Spring 5-20-2020

Sentiment Analysis for Troll Activity Detection on Sina Weibo

Zidong Jiang
San Jose State University

Recommended Citation
Jiang, Zidong, "Sentiment Analysis for Troll Activity Detection on Sina Weibo" (2020). *Master's Projects*. 923.
DOI: https://doi.org/10.31979/etd.vt4x-uaeu
https://scholarworks.sjsu.edu/etd_projects/923

Sentiment Analysis for Troll Activity Detection on Sina Weibo

A Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Zidong Jiang

May 2020

The Designated Project Committee Approves the Project Titled

Sentiment Analysis for Troll Activity Detection on Sina Weibo

by

Zidong Jiang

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

May 2020

Dr. Mark Stamp	Department of Computer Science
Dr. Fabio Di Troia	Department of Computer Science
Dr. Chris Pollett	Department of Computer Science

ABSTRACT

Sentiment Analysis for Troll Activity Detection on Sina Weibo

by Zidong Jiang

The impact of social media on the modern world is difficult to overstate. Virtually all companies and public figures have social media accounts on popular platforms such as Twitter and Facebook. In China, the micro-blogging service provider Sina Weibo is the most popular such service. To overcome negative publicity, Weibo trolls the so called Water Army can be hired to post deceptive comments.

In recent years, troll detection and sentiment analysis have been studied, but we are not aware of any research that considers troll detection based on sentiment analysis. In this research, we focus on troll detection via sentiment analysis with other user activity data gathered on the Sina Weibo platform, where the content is mainly in Chinese. We implement techniques for Chinese sentence segmentation, word embeddings, and sentiment score calculations. We employ the resulting techniques to develop and test a sentiment analysis approach for troll detection, based on a variety of machine learning strategies. Experimental results are generated, analyzed and the troll detection model we proposed achieved 89% accuracy for the dataset presented in this research. A Chrome extension is presented that implements our proposed technique, which enables real-time troll detection and troll comments filtering when a user browses Sina Weibo tweets and comments.

ACKNOWLEDGMENTS

I would like to appreciate Dr. Mark Stamp for accepting me as his master project student and giving me valuable guidance during completion of this master project. I also want to give my appreciation to Dr. Chris Pollett and Dr. Fabio Di Troia for their precious time and advice.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Social media plays a significant role in the ongoing development of the Internet since people tend to acquire more information from social media platforms such as Twitter and Facebook. Trolls have been identified as one of the most challenging problems from social media platforms. Trolls can be hired to publish misleading comments in an effort to affect public opinions of events or people, or even to negatively influence the economy of a country.

Weibo is a widely used micro-blogging social media platform by Sina that is popular in China. A majority of Weibo posts are written in Chinese and, like Twitter, most posts published on Weibo is short (until recently, there was a 140 character limit). With the number of daily active user in excess of 200 million in 2019, Weibo is one of the largest social media platform in China. Weibo is a social media platform based on weak-relationship, where a user can share and post content public to all Weibo users , rather than just to their friends on Weibo. Therefore, many celebrities, businesses, and Internet influencers all over the world register as Weibo users to expand their exposure to the Chinese public. Weibo has became a platform where government and businesses can communicate more efficiently with the general public. Additionally, people can interact directly with celebrities on Weibo platform.

The Chinese Water Army refers to a group of people who get hired to post deceptive comments on Weibo. Such troll activity on Weibo platform is difficult to detect, in part due to the unsegmented characteristic of Chinese sentences, that is, Chinese sentences can be segmented in different ways to yield different meanings.

Recent research has shown that hidden Markov models (HMM) are effective for sentiment analysis for English text [2]. Chinese word segmentation can also be accomplished using HMMs [3, 4, 5]. The primary goal of this research is to use HMM, Word2Vec and other techniques to perform word segmentation and sentiment analysis on Sina Weibo "tweets" written in Chinese for the purpose of detecting potential troll activity. We compare Word2Vec and HMM for word conversion, and we compare HMM and Naïve Bayes for sentiment analysis.

