DEEP LEARNING-BASED DETECTION OF GOVERNMENT ASTROTURFERS IN CHINESE SOCIAL MEDIA

by

ZHOURUI WEI

(Under the Direction of I. Budak Arpinar)

ABSTRACT

With the explosive development of the Internet, information and its spread across the network have grown exponentially. The rapid growth of information places higher demands on the information-censoring work of governments, even in areas where government censorship is rigorous. Astroturfing in political science is the practice of a government or an organization masking ideas with specific content designed to influence ordinary individuals or manipulate the orientation of public opinion. When we try to analyze social media content for multiple tasks, it is difficult to distinguish this government propaganda content from the views of ordinary citizens. In this thesis, a deep learning-based method proposed for detecting the Fifty Cent Army was trained with a database, named CSMCDB, of 24,788 labeled comments posted on mainstream Chinese social media and news websites. According to multiple test cases, this method allows users to predict Fifty Cent Army comments from different resources with high accuracy.

INDEX WORDS: Astroturfer, NLP, Neural Network, classification, Social Media, CNN, RNN, Fifty Cent Army

DEEP LEARNING-BASED DETECTION OF GOVERNMENT ASTROTURFERS IN CHINESE SOCIAL MEDIA

by

ZHOURUI WEI

B.Eng., Lanzhou University of Technology, China, 2011

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2019

DEEP LEARNING-BASED DETECTION OF GOVERNMENT ASTROTURFERS IN CHINESE SOCIAL MEDIA

by

ZHOURUI WEI

Major Professor:	I. Budak Arpinar
Committee:	Rongbin Han
	Jaewoo Lee

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
May 2019

ACKNOWLEDGEMENTS

I want to show my special thanks and appreciation to Dr. Arpinar, my major Professor. Thank you for your guidance and encouragement in this research. Sincere thanks to Dr. Han from the School of Public & International Affairs. Thank you for the insightful discussions and for supporting me in this research as my committee member. I would express great thanks to my other committee member, Dr. Lee for your counsel and precious time. Special thanks to Yu Tian, Zheliang Liu, Fei Gao, and Sindhuri Chandrupatla for your countless help during my research period. Great thanks to all my friends and family for your love and staunch support.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS iv

LIST OF TABLES vii

LIST OF FIGURES viii

CHAPTER

1 INTRODUCTION 1

2 THE FIFTY CENT ARMY 6

3 DATA COLLECTION AND PREPARATION 9

3.1 CRAWLER 9

3.2 DATA PREPARATION 10

3.3 EVALUATOR CONSISTENCY TEST 13

3.4 DATA FORMAT 16

3.5 VOCABULARY SET 17

4 CNN AND RNN MODELS 19

4.1 EMBEDDINGS 19

4.2 CNN MODEL 19

4.3 RNN MODEL 23

5 EXPERIMENT 28

5.1 TEST DATASETS 28

5.2 ENVIRONMENT REQUIREMENT 29

6 RESULTS AND DISCUSSION 30

7 RELATED WORK 36

8 CONCLUSION AND FUTURE WORK 38

BIBLIOGRAPHY 40

APPENDICES 44

A QUESTIONNAIRE FOR EVALUATOR CONSISTENCY TEST 44

B TRAINING TEST SUMMARY 45

LIST OF TABLES

Page

Table 2.1: Commentator categories 6

Table 3.1: Data frame of the raw data...11

Table 3.2: Labeled data frame 12

Table 3.3: Four different test cases...16

Table 3.4: Data Format of 2 categories with 6500 rows in each category.........................16

Table 3.5: Data Format of 10 categories with 6500 rows in each category.......................16

Table 3.6: Data Format of 2 categories with 12000 rows in each category.......................16

Table 3.7: Data Format of 10 categories with 12000 rows in each category.....................17

Table 6.1: Average F1-Measure of two models...32

Table 6.2: Truth Table 33

Table A.1: Questions and results for evaluator consistency test 44

Table A.2: Training test summary 45

LIST OF FIGURES

Page

Figure 3.1: The architecture of a standard Scrapy 10

Figure 3.2: The results of Kendall's W coefficients..15

Figure 4.1: The architecture of the proposed CNN model...20

Figure 4.2: ReLU activation function 22

Figure 4.3: Example of RNN 24

Figure 4.4: LSTM architecture 26

Figure 4.5: The architecture of the proposed RNN model...27

Figure 6.1: CNN & RNN model result Summary 30

Figure 6.2: CNN & RNN model Test Accuracy Summary 31

Figure 6.3: CNN & RNN model Test loss Summary 31

Figure 6.4: CNN & RNN model Training time cost Summary 32

CHAPTER 1

INTRODUCTION

For governments that rigorously review and censor information, the strategy of merely blocking websites and deleting comments and articles has gradually become more difficult. Over time, a more aggressive approach has been introduced – deliberately posting comments in favor of the government on social media and news websites, to guide public opinion toward the government's side, on the one hand, and on the other hand, to cover up anti-government speech [1][2]. The government astroturfers in China are collectively referred to as the “Fifty Cent Army (五毛党).” The Fifty Cent Army is composed of network commentators (their official title in mainland China) hired by the government. The main purpose of the Fifty Cent Army is to guide opinions with pro-regime comments applied within a unique framework. It is said that the salary of the Fifty Cent Army is 0.50 RMB per comment (0.07 USD), which may be the origin of the group's name. This thesis will focus on a mechanism to identify the Fifty Cent Army's comments on Chinese social media.

Several articles have been written in this area by researchers such as Gary King [3]. King built a model based on the patterns found from the leaked email archive [16] containing 2013 and 2014 emails to and from the account of the Internet Propaganda Office in Zhanggong District, Ganzhou, Jiangxi, China. The main idea of this model is to map patterns like who, what, when, where, and why to identify the Fifty Cent Army workers in texts additional to the leaked emails. King’s work [3] was outstanding and effective at that time. The leaked emails [16], however, are significant yet unstructured messy data created from 2013 to 2014 that is now outdated, and the age of the data has substantially weakened its value. The Internet slang in Chinese social media changes continuously and rapidly [29], which can influence the commenting style of the Fifty Cent Army. To achieve the best results, the Internet Propaganda Office may require the Fifty Cent Army workers to improve their commenting style by using the most popular expressions.

Blake Miller conceived of another method to automatically identify and analyze the Fifty Cent Army by tracking certain specific behaviors among Fifty Cent Army commentators [4]. Miller traces the following identifiers: multiple users posting from the same IP address; a massive volume of posts originating from one IP address; sentence-long posts associated with the IP address; and the IP address having Weibo accounts that follow or have been followed by official government accounts. With that set of classifiers, Miller has achieved accuracy rates as high as 94% in predicting comments from the leaked email archive, which is the same archive used by King [3].

Miller has likely achieved the most effective method [4] to date, but there are a few potential deficiencies in his approach. First, the mechanism Miller proposed requires IP addresses. However, most platforms that support online comments no longer reveal IP addresses. Moreover, individuals in mainland China increasingly prefer to use virtual private networks (VPNs) or other kinds of IP-masking techniques for security or anonymity. We believe that such a trend will also occur among the Fifty Cent Army and further weaken the effectiveness of features that learn from IP addresses. Second, the idea is to identify Fifty Cent Army workers through a current model based on the user account system called Sina Passport created by Sina (which also created the popular social media site Sina Weibo). All official government accounts are based on this user account system. However, Sina Weibo is not the only mainstream social media in mainland China; other websites such as qq.com, 163.com, ifeng.com, bilibili.com, blogs, and local forums employ user account systems other than Sina Passport. Third, the dataset prior researchers have used was not focused on human labeling. As mentioned, the dataset [16] was extracted from a very unstructured source that contains excessive noise. It even includes many Q&As about community life, which are not related to the Fifty Cent Army at all. In our view, both King's work [3] and Miller's work [4] may not have fully considered the complexity of comments within mainland China due to these limitations.

To widen the scope of this study, we decided to build a new database. In this thesis, the difficulty of data collection became a primary obstacle. As shown by the previous studies by King and Miller, Chinese-language–based comments can be difficult to distinguish using only a few patterns, such as the inclusion of certain word combinations or specific keywords, relation to some particular person or event, or inclusion of particular emotions. The Fifty Cent Army certainly does not use the same commenting style as when it was discovered. As times have changed, the commenting style has changed as well. Research requiring the abovementioned conditions is challenged by a perplexing reality: the variability of the Chinese language and the rapid change of Internet slang [29].

To better study the Fifty Cent Army, we need to understand their work. First, they operate on the web, which has a high degree of anonymity and activity. In recent years, the use of VPNs in China has increased. As a result, the propaganda departments in China have also increased their interest in posting through IP-masking methods such as VPNs to create “meticulous external publicity work” [30]. Second, the language of such posts is no longer monotonous. The propagandists have learned to make secondary creations based on an existing corpus. After careful analysis of the Fifty Cent Army’s purposes in a Chinese-language context, we decided to filter Fifty Cent Army comments by news, events, and topics highly related to government decision making, policy announcements, and promotions of government leaders. The Fifty Cent Army workers should strictly follow the trends if they want to manipulate public opinion efficiently.

We used crawlers to collect all related comments, and other metadata, if possible, from sina.com.cn, qq.com, and weibo.com. After collecting all the necessary comments (from both real netizens and the Fifty Cent Army), comments were manually labelled in an extensive process by native language experts. This thesis took advantage of expert-sourcing identification. Due to this difficult manual work, this new database can be considered strongly grounded in reality with highly reliable results, since we also conducted an evaluator consistency test among Chinese-speaking local groups (individuals born in mainland China and who live in mainland China) using Kendall’s coefficient of concordance [8].

Deep learning [17] models have achieved many outstanding results in recent years. Many studies have introduced word embedding [19] and made substantial progress in the field of natural language processing (NLP) [18]. Word embedding is a technical means of mapping words in space. To achieve better results, we also used the word-embedding [19] method in this thesis.

