IEEE*, 16(6):91–94, 2012b.

A. Ghalanos. rugarch: Univariate garch models. *R package version*, 1, 2012.

A. Ghalanos. Introduction to the rugarch package, 2013.

A. Giachanou and F. Crestani. Tracking sentiment by time series analysis. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1037–1040. ACM, 2016.

W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. Introducing markov chain monte carlo. *Markov chain Monte Carlo in practice*, 1:19, 1996.

M. Girolami and A. Kabán. On an equivalence between plsi and lda. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 433–434. ACM, 2003.

A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi. A survey of statistical network models. *Foundations and Trends® in Machine Learning*, 2 (2):129–233, 2010.

J. C. Gower and G. Ross. Minimum spanning trees and single linkage cluster analysis. *Applied statistics*, pages 54–64, 1969.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 101:1, 2004.

B. Grün, K. Hornik, and M. B. Grün. Package topicmodels. 2015.

Z. S. Harris. Distributional structure. *Word*, 1954.

J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1): 100–108, 1979.

L. Hentschel. All in the family nesting symmetric and asymmetric garch models. *Journal of Financial Economics*, 39(1):71–104, 1995.

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

K. Hornik and B. Grün. topicmodels: An r package for fitting topic models. *Journal of Statistical Software*, 40(13):1–30, 2011.

R. J. Hyndman. Time series cross-validation: an r example. `http://robjhyndman.com/hyndsight/tscvexample/`, 2011. Accessed: 2016-05-24.

BIBLIOGRAPHY

B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Twitter power: Tweets as electronic word of mouth. *Journal of the American society for information science and technology*, 60(11):2169–2188, 2009.

P. S. Kalekar. Time series forecasting using holt-winters exponential smoothing. *Kanwal Rekhi School of Information Technology*, 4329008:1–13, 2004.

L. Kaufman and P. J. Rousseeuw. Finding groups in data. an introduction to cluster analysis. *Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, New York: Wiley, 1990*, 1, 1990.

J. A. Khan, S. Van Aelst, and R. H. Zamar. Robust linear model selection based on least angle regression. *Journal of the American Statistical Association*, 102 (480):1289–1299, 2007.

D. Kwiatkowski, P. Phillips, P. Schmidt, and Y. Shin. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of econometrics*, 54(1-3):159–178, 1992.

T. K. Landauer and S. T. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

T. K. Landauer, P. W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

J. Li. Scm repository, index of /pkg/rwordseg - r-forge, 2012. URL `https://r-forge.r-project.org/scm/viewvc.php/pkg/Rwordseg/?root=rweibo`.

J. Li and Y. Chen. Rweibo: An interface to the weibo open platform. *R package version 0*, pages 2–9, 2012.

J. S. Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.

G. M. Ljung and G. E. Box. On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303, 1978.

A. Logunov. *A Tweet in Time: Can Twitter Sentiment analysis improve economic indicator estimation and predict market returns?* PhD thesis, The University of New South Wales Australia, 2011.

C. Louis and G. Zorlu. Can twitter predict disease outbreaks? *BMJ*, 344:1–3, 5 2012.

A. McCallum, K. Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

D. M. McNair, M. Lorr, L. F. Droppleman, et al. *Profile of mood states*. Educational and Industrial Testing Service San Diego, CA, 1981.

N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on*

Empirical Methods in Natural Language Processing, pages 262–272. Association for Computational Linguistics, 2011.

A. Mittal and A. Goel. Stock prediction using twitter sentiment analysis.

E. A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.

D. Newman, J. H. Lau, K. Grieser, and T. 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. Association for Computational Linguistics, 2010.

J. Ong. Chinas sina weibo grew 73% in 2012, passing 500 million registered accounts. *The Next Web*, 2013.

OpinionFinder. Opinion finder system, 2016. URL `https://code.google.com/archive/p/opinionfinder/`.

T. Oreilly. What is web 2.0, 2005.

T. Pavlidis. *Structural pattern recognition*, volume 2. Springer-verlag New York, 1977.

D. Pelleg, A. W. Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML*, volume 1, 2000.

P. C. Phillips and P. Perron. Testing for a unit root in time series regression. *Biometrika*, 75(2):335–346, 1988.

R. R. Picard and R. D. Cook. Cross-validation of regression models. *Journal of the American Statistical Association*, 79(387):575–583, 1984.

X. Ramon. Using twitter as a source of information for time series prediction. Master's thesis, 2012.

K. Rapoza. Chinas weibos vs uss twitter: And the winner is. *Forbes (May 17, 2011)(retrieved August 4, 2011)*, 2011.

A. P. Reynolds, G. Richards, and V. J. Rayward-Smith. The application of k-medoids and pam to the clustering of rules. In *Intelligent Data Engineering and Automated Learning–IDEAL 2004*, pages 173–178. Springer, 2004.

B. Ripley, B. Venables, D. M. Bates, K. Hornik, A. Gebhardt, D. Firth, and M. B. Ripley. Package mass. *CRAN Repository. http://cran. r-project. org/web/packages/MASS/MASS. pdf*, 2013.

J. Royston. An extension of shapiro and wilk's w test for normality to large samples. *Applied Statistics*, pages 115–124, 1982.