Again, a key point of this research is to apply sentiment analysis to the troll detection problem. In this research, we first crawled our training and testing data from the Sina Weibo and Tencent Weibo platforms. Using HMM-based the Chinese sentence segmentation model in [3], we pre-processed all the posts and comments into lists of words. Then, following the approach in [6], we construct a Word2Vec similarity scoring matrix based on the word list that we have generated. We need to then determine a baseline of sentiment from the corpus that we collected from Sina Weibo and Tencent Weibo.

For sentiment analysis part, we use the Word2Vec words embedding method to calculate sentiment scores for each sentence. We used extracted features from Weibo comments to feed each HMM model in order to train for each emotion and use the trained model to categorize emotion for each comment. We will use XGBoost model to aggregate the sentiment analysis result with user activity data to build the troll detection model and, as another point of comparison, we will experiment with a third approach based on Support Vector Machine.

Finally, we will create and test a Chrome extension that employs a model that we have develop. This enables real-world users to easily detect potential troll activity on the Weibo platform.

The remainder of this report is organized as follows. In Section 2 we will discuss relevant background topics, including troll activity, machine learning for troll detection, and sentiment analysis. In Section 3, we will specify the data source and gathering method for all parts for the project. In Section 4, we will illustrate different machine learning methodologies applied to this project as well as mathematical background. In Section 5, we will explain how individual parts of this project are implemented and put together as well as test results for the developed models. Last but not least, in Section 6, we will give a summary for this project as well as talk about limitation and future development.

CHAPTER 2

Background

2.1 What is a Troll?

Troll users are a group of Internet users who publish misleading, offensive or trivial following-up contents in the online communities. The content of a troll posting generally falls into one of several categories. It may consist of an apparently foolish contradiction of common knowledge, a deliberately offensive insult to the readers of a newsgroup or mailing list, or a broad request for trivial follow-up postings. The result of such postings is frequently a flood of angry responses. In some cases, the follow-up messages posted in response to a troll can constitute a large fraction of the contents of a newsgroup or mailing list for as long as several weeks. These messages are transmitted around the world to thousands of computers, wasting network resources and costing money for people who pay to download email or receive Usenet news. Troll threads also frustrate people who are trying to carry on substantive discussions [7].

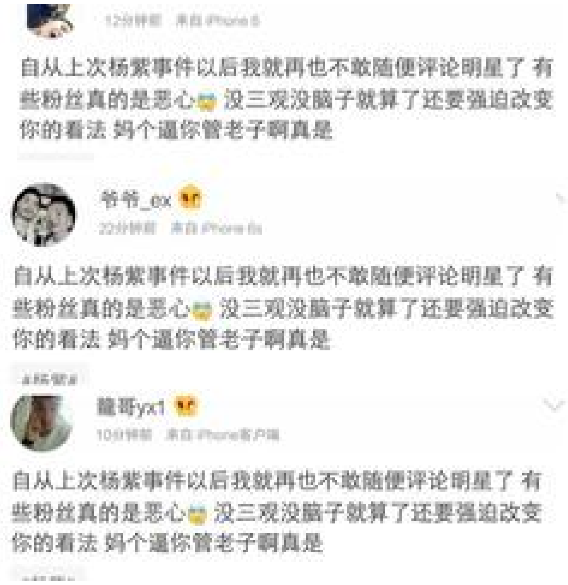
One example showing the influence from troll users is the Russian troll scandal happened in the 2016 U.S. Presidential election. The suspected troll users on Twitter, Reddit are accused on posting repetitive, destructive comments and contents towards the Democratic party candidates. Most of those accounts are later find out controlled by Russian-government–sponsored propaganda operation designed in part to help Donald Trump get elected president [8].

Troll activities on the Chinese Sina Weibo platform are first found around 2013. The initial group of troll users of around 20,000 people are managed in 50 OICQ chat groups by a person nicked name as Daxia. In the later years, troll PR became an online business on the Weibo platform. Trolls are hired by one business to publish negative comments towards their competitors or spread false anonymous good review or comments towards themselves. There are several PR events happened in Chinese online communities including “The of 360 vs QQ” happened between 2010 and 2014 are all related to the subtle promotion of trolls on Weibo. At that time, trolls were hired to post repeated, destructive comments towards the business competitors and by dominating the comment area tried to mislead public opinion. After 2015, Chinese government managed to set more strict control on speech in Internet and Sina Weibo developed a more sophisticated infrastructure to filter the repeated, destructive troll comments. Most of the troll activities then turned to provide service to promoting celebrities and companies.