After the labeling work was completed, we split this dataset into three subsets: a training set, a test set, and a validation set. These data were then fed to recurrent neural network (RNN) and convolutional neural network (CNN) models with a word-embedding [19] preprocessing method for training, testing, and validating. The test accuracy is over 95.95%, resulting in reasonable certainty that the neural networks can effectively detect the comments originating from the Fifty Cent Army in future studies.

CHAPTER 2

THE FIFTY CENT ARMY

Since our task is to distinguish Fifty Cent Army comments from those of ordinary netizens, the classification of commentators is crucial. In this thesis, commentators are divided by orientation and by group. This method was inspired by Miller's paper [4].

	Aligned to State Interests	Neural	Opposed to State Interests
Individual	Political supporters	Irrelevant politics	Political critics
Organized	Fifty Cent Army	Paid posters	Foreign hostile forces

Table 2.1: Commentator categories

As shown in Table 2.1, commentators were divided into six categories:

1) *Political supporters* refers to individuals who publish statements in support of the Chinese Communist Party.
2) *Fifty Cent Army* refers to government officials or employees who publish statements in support of the party.
3) *Irrelevant politics* represents individuals who post apolitical statements on various topics.

4) *Paid posters*, also known as the Water Army [32], are netizens hired by an individual or organization to leave fake comments and delete genuine ones for various purposes.
5) *Political critics* refer to individuals who publish statements that criticize the party.
6) *Foreign hostile forces* represent Chinese or foreign individuals or official employees who publish statements against the party.

This thesis focuses primarily on distinguishing Fifty Cent Army comments from all other categories. We know it is complicated to differentiate between the Fifty Cent Army and the Political Supporter comments merely by observing their content. Miller's work suggests setting a threshold of 20 characters to ensure that the comments are written in complete sentences [4]. He claimed that the average post length in the leaked archive was approximately 30 characters, and most were complete sentences. However, we analyzed the leaked posts by eliminating null values, duplicate symbols, and URLs within posts. The average length of the leaked posts in this analysis was 125 characters. There were 5,498,213 valid characters distributed in 43,827 lines. As identified by a manual sampling of the leaked archive posts, many posts with fewer than 20 characters were also posted by the Fifty Cent Army, some of which were tagged with departmental identities within brackets; examples include "英名永在浩气长存！（区行管委）Your reputation and noble spirit live forever! (District Administration Committee)"; "追怀英烈，永垂不朽！（区行馆委雷）Heroes never die! (Lei from District Administration Committee)". Other posts were tagged with a student's name, school name, and grade level: "缅怀先烈,向革命烈士致敬！革命烈士永垂不朽！文清路小学叶坪校区一【4】班吴仁杰 Remember the martyrs and pay tribute to the revolutionary martyrs! Revolutionary martyrs are immortal! Wenqing Road primary school, Yeping campus, grade one, class 4, Wu Renjie"; "水东镇第一保育院赖茜茜 Shuidong Town First Kindergarten, Lai Qianqian". Such comments are more difficult to identify as the Fifty Cent Army's work. Most of the short comments from the Fifty Cent Army have obvious commonalities. A pair or several pairs within groups consisting of three or four Chinese characters form a particular type of Chinese slogan. Many of the comments that were made include typical Chinese slogans for propaganda purposes [31], as illustrated by the leaked archive. The most common Chinese slogan for propaganda purposes was formed by a pair of four Chinese characters with two punctuation marks, which is exactly 10 characters. Therefore, we suggest shortening the length of the threshold to 10 characters. This choice was made because comments like "追怀英烈，永垂不朽！ Heroes never die!" contained only ten characters, were complete sentences, and were common among the work of the Fifty Cent Army.

With a threshold of 10 characters, the CSMCDB set has 12,394 valid comments with the label "Yes". There were 446 comments (3.6%) with fewer than 11 characters. The leaked archive has 43,827 valid comments identified as the Fifty Cent Army's work. There were 4,041 comments (9.2%) with fewer than 11 characters.

We have decided to combine these two types because the instances of Political Supporter individuals (3.6%) were few. Because this thesis examines the Fifty Cent Army comments with the highest percentages, its accuracy can still be trusted.

CHAPTER 3

DATA COLLECTION AND PREPARATION

Data collection is a challenge in this project due to the anti-spider mechanisms on multiple websites. The metadata was not richly constructed. In this thesis, metadata includes usernames (partly), UIDs (partly), comments, and user location (partly). After we decided to build a neural network [20] to detect comments from the Fifty Cent Army, We retrieved all comments as either training, testing, or validating resources.

3.1 CRAWLER

The source websites for this thesis are sina.com.cn, weibo.com, and news.qq.com. We used Scrapy [5] in Python to extract comments from most of the source websites. The architecture of a standard Scrapy is shown in Figure 3.1. The data flow, controlled by the Scrapy engine, is as follows:

The Engine opens a domain, finds the specific Spider handling the website, and requests the Spider for the first URL(s) to crawl. The Engine gets the first URL(s) to crawl from the Spider and then dispatches it in the Scheduler. The Engine requests the Scheduler for the next URL(s) to crawl. The Scheduler returns the next URL(s) to the Engine, which forwards the URL(s) to the Downloader via the Downloader middleware. Once the page has been downloaded, a response will be generated by the Downloader, then sent to the Engine via the Downloader middleware. The Engine receives the response and sends it to the Spider via the Spider middleware. The Spider processes the response and returns items and new requests to the Engine. The Engine hands over the items to the Item Pipeline, then sends the requests to the Scheduler. This process repeats until there are no further requests in the Scheduler.

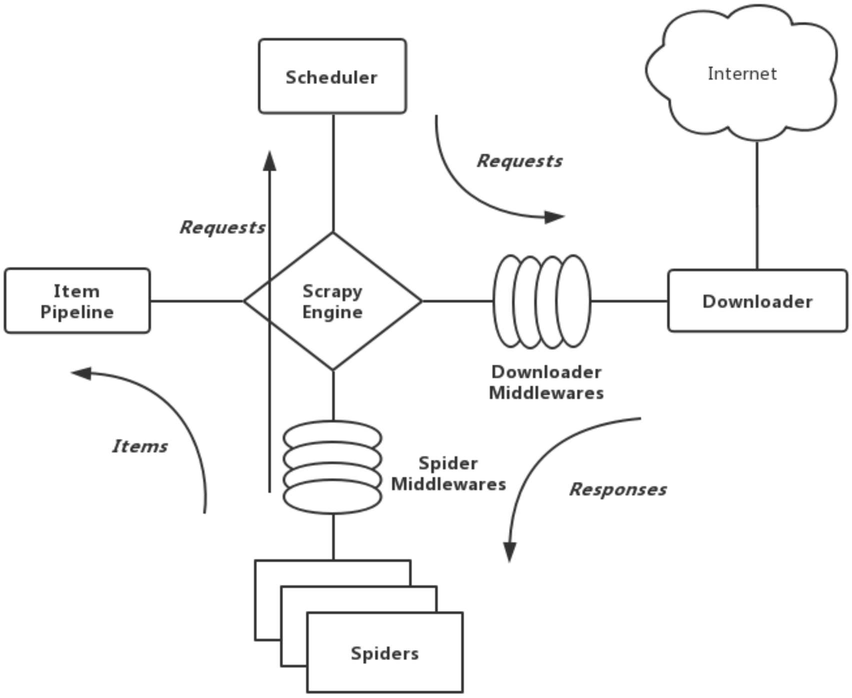


Figure 3.1: The architecture of a standard Scrapy

3.2 DATA PREPARATION

As was predicted, the username, UID, time and location were not fully accessible due to cross-platform and website limitations. The comments, therefore, are the real source for building our database. The comments contain more than just Chinese characters; they may also include blank spaces, emoji, emoticons, pictures, and abuses of punctuation marks. There were a large number of comments with pure emoji, emoticons, pictures, and punctuation marks. The raw data frame is described in Table 3.1.

Username	UID	Comment	Time	Location
The displayed username	The Unique identification ID	Comment text	Post time	User Location

Table 3.1: Data frame of the raw data

To more accurately judge semantics and remove data noise, all comments not containing Chinese characters were rejected. The content of all non-Chinese characters and punctuation, such as emoji and emoticons, were deleted.

Another critical step was to remove duplicate values, called deduplication. The reason to execute deduplication is that each comment within CSMCDB can be viewed as an independent structure with no features related to the frequency of usage. More importantly, for the detection task, every comment has the same possibility to be recognized as originating from the Fifty Cent Army. The existence of duplicates exist could impact the training results, since duplicate comments will be weighted more heavily, which will also offer an inflated sense of model efficacy.

We executed a simple MD5 value comparison and manual verification method for deduplication. This work also can be accomplished using the simhash [6] approach, which was proposed by Moses Charikar's paper entitled "Similarity Estimation Techniques from Rounding Algorithms" [6]. The paper "Detecting Near-Duplicates for Web Crawling: Similarity Research" [7] applied simhash to the deduplication of web pages and also offered a quick way to find neighbors. Unfortunately, due to time limitations, simhash [6] was not applied here to optimize the work of deduplication. After the duplicate content has been removed and the data has been cleaned, the next step is to mark up the data with correct labels. The labeling is completed using either Yes or No indicating if a comment belongs to Fifty Cent Army or not. The labeled data frame is presented in Table 3.2.

Comment	Label
Comment after cleaning	"Yes" or "No"

Table 3.2: Labeled data frame

According to the research by Rongbin Han [9], the ideal comment from the Fifty Cent Army will satisfy at least one of four conditions:

1) The comments should express a positive attitude and should promote the priorities of the party, the leader, and the government;
2) The comments should direct online public opinion with authoritative information on trend incidents in favor of the government and prevent rumors from spreading;
3) The comments may interpret the policies and deployments of the party and direct the emotions of netizens;
4) The comments will uphold the party's direction and champion the decisions and policies made by the party, as well as by the party leaders.