D. B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

D. B. Rubin. Multiple imputations in sample surveys-a phenomenological bayesian approach to nonresponse. In *Proceedings of the survey research methods section of the American Statistical Association*, volume 1, pages 20–34. American Statistical Association, 1978.

D. B. Rubin. An overview of multiple imputation. In *Proceedings of the survey research methods section of the American statistical association*, pages 79–84, 1988.

D. Ruppert, S. J. Sheather, and M. P. Wand. An effective bandwidth selector for local least squares regression. *Journal of the American Statistical Association*, 90(432):1257–1270, 1995.

G. Salton, A. Wong, and C. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

H. Schmid. Probabilistic part-of-speech tagging using decision trees. Citeseer, 1994.

H. Schmidbauer, A. Roesch, and V. S. Tunalioglu. *Package 'mgarchBEKK'*, 2016. R package version 0.0.2.

Y. Sheng. Subject hotspots discovery, tracking and analysis based on microblog. *Library and Information Service*, 56(8):32–37, 4 2012.

SinaCorp. Sina released third-quarter financial statement, 2013. URL `http://tech.sina.com.cn/i/2013-11-13/05378908364.shtml`.

B. Singh, R. Sharon, J. Ben, and K. Xu. Real time prediction of road traffic condition in london via twitter and related sources. 2012.

D. Sonderegger. Sizer: Significant zero crossings. r package version 0.1–4, 2011.

M. Stephens. Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):795–809, 2000.

J. Sun. Ansj chinese word segmentation - github, 2012. URL `https://github.com/ansjsun/ansj_seg`.

S. Taylor. Modeling financial time series, 1986, 1986.

M. Thelwall, K. Buckley, and G. Paltoglou. Sentiment in twitter events. *Journal of the American Society for Information Science and Technology*, 62(2):406–418, 2011.

D. Tian and W. Zheng. Chinese microblogger character analysis using svm. In *Wavelet Active Media Technology and Information Processing (ICWAMTIP), 2012 International Conference on*, pages 385–389. IEEE, 2012.

Y. Tian. *Trends Analysis and Prediction Based on Micro-Blogging Platforms*. PhD thesis, Computer School Wuhan University, Wuhan University Press, 5 2012.

H. Tong. *Non-linear time series: a dynamical system approach*. Oxford University Press, 1990.

C. H. Tsai. Mmseg: A word identification system for mandarin chinese text based on two variants of the maximum matching algorithm, 2000. URL `http://technology.chtsai.org/mmseg/`.

R. S. Tsay. Testing and modeling threshold autoregressive processes. *Journal of the American Statistical Association*, 84(405):231–240, 1989.

R. S. Tsay. *Multivariate time series analysis: with R and financial applications*. John Wiley & Sons, 2013.

R. S. Tsay. *Package 'MTS'*, 2015. R package version 0.33.

P. D. Turney, P. Pantel, et al. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, 2010.

TwitterInc. Twitter,inc. - securities registration statement, 2013. URL `http://twitter.q4cdn.com/86436e11-a11a-423a-a38b-6f07c7e10a6a.pdf?noexit=true`.

TwitterInc. Twitter data with hashtag apple, 2016. URL `https://data.world/kike/170k-apple-tweets`.

H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1105–1112. ACM, 2009.

M. P. Wand and M. C. Jones. *Kernel smoothing*. Crc Press, 1994.

Y. Wang, E. Agichtein, and M. 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*, pages 123–131. ACM, 2012.

J. H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

G. S. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 359–372, 1964.

M. W. Watson. Vector autoregressions and cointegration. *Handbook of econometrics*, 4:2843–2915, 1994.

X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185. ACM, 2006.

BIBLIOGRAPHY

H. White. Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the Econometric Society*, pages 1–25, 1982.

D. Wurtz, Y. Chalabi, and L. Luksan. Parameter estimation of arma models with garch/aparch errors an r and splus software implementation. *Journal of Statistical Software, forthcoming*, 2006.

L. Xie, M. Zhou, and M. Sun. Chinese microblogging sentiment analysis based on svm. *Journal of Chinese Information Processing*, 26(1):73–83, 2012.

Yahoo!Finance. Vanke (000002.sz) historical prices, 2016. URL `https://uk.finance.yahoo.com/q/hp?s=000002.SZ&b=3&a=04&c=2013&e=9&d=11&f=2013&g=d`.

H. P. Zhang, H. K. Yu, D. Y. Xiong, and Q. Liu. Hhmm-based chinese lexical analyzer ictclas. In *Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17*, pages 184–187. Association for Computational Linguistics, 2003.

F. Zheng, D. Miao, Z. Zhang, and C. Gao. News topic detection approach on chinese microblog. *Computer Science*, 39(1):138–141, 1 2012.

S. Zhou, X. Shi, Y. Sun, W. Qu, and Y. Shi. Stock market prediction using heat of related keywords on micro blog. 2013.

D. Zimbra, M. Ghiassi, and S. Lee. Brand-related twitter sentiment analysis using feature engineering and the dynamic architecture for artificial neural networks. In *System Sciences (HICSS), 2016 49th Hawaii International Conference on*, pages 1930–1938. IEEE, 2016.