2.2 Types of Troll

Troll users on the Weibo platform can be categorized by their source of content. The traditional troll users use automated fake accounts to post repeated messages to dominate the comment areas so that normal users can only see these repetitive comments shown in Figure 1 [9]. However, the Weibo platform has recently improved their infrastructure to block these repeated messages from different users using IP and proxy detection combined with message filters for repeated comments.

The new kind of troll users are more sophisticated. They are supervised by a management group who controls what, when and where they reply to on the Weibo platform. However, the details of the comments each troll account publishes are made by the individual troll users rather than copied from the management group. The management group only gives out the overall emotion trend which the comments should convey. Thus, contents made by troll users are repetitive but not monotonously repeated. This fact made the troll detection on Weibo even harder than before since all the comments are composed and published by real human users since the Weibo infrastructure cannot use IP and proxy detection with message filters to rule out non-repeated comments. Additionally, in the past, trolls are mostly hired to post



12分钟前 来自iPhone 6

自从上次杨紫事件以后我就再也不敢随便评论明星了 有些粉丝真的是恶心😱 没三观没脑子就算了还要强迫改变你的看法 妈个逼你管老子啊真是

爷爷_ex

22分钟前 来自iPhone 6s

自从上次杨紫事件以后我就再也不敢随便评论明星了 有些粉丝真的是恶心😱 没三观没脑子就算了还要强迫改变你的看法 妈个逼你管老子啊真是

#杨紫#

龍哥yx1

10分钟前 来自iPhone客户端

自从上次杨紫事件以后我就再也不敢随便评论明星了 有些粉丝真的是恶心😱 没三观没脑子就算了还要强迫改变你的看法 妈个逼你管老子啊真是

Figure 1: Screenshot of an actress Weibo comment area dominated by repeated comments from troll users

destructive comments to make negative effects towards competitors of their client. In recent years, trolls are mostly hired by companies and celebrities to make positive comments towards themselves.

2.3 Machine Learning for Troll Detection

2.3.1 Content Characteristic-based Troll Detection

Content based troll detection usually utilizes natural language processing in machine learning to categorize, analyse sentiment and emotion trend behind the text. This is accomplished by construct language processing model on comments and posts from Weibo platform, specifically the labelled comments with high polarity of emotion and repetitiveness. By applying sentiment analysis methods, we can easily filter comments with either high or low sentiment scores representing extreme positive or negative sentiments. This is accomplished by calculating word relevance at training stage. And by analyzing correlation between those filtered comments using word vectoring techniques such as Word2Vec with high polarity emotion. We can mark them as potential troll comments or pass the user information behind those comments to the next stage of troll detection model.

The key point of this research is to covert a sentiment analysis problem into a troll detection problem which few researches have done in a similar regime. Getting different features from text is a classic approach in the text mining studies. Zhao and Wang in [10] proposed a solution of combining emotional orientation and logistical regression in analyzing online comments from Amazon.com. By filtering the training dataset by text length, vocabulary complexity, text correlation with the product, sentiment similarity and transition words, the proposed model achieved 91.2% accuracy based on input dataset of pure comment text on 100 recursions of tests.

2.3.2 User Relationship Characteristic-based Troll Detection

Since there are less simple, repeated and destructive comments on Weibo. It is harder for content based troll detection mechanism to find out troll activities from sophisticated and logic comments. Therefore, utilizing user behavioral information behind comments to find out troll activity became another popular trend of troll detection. Like most social media platform, Weibo has numerous of user relationship qualification such as number of follower, number of following, user rank, number of original Weibo tweets. Also, trolls commonly make attack in a close time period under a tweet [11]. We can utilize this fact in combine with other user relationship information from Weibo platform to conduct troll detection. The advantage for ruling in user relationship data into troll detection will eliminate the potential false negative classification from content-based detection method.