The labeling work strictly followed these four conditions, with qualifying comments marked as "Yes". Here, "qualifiying comments" identify those comments that satisfy one or more conditions. Comments that did not meet any of the four conditions were marked as "No". The labeling work was distributed to 12 Chinese evaluators for manual labeling, and all evaluators used the same evaluation criteria based on the four conditions. Two individuals invited by my previous supervisor in Novartis are coworkers who live in Shanghai, China. Six graduate students, invited by one of my friends, work for the Chinese Academy of Science and live in Lanzhou, Beijing, and Hefei, China. One graduate student who lived in Qingdao, China, was invited by me. Three individual citizens, invited by my family members, live in Chengdu, China. Every evaluator had or was pursuing a bachelor's degree during the evaluation. Nine evaluators did not know each other, and three are friends living in different cities.

The final number of comments after cleaning and labeling was 45,225. Among them, 12,394 comments were labeled as from the Fifty Cent Army, and 32,831 comments were labeled as from the non-Fifty Cent Army. The final database was named Chinese Social Media Comments DB (or CSMCDB in short).

3.3 EVALUATOR CONSISTENCY TEST

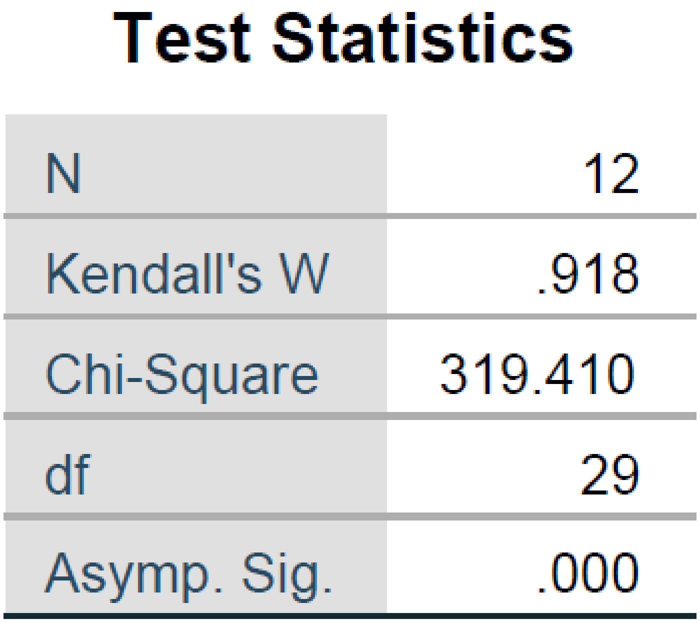
To ensure the reliability of the labeling work, we designed a simple questionnaire based on the Likert scale [25] to assess the reliability of the data labelers (i.e., 12 recruited Chinese individuals). First, we randomly selected 30 comments from the whole dataset without labels. Then, we conducted a questionnaire survey with the selected comments (Q1 - Q30), which were scored from 1 (Absolute non-Fifty Cent Army) to 5 (Absolute Fifty Cent Army) by 12 Chinese evaluators who were born and grew up in mainland China. To ensure the consistency of the evaluators, I was also one of the participants.

For a consistency test with three or more observers and a continuous variable or an ordered categorical variable, Kendall's W test [8] is recommended. In general, research designs using Kendall's W test [8] must meet the following three assumptions:

- Hypothesis 1: The observer group has no fewer than three people, and the result is a continuous variable or an ordered categorical variable. For example, in this study, it is necessary to judge the consistency of the judgment results of 12 evaluators, and the outcome variables are scores from 1 to 5 and are therefore ordered categorical variables.
- Hypothesis 2: The decision result pairing is required; that is, the objects determined by different observers are the same. In this study, 12 evaluators observed the same group of subjects with uniform numbering.
- Hypothesis 3: Observers are independent of one another. This hypothesis requires different observers to complete the outcome judgment independently, without mutual interference.

Based on the existing conditions, we believe that this study meets the three hypotheses of Kendall's W test [8], which can be used for consistency evaluation. The null hypothesis (H0) is that the 12 evaluators do not have consensus, that is, there is no agreement among the 12 evaluators. When the null hypothesis is accepted, W = 0. The results of Kendall's W coefficients test calculated using IBM SPSS [26] are shown in Figure 3.2.

The results in Figure 3.2 suggest that Kendall's W test has asymptotic significance, $P < 0.001$. Thus, the null hypothesis is rejected, which is statistically significant. The results suggest that Kendall's W coefficient [8] for this study was 0.918. In general, Kendall's W coefficients are distributed between 0-1, and the larger the value is, the stronger the consistency. A Kendall's W coefficient [8] higher than 0.9 indicates that the research data has strong consistency, that is, the 12 evaluators in this study show strong consistency in the judgment of whether or not a comment originated with the Fifty Cent Army.



Test Statistics

N	12
Kendall's W	.918
Chi-Square	319.410
df	29
Asymp. Sig.	.000

Figure 3.2: The results of Kendall's W coefficients

The results in Figure 3.2 suggest that Kendall’s W test has asymptotic significance, P < 0.001. Thus, the null hypothesis is rejected, which is statistically significant. The results suggest that Kendall’s W coefficient [8] for this study was 0.918. In general, Kendall’s W coefficients are distributed between 0-1, and the larger the value is, the stronger the consistency. A Kendall’s W coefficient [8] higher than 0.9 indicates that the research data has strong consistency, that is, the 12 evaluators in this study show strong consistency in the judgment of whether or not a comment originated with the Fifty Cent Army.

In conclusion, Kendall’s W coefficient of the 12 evaluators was 0.918, P < 0.001, which indicates that their labeling work can be considered as highly consistent.

3.4 DATA FORMAT

To better test the ability of the model in different types of text classification scenarios, four different test cases were formulated based on different categories and comments in each category for the THUCNews dataset [28]. There are groups of two categories and groups of ten categories, each of which has 6,500 comments or 12,000 comments, respectively. The actual distributions are described in Table 3.3 through 3.7. The detailed categories can be found in section 5.1.

THUCNews	6,500 comments in each category	12,000 comments in each category
2 categories	13,000 Total rows	24,000 Total rows
10 categories	65,000 Total rows	120,000 Total rows

Table 3.3: Four different test cases

Dataset	Shape	Dataset	Shape
x_train	[10,000, 100]	y_train	[10,000, 2]
x_val	[1,000, 100]	y_val	[1,000, 2]
x_test	[2,000, 100]	y_test	[2,000, 2]

Table 3.4: Data Format of 2 categories with 6500 rows in each category

Dataset	Shape	Dataset	Shape
x_train	[50,000, 100]	y_train	[50,000, 10]
x_val	[5,000, 100]	y_val	[5,000, 10]
x_test	[10,000, 100]	y_test	[10,000, 10]

Table 3.5: Data Format of 10 categories with 6500 rows in each category

Dataset	Shape	Dataset	Shape
x_train	[18,000, 100]	y_train	[18,000, 2]
x_val	[4,000, 100]	y_val	[4,000, 2]
x_test	[2,000, 100]	y_test	[2,000, 2]

Table 3.6: Data Format of 2 categories with 12000 rows in each category

Dataset	Shape	Dataset	Shape
x_train	[90,000, 100]	y_train	[90,000, 10]
x_val	[10,000, 100]	y_val	[10,000, 10]
x_test	[20,000, 100]	y_test	[20,000, 10]

Table 3.7: Data Format of 10 categories with 12000 rows in each category

3.5 VOCABULARY SET

The number of Chinese characters is vast, with a total of approximately 90,000. However, the number of commonly used words is approximately 3,000. Even if it is a common word, the difference in frequency of use may be substantial; the national standard GB2312-80 “Basic Collection of Chinese Character Codes for Information Exchange” [10] was published based on this difference. There are two levels of font libraries. The first-level font library contains 3,755 common words. The second-level font library contains 3,008 infrequently used words. The words of the first-level font library have a total frequency of 99.7%. In other words, among every 10,000 Chinese characters in modern Chinese texts, these words will appear more than 9,970 times, and the rest of all Chinese characters will occur fewer than 30 times. For the most commonly used 1,000 Chinese characters, the frequency of use is above 90%.

Character vectors were used in our work instead of word vectors mainly due to the following points. The first is that the word vector requires a fine-tuned word segmentation function. Although there are significant Chinese word segmentation libraries, such as jieba, an excellent word segmentation effect should be built on a large-scale sample, and it is best to include the sentences above and below the target; this effect is especially remarkable in texts with highly coherent style, such as novels. Our current database sample size is small, and the comments are unrelated to the sentences before and after; furthermore, each comment is quite concise. Second, the number of Chinese characters is approximately 90,000, whereas the number of commonly used characters is only approximately 4,000. If we want to cover all possible words, this number will rise to about 400,000. Since this project did not have pre-trained word vectors, it would be difficult to generate such a large vocabulary from our database. Therefore, this thesis set the vocabulary at 5,000 to include as many Chinese characters as possible. This vocabulary set was generated from the top 5,000 most frequent characters from the training set.

CHAPTER 4

CNN AND RNN MODELS

4.1 EMBEDDINGS

An embedding [21] is a mapping from discrete objects to vectors of real numbers. In this case, discrete objects are characters. To meet the requirement of machine learning models, continuous vectors are must, but for NLP [20] tasks, languages do not have a natural vector representation. Hence, a method to transform discrete objects into continuous vectors is necessary.

The introduced embedding layer is different from the famous word2vec [22], but they are doing similar work. This embedding layer can be seen as a fully connected layer.

$$y = Wx + b \tag{1}$$

W here is the vector of words. The initial situation is that W was initialized randomly, and then the word vector matrix will be optimized along with the training process.

The embedding layer in TensorFlow [12] will convert one sentence into a two-dimensional matrix. In this matrix, each column is the character vector for one character, and each row is the sequence of characters of this sentence in order.

4.2 CNN MODEL

The convolutional neural network (CNN) [23] is one class of the deep learning neural networks, inspired by the biological processes of the connectivity patterns in the visual cortex of animals. Each visual cortical neuron responds to limit the range of the visual field, while different cortical neurons can partially overlap and construct vision within the entire visual field [27]. A typical CNN model is formed by one or multiple convolutional layers and a fully connected layer, and it can also include a pooling layer. The architecture of CNN allows it to make use of the two-dimensional structure. Many researchers are also looking to use the CNN model to process Chinese content. The paper entitled "Convolutional Neural Networks for Sentence Classification" from Kim [23] has proven that CNN models are fit for text classification tasks. The model Kim proposed has multiple filter widths, which is why the model can produce various feature maps with different shapes from a sentence matrix. Since short texts can express meaning with limited length, the CNN model is demonstrably capable of impressive performance in short text classification tasks.

We propose a CNN model especially for Chinese text classification that is derived from the work of a researcher named Gaussic [24]. The architecture of this CNN model can be found in Figure 4.1.

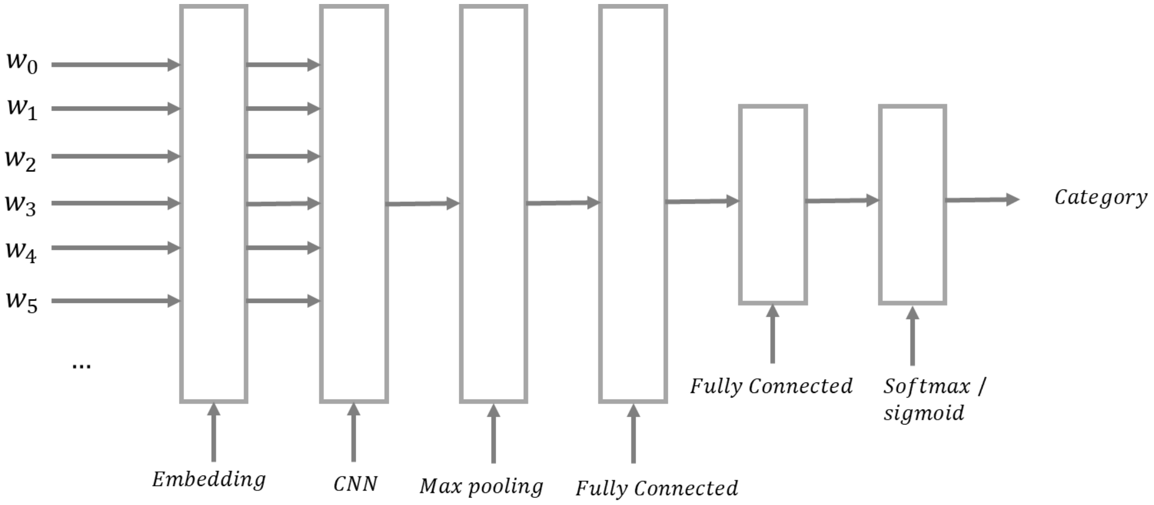


Figure 4.1: The architecture of the proposed CNN model

The embedding layer was discussed before. This layer maps characters in each sentence to a vector matrix, so that the CNN model can process them.

A CNN layer in deep learning models serves to extract different features from the input. In general, the first layer of CNN may only be able to extract low-level features, and additional layers can extract higher-level or more complex features from the output of previous layers. In this thesis, the model is a simplified version that only has one fixed width = 5, which means the kernel of the convolutional layer is a 5 x 64 matrix. Setting the number of filters to 128 provided 128 5 x 64 filter matrices to extract features from the sentence matrix. After convolution operations, there will be 128 different feature maps formed by 96 x 1 (96 = length of each comment – kernel width + 1 = 100 – 5 +1). The output of the CNN layer will be 128 * 96 x 1 feature maps.

A max pooling layer will divide the input matrix into several rectangular areas and will output the maximum value of each area. Pooling layers will reduce the size of the data space, which will reduce the numbers of parameters, and the calculation amount will be reduced as well. Based on the settings, the output of this layer is a 128 x 1 matrix.

The fully connected layer acts as a classifier. While the operation of a convolutional layer, pooling layer, and activation function layer is to map the original data to the feature space of the hidden layer, the fully connected layer plays the role of mapping the learned “distributed feature representation” to the sample mark space. The first fully connected layer continued with dropout [13] and the ReLU [15] activation function.

Dropout means randomly allowing the weights of some hidden layer nodes of the network to not work during the training of the model. Those nodes that do not work can be temporarily considered not part of the network structure, but their weights are retained (merely temporarily, not updated) because next time those nodes may work again. According to Hinton's paper [14], the dropout method can prevent the overfitting problem.

The activation function is ReLU [15], which has several advantages compared to other activation functions. Figure 4.2 shows the ReLU activation function. The ReLU function turns all negative values into 0, whereas the positive value, which is a piecewise linear function, is unchanged. This operation is called unilateral suppression. If there is no activation mechanism, signal delivery and network training will be costly. The ReLU activation function causes the neuron to be silent when it is below the threshold. When the input signal is strong, the difference between the signals can still be retained.

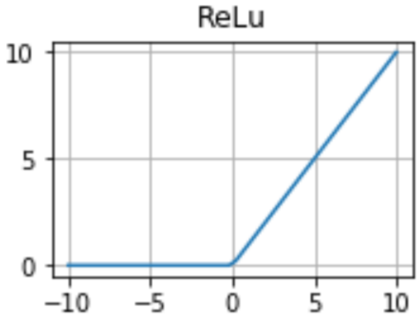


Figure 4.2: ReLU activation function

The use of a softmax layer normalizes the vector to highlight the most significant value and suppress other components far below the maximum. It maps the output of multiple neurons to the (0,1) interval, which can be understood as a probability. For instance, with an array, V, V_i represents the ith element in V, and thus the softmax value of this element is:

$$S_i = \frac{e^i}{\sum_j e^j} \tag{2}$$

The softmax layer is generally used for multiple classification tasks other than binary classification tasks. For binary classification tasks, it is recommended to use a sigmoid layer because the softmax layer will assign probabilities for every class, and the total sum of all probabilities equals 1. If any input has only one neuron, the softmax layer will cause a large loss that is hard to converge. Therefore, for binary classification, a sigmoid layer should be used instead of the softmax layer to ensure that the loss will converge. When there are multiple categories, the softmax layer should be used for the best results.

4.3 RNN MODEL

The recurrent neural network (RNN) can be seen as a class of neural networks. Its primary goal is to process sequence data. In the traditional neural network model, from the input layer to the hidden layer to the output layer, the layers are fully connected, and the nodes between each layer are disconnected. However, this standard neural network is powerless to address many problems. For example, if we want to predict what the next word of a sentence is, we usually need to use the previous word, because the words in a sentence are not independent. RNNs are called cyclic neural networks, where the current output of a sequence is also related to the previous output. The specific form of expression is that the network memorizes the previous information and applies it to the calculation of the current output, that is, the nodes between the hidden layers are no longer disconnected but connected, and the input of the hidden layer includes not only the output of the input layer but also the output of the hidden layer at the previous moment (one RNN example is seen in Figure 4.3). In theory, RNNs can process sequence data of any length. However, in practice, to reduce complexity, it is often assumed that the current state is only related to the previous states.

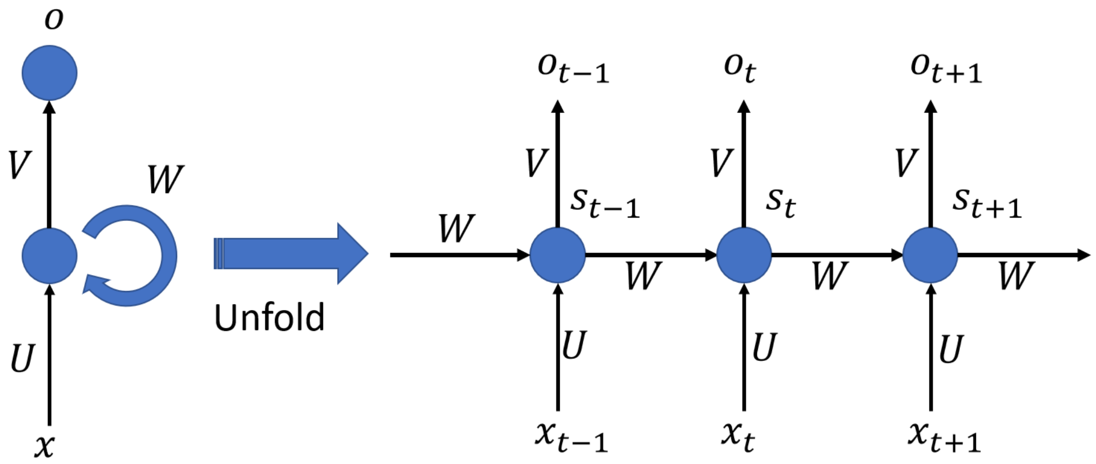


Figure 4.3: Example of RNN

RNNs contain input units, marked as $\{x_0, x_1, \ldots, x_t, x_{t+1}, \ldots\}$, and output units, marked as $\{o_0, o_1, \ldots, o_t, o_{t+1}, \ldots\}$. In addition to these, an RNN also has hidden units, which are marked as $\{s_0, s_1, \ldots, s_t, s_{t+1}, \ldots\}$. Those hidden units complete the most important work. As Figure 4.3 shows, there is a one-way information flow from the input unit (x) to the hidden unit (s) and another one-way information flow from the hidden unit (s) to the output unit (o). The RNN will break the limitation, directing the information back to the hidden unit (s) from the output unit (o). This process is called back projection. The inputs of the hidden unit also contain the states of some previously hidden units.

Figure 4.3 unfolds the RNN into a full neural network. For instance, for a sentence with three words, the unfolded network would be three layers. Each layer represents one word in that sentence. The general calculation process is as follows: x_t represents the tth ($t = 1, 2, \ldots$) input. That is to say, x_0 is the vector of the first word in a sentence. s_t represents the state of the tth step on the hidden layer. It is the memory unit of the network. S_t calculates based on the output of the current input layer and the state of the hidden layer in the previous step:

$$s_t = f(Ux_t + Ws_{t-1}) \qquad (3)$$

The f is generally a nonlinear activation function such as ReLU. When calculating s_0, that is, the hidden layer state of the first word, s_{-1} is needed, but it does not exist, and is generally set to 0 vectors in the implementation. o_t is the output of the tth step.

$$o_t = softmax(Vs_t) \qquad (4)$$

In conclusion, s_t is the memory unit of the network, and s_t contains the hidden layer state of all previous steps. However, o_t only relates to s_t of the current step. To reduce the computing range, in practice, the s_t contains only a few previous steps.

RNNs have proven to be very handy for NLP [20] in practice. The most widely used and most successful model is LSTM (long short-term memory) [11]. It is essentially the same as the general RNN structure, except that different functions are used to calculate the state of the hidden layer. In LSTMs, the i structure is called cells. We can think of cells as black boxes to save the saved state h_{t-1} before the current input x_t. These cells have additional certain conditions to determine which cells are suppressed and which cells are excited. They combine the previous state, current memory, and current input. Figure 4.4 displays an example of LSTM architecture [11].

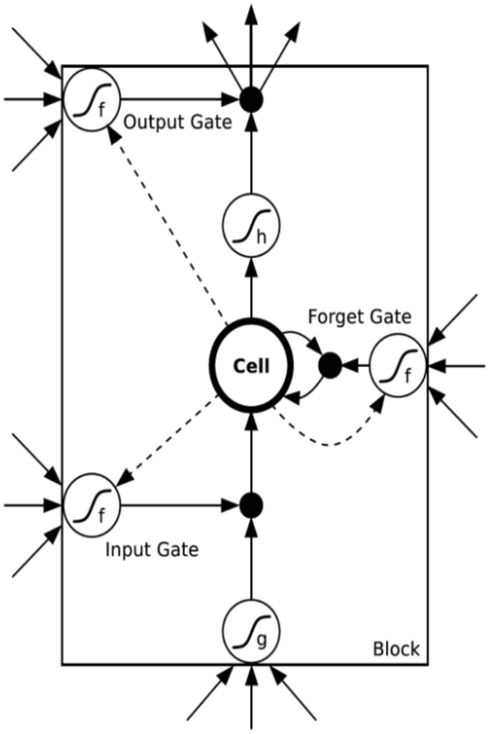


Figure 4.4: LSTM architecture

Here, we propose an RNN model, especially for Chinese text classification, that is derived from the work of a researcher named Gaussic [24]. The architecture of this RNN model can be found in Figure 4.5.

The embedding layer is the same as that in the CNN model. The two connected RNN layers are constructed by LSTM kernels, and each of them has a dropout function to prevent overfitting problems.

The subsequent fully connected layers have dropout and ReLU. Here, a neuron in a layer of the fully connected layer can be seen as a polynomial. Many neurons are used to fit the data distribution; however, using only one layer of the fully connected layer sometimes cannot solve nonlinear problems, which is why sometimes at least two fully connected layers are needed to solve nonlinear problems.

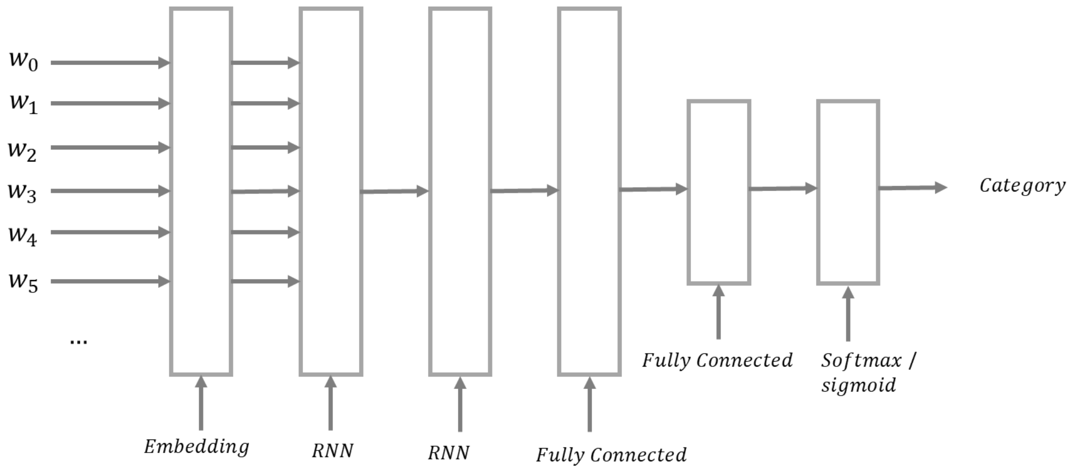


Figure 4.5: The architecture of the proposed RNN model

CHAPTER 5

EXPERIMENT

5.1 TEST DATASETS

We tested two proposed models on different datasets. The data format can be found in Chapter 3.

1. THUCNews: THUCNews was generated with the data from the Sina News RSS subscription channel from 2005 to 2011. It contains 740,000 news documents (2.19 GB), all in UTF-8 plain text format. It also comprises 14 candidate categories: finance, lottery, real estate, stocks, home, education, technology, society, fashion, politics, sports, constellations, games, entertainment.
    a. Group of two categories: education, games
    b. Group of ten categories: sports, finance, real estate, home, education, technology, fashion, politics, games, entertainment
2. CSMCDB: The CSMCDB database was generated from the mainstream news websites (sina.com.cn, weibo.com, news.qq.com) from 2018 to 2019. It contains 45,225 comments labeled Yes or No, indicating whether or not the comment originated from the Fifty Cent Army. CSMCDB has only two categories: Fifty Cent Army, non-Fifty Cent Army.

3. Leaked archive: The leaked archive was retrieved from King’s work [3]. It contains 43,827 comments identified as the work of the Fifty Cent Army after eliminating null values, duplicate symbols, and URLs.

For all the datasets we use in this project: charset was converted to UTF-8, using a dropout rate of 0.5, embedding dimension of 64, sequence length limited to 100, and batch size of 128. No other dataset tuning task was introduced to the datasets we used for the experiment. The leaked archive dataset was used for performance testing purposes only.

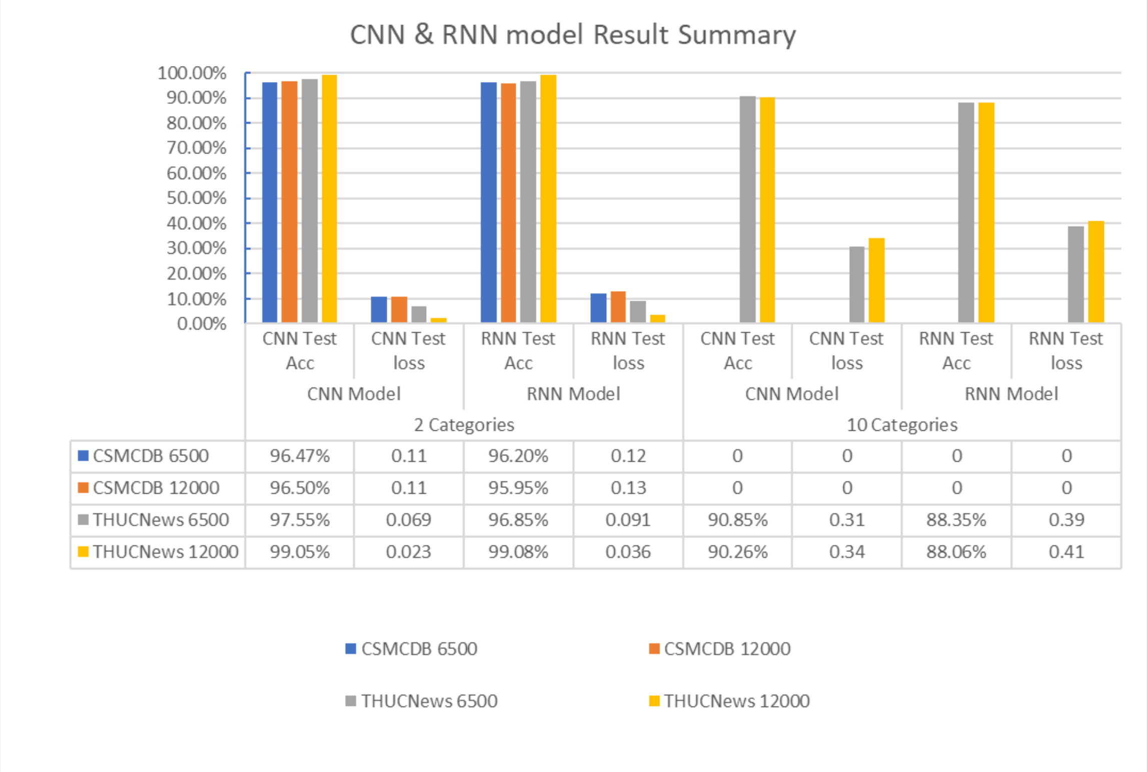
5.2 ENVIRONMENT REQUIREMENT

All results in this paper were fine-tuned on a single GPU with 6 GB of RAM. Running on a single GPU with less than 6 GB of RAM may encounter out-of-memory issues. The models are runnable on an environment with python 3.6.8, tensorflow-gpu 1.13.1, Keras 2.2.4, numpy 1.16.2 and scikit-learn 0.20.3.

CHAPTER 6

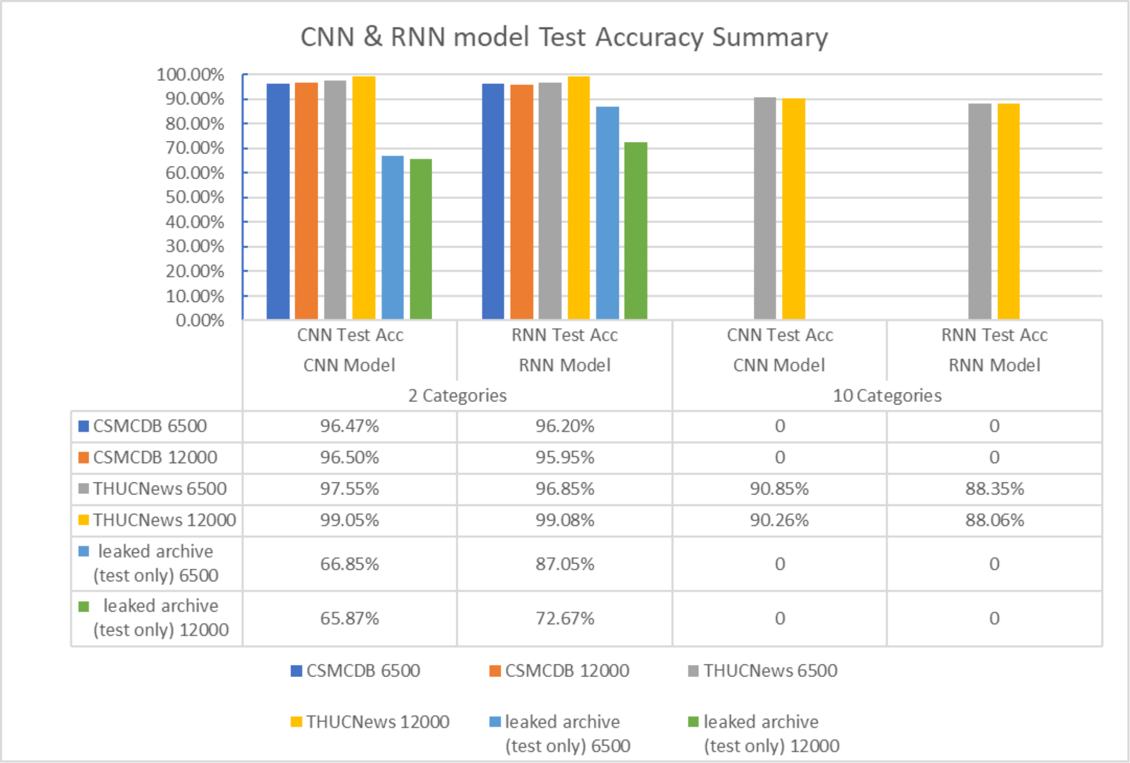
RESULTS AND DISCUSSION

The result summary of different datasets on our models is listed in Table A.2 within the Appendix. Since CSMCDB does not have more than two categories, there are six different paired categories of datasets involved in the experiment, which are clearly shown in Figure 6.1.



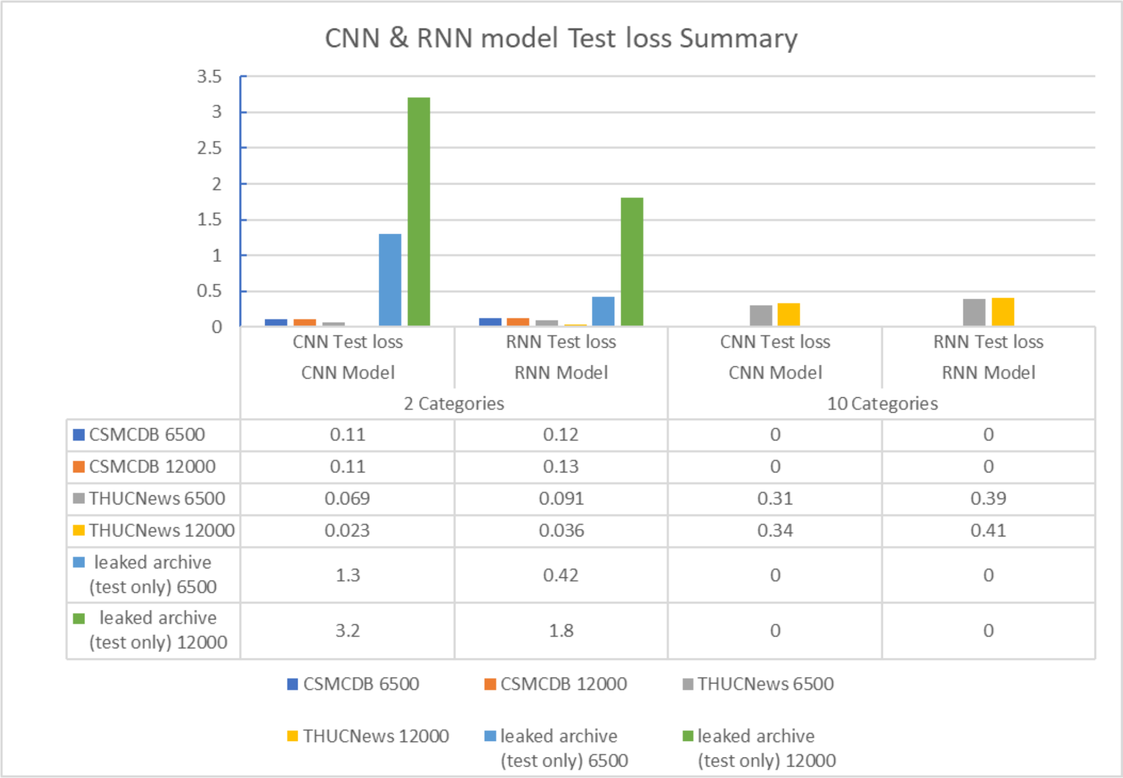
	2 Categories				10 Categories			
	CNN Model		RNN Model		CNN Model		RNN Model	
	CNN Test Acc	CNN Test loss	RNN Test Acc	RNN Test loss	CNN Test Acc	CNN Test loss	RNN Test Acc	RNN Test loss
🟦 CSMCDB 6500	96.47%	0.11	96.20%	0.12	0	0	0	0
🟧 CSMCDB 12000	96.50%	0.11	95.95%	0.13	0	0	0	0
⬜ THUCNews 6500	97.55%	0.069	96.85%	0.091	90.85%	0.31	88.35%	0.39
🟨 THUCNews 12000	99.05%	0.023	99.08%	0.036	90.26%	0.34	88.06%	0.41

Figure 6.1: CNN & RNN model result Summary



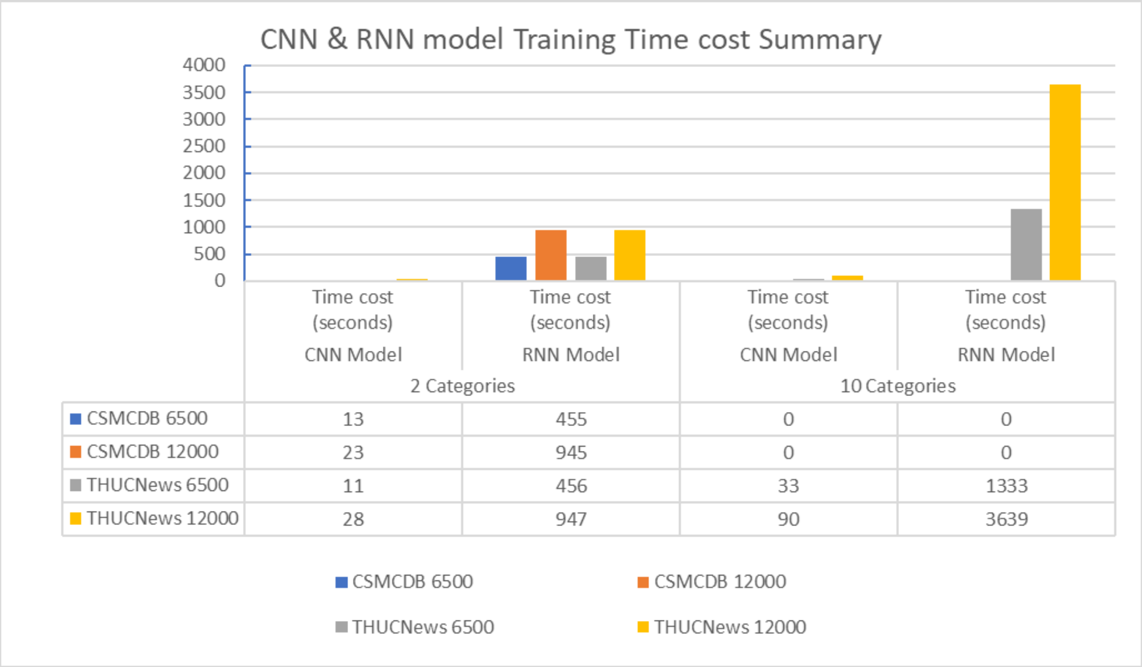
	CNN Test Acc	RNN Test Acc	CNN Test Acc	RNN Test Acc
	CNN Model	RNN Model	CNN Model	RNN Model
	2 Categories		10 Categories	
■ CSMCDB 6500	96.47%	96.20%	0	0
■ CSMCDB 12000	96.50%	95.95%	0	0
■ THUCNews 6500	97.55%	96.85%	90.85%	88.35%
■ THUCNews 12000	99.05%	99.08%	90.26%	88.06%
■ leaked archive (test only) 6500	66.85%	87.05%	0	0
■ leaked archive (test only) 12000	65.87%	72.67%	0	0

Figure 6.2: CNN & RNN model Test Accuracy Summary



	CNN Test loss	RNN Test loss	CNN Test loss	RNN Test loss
	CNN Model	RNN Model	CNN Model	RNN Model
	2 Categories		10 Categories	
■ CSMCDB 6500	0.11	0.12	0	0
■ CSMCDB 12000	0.11	0.13	0	0
■ THUCNews 6500	0.069	0.091	0.31	0.39
■ THUCNews 12000	0.023	0.036	0.34	0.41
■ leaked archive (test only) 6500	1.3	0.42	0	0
■ leaked archive (test only) 12000	3.2	1.8	0	0

Figure 6.3: CNN & RNN model Test loss Summary



	2 Categories		10 Categories	
	Time cost (seconds) CNN Model	Time cost (seconds) RNN Model	Time cost (seconds) CNN Model	Time cost (seconds) RNN Model
🟦 CSMCDB 6500	13	455	0	0
🟧 CSMCDB 12000	23	945	0	0
◼ THUCNews 6500	11	456	33	1333
🟨 THUCNews 12000	28	947	90	3639

Figure 6.4: CNN & RNN model Training time cost Summary

Avg. F1	CNN	RNN
CSMCDB	0.96488	0.96076
THUCNews	0.94375	0.930265
leaked archive	0.63269	0.79057

Table 6.1: Average F1-Measure of two models

First, we evaluate the accuracy performance of proposed models on all datasets. Several interesting results are shown in Figure 6.2. 1) In the two-categories test cases, the CNN model performs slightly better than the RNN model; all THUCNews datasets perform slightly better than the CSMCDB do. We believe that this result is related to the vocabulary of a single piece of data. Each content in THUCNews is a piece of news, and the content usually contains more than one sentence. However, each comment in CSMCDB is usually concise, and it is rare to have more than one sentence. Here, we can say that in a text classification task, the longer content could likely increase the accuracy of the training result. 2) The two-categories groups perform better than the results of the ten-categories groups. This difference in results is because training difficulty is proportional to the number of categories for classification jobs.

As shown in Figure 6.3, the test loss values indicate that longer content could potentially reduce the difficulty of classifier training jobs.

All models are running on the same hardware with GPU enabled, as shown in Figure 6.4. For two-categories groups, overall time cost on the RNN model is equal to 37 times the overall time cost on the CNN model. For ten-categories groups, the rate rises to 40. These results indicate that the proposed RNN model is less time efficient than the CNN model. CNN can fully parallelize calculations, whereas RNN/LSTM can only serialize when expanded along the time dimension. This limitation might be the root reason why the RNN model runs slower than the CNN model.

True positive (TP)	False positive (FP)
False negative (FN)	True negative (TN)

Table 6.2: Truth Table

When dealing with classification tasks, there are four different cases, as shown in Table 6.2. True positive (TP) is the set of successfully predicted positive cases. False positive (FP) is the set of unsuccessfully predicted positive cases. False negative (FN) is the set of unsuccessfully predicted negative cases. True negative (TN) is the set of successfully predicted negative cases.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

In addition to accuracy, additional measurements (precision, recall, and F1-measure) help us evaluate how the model works.

$$\text{Precision(P)} = \frac{TP}{TP + FP} \tag{6}$$

$$\text{Recall(R)} = \frac{TP}{TP + FN} \tag{7}$$

$$\text{F} = \frac{(a^2 + 1) * P * R}{a^2 * (P + R)} \tag{8}$$

$$\text{F1} = \frac{2 * P * R}{P + R} \tag{9}$$

Precision (P) is the fraction of successfully predicted positive elements among all predicted positive elements. Recall (R) is the fraction of successfully predicted positive elements among the total number of positive elements. If the prediction result contains 0 FP, the precision is equal to 1. If the prediction result contains 0 FN, the recall is equal to 1. In practice, P and R indicators sometimes have contradictory situations, so they must be considered comprehensively. The most common method is the F-measure (also known as F-score). The most common F1-measure We use in this thesis is when parameter a = 1. According to Table 6.1, the proposed CNN model has a 0.96488 of average F1 on CSMCDB, and the proposed RNN model has a 0.96076 of average F1 on CSMCDB. In other words, the proposed models are well fitted to the CSMCDB dataset with classification requests.

Tested with the leaked archive, the proposed CNN model has a 0.63269 of average F1, and the proposed RNN model has a 0.79057 of average F1. These results imply that the RNN model we proposed has better generalization ability compared to the

CNN model. Our RNN model (average F1 = 0.79057) has achieved a better performance compared with Miller’s work (0.68) [4].

As the results of this project demonstrate, character-level word embedding performs well in text classification tasks. The proposed two models using neural networks to detect the Fifty Cent Army comments trained by CSMCDB are effective, with a high accuracy rate of 95.95%. In performance tests of the proposed models on the leaked archive, accuracy dropped to 65.87%-87.05% with acceptable average F1 values.

CHAPTER 7

RELATED WORK

Numerous articles conducted by other researchers have preceded this thesis. “International knowledge and domestic evaluations in a changing society: The case of China” from Haifeng Huang provided detailed information about the background of the Fifty Cent Army. Gary King’s work [3] has constructed a model that can map multiple patterns to identify the Fifty Cent Army members other than those implied in the leaked emails. Miller also performed substantial work to develop a mechanism [4] that can automatically collect, identify, and analyze the Fifty Cent Army with outstanding accuracy, using IP addresses combined with Weibo accounts and posts to perform the work effectively. However, none of these works considered using deep learning methods to achieve the same or even better results. Extraordinary work has also been conducted with NLP text classification. The paper “Character-level convolutional networks for text classification” from Xiang Zhang, Junbo Zhang, and Yann LeCun [19] proposed a classification model that no longer considers words, phrases or sentences, or semantic and grammatical structure as its aspects for analysis. The model can extract high-level features simply from the character-level aspect. Such an approach has two major advantages. First, it does not require pre-trained word vectors and grammatical sentence structures, which can lead to fewer computation requirements. Second, it can be easily extended to other languages since it has nothing to do with word vectors and grammatical sentence structures. Thus, language can also be considered as a kind of signal that is no different from other signal types. “Convolutional neural networks for sentence classification” from Yoon Kim [23] described an example using CNN to extract features with pre-trained word vectors. Richard Socher et al. [20] introduced an RNN architecture that can outperform convincible results in sentence classification tasks.

CHAPTER 8

CONCLUSION AND FUTURE WORK

In this thesis, we outline a method with two models for detecting government astroturfers with comments only. Based on character-level word embedding, convolutional neural networks, and recurrent neural networks, our proposed models that were validated using the CSMCDB dataset labeled by human experts. Overall detection accuracy on the CSMCDB dataset is over 95.95%. Although the testing accuracy dropped slightly when testing the two models with the leaked archive, the bias is acceptable since the leaked archive contains minor errors and several identification problems. The saved model and weights can be implemented with multiple platforms as applications for future research requirements. In other words, applying proposed convolutional neural networks is an effective way to detect government astroturfers. For short content classification tasks in general, the proposed CNN model has a slight advantage over the performance of the RNN model.

Building upon this project, we will continue to experiment to find a reasonable use of simhash for data deduplication and optimization of results. This database CSMCDB is not the final form. Thus, we will continue to expand the database so that it can be kept accurate and up to date. Although this paper does not use Chinese word segmentation such as jieba for dictionary construction, we have reason to believe that the proposed models will have a better effect after a reasonable set of stop words and word segmentation is introduced. The models might also perform better if LSTM+CNN is used as a new model on the same dataset. Furthermore, the Chrome plug-in for real-time prediction on social networks was not completed due to time limitations, and the project will be perfected and applied to more aspects in the future, for example, classification for multiple labels, sentiment classification for text, suicide prevention, etc.

BIBLIOGRAPHY

[1] Huang, Haifeng. "International Knowledge and Domestic Evaluations in A Changing Society: The Case of China." American Political Science Review 109.3 (2015): 613-634.

[2] LAGERKVIST*, J. O. H. A. N. "Internet Ideotainment in The PRC: National Responses to Cultural Globalization." Journal of Contemporary China 17.54 (2008): 121-140.

[3] King, Gary, Jennifer Pan, and Margaret E. Roberts. "How The Chinese Government Fabricates Social Media Posts for Strategic Distraction, not Engaged Argument." American Political Science Review 111.3 (2017): 484-501.

[4] Miller, Blake Andrew Phillip. "Automatic Detection of Comment Propaganda in Chinese Media." Available at SSRN (2016).

[5] "Scrapy" https://scrapy.org, September, 2018.

[6] Charikar, Moses S. "Similarity Estimation Techniques from Rounding Algorithms." Proceedings of The Thirty-Fourth Annual ACM Symposium on Theory of Computing. ACM, 2002.

[7] Manku, Gurmeet Singh, Arvind Jain, and Anish Das Sarma. "Detecting Near-Duplicates for Web Crawling." Proceedings of The 16th International Conference on World Wide Web. ACM, 2007.

[8] Kendall, Maurice G., and B. Babington Smith. "The Problem of m Rankings." Annals of Mathematical Statistics (1939).

[9] Han, Rongbin. "Manufacturing Consent in Cyberspace: China's "Fifty-Cent Army"" Journal of Current Chinese Affairs [Online], 44 2015: 105-134

[10] GB2312-80 "Basic Collection of Chinese Character Codes for Information Exchange" http://www.moe.gov.cn/ewebeditor/uploadfile/2015/01/12/20150112161650162.pdf, October 2018.

[11] Graves, Alex. Supervised Sequence Labeling with Recurrent Neural Networks. Springer.

[12] "Tensorflow" https://www.tensorflow.org/ , September 2018.

[13] Hinton, Geoffrey E., et al. "Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors." arXiv Preprint arXiv:1207.0580 (2012).

[14] Srivastava, Nitish, et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." The Journal of Machine Learning Research 15.1 (2014): 1929-1958.

[15] Nair, Vinod, and Geoffrey E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines." Proceedings of The 27th International Conference on Machine Learning (ICML-10). 2010.

[16] "Leaked Email archive" https://xiaolan.me, September 2018.

[17] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521.7553 (2015): 436.

[18] Collobert, Ronan, and Jason Weston. "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning." Proceedings of The 25th International Conference on Machine Learning. ACM, 2008.

[19] Zhang, Xiang, Junbo Zhao, and Yann LeCun. "Character-Level Convolutional Networks for Text Classification." Advances in Neural Information Processing Systems. 2015.

[20] Socher, Richard, et al. "Parsing Natural Scenes and Natural Language with Recursive Neural Networks." Proceedings of The 28th International Conference on Machine Learning (ICML-11). 2011.

[21] "Embedding" https://www.tensorflow.org/guide/embedding, October 2018.

[22] Rong, Xin. "Word2vec Parameter Learning Explained." arXiv Preprint arXiv:1411.2738 (2014).

[23] Kim, Yoon. "Convolutional Neural Networks for Sentence Classification." arXiv Preprint arXiv:1408.5882 (2014).

[24] "Chinese Text Classification" https://github.com/gaussic/text-classification-cnn-rnn, September 2018.

[25] Norman, Geoff. "Likert Scales, Levels of Measurement and The “Laws” of Statistics." Advances in Health Sciences Education 15.5 (2010): 625-632.

[26] "SPSS Software" https://www.ibm.com/analytics/spss-statistics-software, March 2019.

[27] Hu, Baotian, et al. "Convolutional Neural Network Architectures for Matching Natural Language Sentences." Advances in Neural Information Processing Systems. 2014.

[28] "THUCTC" http://thuctc.thunlp.org, October 2018.

[29] Ge, J., and U. Gretzel. "A new Cultural Revolution: Chinese Consumers’ Internet and Social Media Use." Advances in Social Media for Travel, Tourism and Hospitality: New Perspectives, Practice and Cases. Ashgate Publishing, Ltd., NY (2018): 102-118.

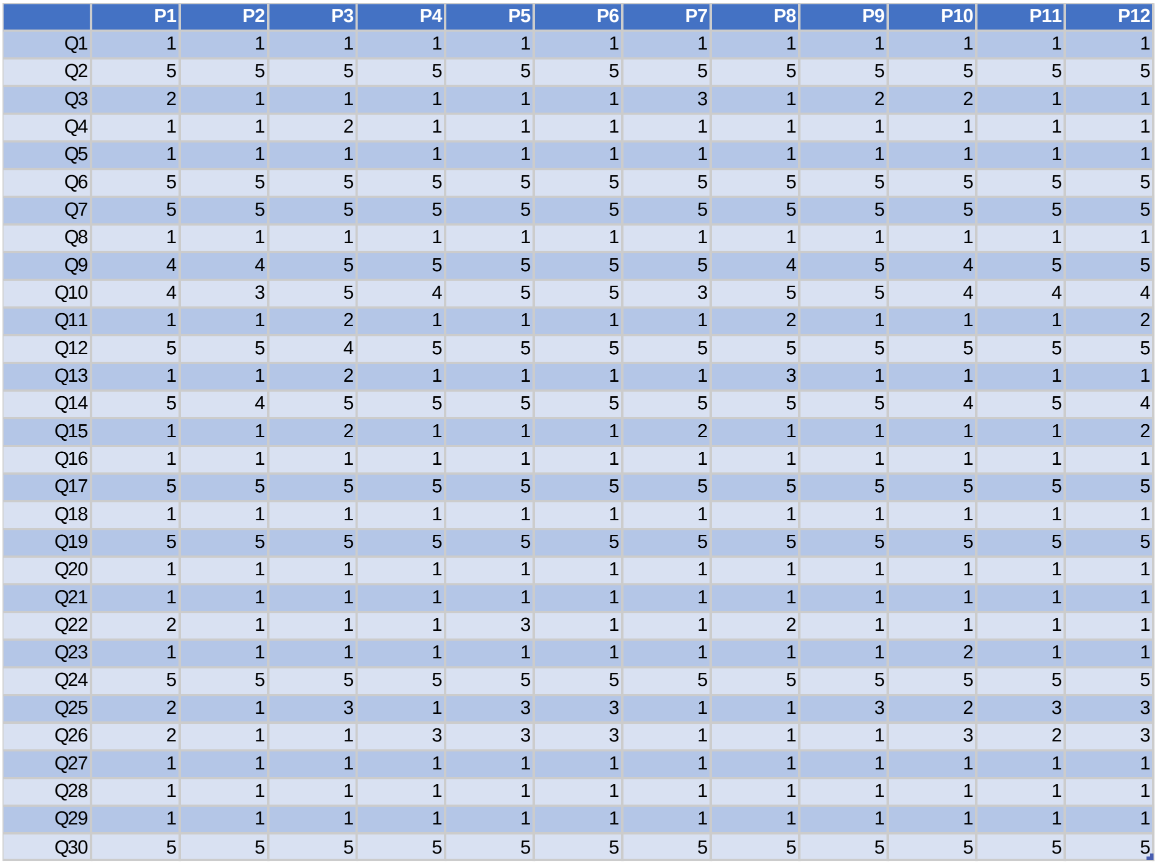
[30] Yang, Kejing. The Door Is Closed, but Not Locked: China's VPN Policy. Diss. Georgetown University, 2017.

[31] Lewis, John W., and Xue Litai. "Social Change and Political Reform in China: Meeting The Challenge of Success." The China Quarterly 176 (2003): 926-942.

[32] Chen, Cheng, et al. "Battling The Internet Water Army: Detection of Hidden Paid Posters." 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013). IEEE, 2013.

APPENDIX A

QUESTIONNAIRE FOR EVALUATOR CONSISTENCY TEST

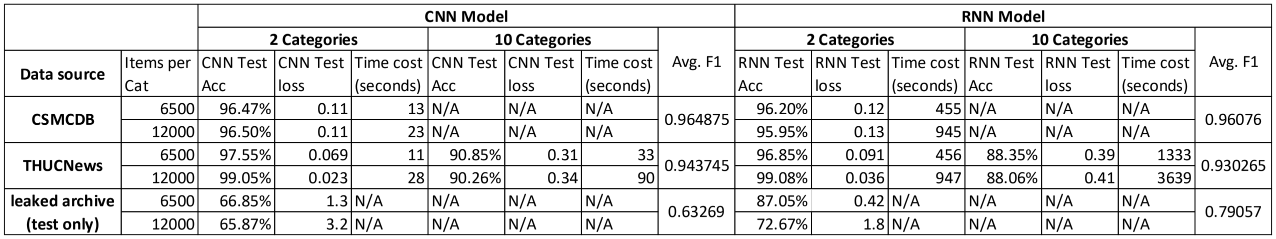


	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
Q1	1	1	1	1	1	1	1	1	1	1	1	1
Q2	5	5	5	5	5	5	5	5	5	5	5	5
Q3	2	1	1	1	1	1	3	1	2	2	1	1
Q4	1	1	2	1	1	1	1	1	1	1	1	1
Q5	1	1	1	1	1	1	1	1	1	1	1	1
Q6	5	5	5	5	5	5	5	5	5	5	5	5
Q7	5	5	5	5	5	5	5	5	5	5	5	5
Q8	1	1	1	1	1	1	1	1	1	1	1	1
Q9	4	4	5	5	5	5	5	4	5	4	5	5
Q10	4	3	5	4	5	5	3	5	5	4	4	4
Q11	1	1	2	1	1	1	1	2	1	1	1	2
Q12	5	5	4	5	5	5	5	5	5	5	5	5
Q13	1	1	2	1	1	1	1	3	1	1	1	1
Q14	5	4	5	5	5	5	5	5	5	4	5	4
Q15	1	1	2	1	1	1	2	1	1	1	1	2
Q16	1	1	1	1	1	1	1	1	1	1	1	1
Q17	5	5	5	5	5	5	5	5	5	5	5	5
Q18	1	1	1	1	1	1	1	1	1	1	1	1
Q19	5	5	5	5	5	5	5	5	5	5	5	5
Q20	1	1	1	1	1	1	1	1	1	1	1	1
Q21	1	1	1	1	1	1	1	1	1	1	1	1
Q22	2	1	1	1	3	1	1	2	1	1	1	1
Q23	1	1	1	1	1	1	1	1	1	2	1	1
Q24	5	5	5	5	5	5	5	5	5	5	5	5
Q25	2	1	3	1	3	3	1	1	3	2	3	3
Q26	2	1	1	3	3	3	1	1	1	3	2	3
Q27	1	1	1	1	1	1	1	1	1	1	1	1
Q28	1	1	1	1	1	1	1	1	1	1	1	1
Q29	1	1	1	1	1	1	1	1	1	1	1	1
Q30	5	5	5	5	5	5	5	5	5	5	5	5

Table A.1: Questions and results for evaluator consistency test

APPENDIX B

TRAINING TEST SUMMARY



		CNN Model							RNN Model						
		2 Categories			10 Categories				2 Categories			10 Categories			
Data source	Items per Cat	CNN Test Acc	CNN Test loss	Time cost (seconds)	CNN Test Acc	CNN Test loss	Time cost (seconds)	Avg. F1	RNN Test Acc	RNN Test loss	Time cost (seconds)	RNN Test Acc	RNN Test loss	Time cost (seconds)	Avg. F1
CSMCDB	6500	96.47%	0.11	13	N/A	N/A	N/A	0.964875	96.20%	0.12	455	N/A	N/A	N/A	0.96076
	12000	96.50%	0.11	23	N/A	N/A	N/A		95.95%	0.13	945	N/A	N/A	N/A	
THUCNews	6500	97.55%	0.069	11	90.85%	0.31	33	0.943745	96.85%	0.091	456	88.35%	0.39	1333	0.930265
	12000	99.05%	0.023	28	90.26%	0.34	90		99.08%	0.036	947	88.06%	0.41	3639	
leaked archive (test only)	6500	66.85%	1.3	N/A	N/A	N/A	N/A	0.63269	87.05%	0.42	N/A	N/A	N/A	N/A	0.79057
	12000	65.87%	3.2	N/A	N/A	N/A	N/A		72.67%	1.8	N/A	N/A	N/A	N/A	

Table A.2: Training test summary