2.3.3 Related Work

Across all the literature reviews conducted, the only research applied sentiment analysis into troll detection is proposed by Seah, et al [12]. They applied domain adaptation techniques to recursive neural tensor network sentiment analysis model (RNTN) to detect repetitive, destructive and deceptive forum posts mainly written in Colloquial Singapore English, achieving 78% accuracy with unannotated training data with their AdaptCo model. Seah, et al. set a very solid baseline for future research and suggest the best model conduct pure text sentiment analysis unsupervised with the potential amount of data being processed [12].

As for troll detection by user characteristic data, Zhang, et al. [11] proposed a Weibo troll detection solution based on the Bayesian model and genetic algorithm adding features such as the ratio of follower and following, average posts, favorite posts, and Weibo credibility. They achieved about 90% accuracy without the Weibo credibility property and 96% accuracy with that property. Liu, Wang and Long [13] used XGBoost to classify for fake Weibo posts based on features such as accounts' number of posts, description, gender, followers, followings and reposts content. With accuracy over 95%, their model achieved product level precision with the amount of feature feeding into the model. Both of these reviewed researches used data beyond Weibo post itself and achieved impressive results which can be used as a comparison in future research.

2.4 Machine Learning for Sentiment Analysis

2.4.1 Hidden Markov Model

The Hidden Markov Model is famous for its usage in pattern prediction and derive hidden state from known information. HMM is a stochastic model representing changing states where each the state in the future is solely depend on current state but not past state. By feeding a certain HMM model with observation sequence, we can calculate the probability of the transitions between each hidden state so as to find out the direct projection of observation to the hidden state. This process is called the Markov process and is illustrated in Figure 2. The notation used with HMMs is given in Table 1.

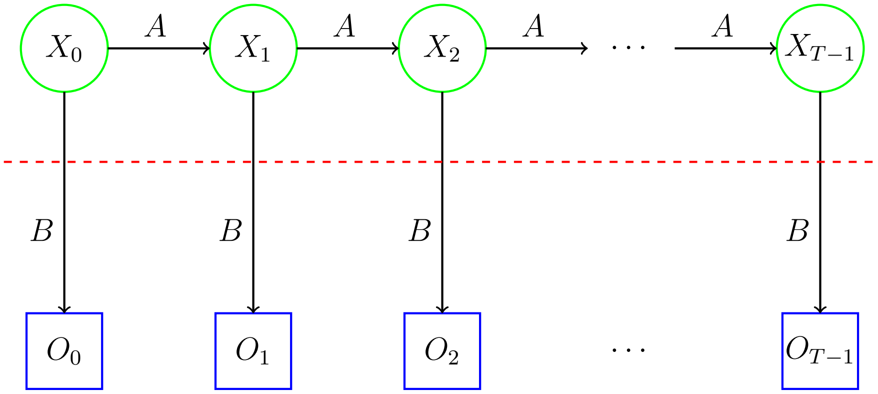


Figure 2: Hidden Markov model

Table 1: HMM notation

Notation	Description
T	Length of the observation sequence
N	Number of states in the model
M	Number of observation symbols
Q	Distinct states of the Markov process, $q_0, q_1, \ldots, q_{N-1}$
V	Possible observations, assumed to be $0, 1, \ldots, M-1$
A	State transition probabilities
B	Observation probability matrix
π	Initial state distribution
O	Observation sequence, $O_0, O_1, \ldots, O_{T-1}$

The matrix A is $N \times N$ and contains the transition probability of each state, while B is $N \times M$, with row i being the emission probability distribution for state i. The matrix π contains the initial state probability distribution.

Hidden Markov Model is widely used in machine learning regime to predict state change for random changing systems. The very basic usage such as English text property study and speech recognition [14]. Also, HMM is known for solving different cipher decryption application such as simple substitution cipher, homophonic cipher and famous Zodiac 408 cipher [15]. Recent researches applied HMM widely into malware detection including metamorphic malware [16], encrypted and polymorphic malware [17], where HMM is used to analyze the opcode from certain application file to detect different malware code sequence.

2.4.2 Sentiment Analysis Using HMM

Sentiment analysis is a classic topic in natural language processing by machine learning. The purpose of sentiment analysis is to find out the subjective attitude of the author or speaker when making a statement. One of the two models used to represent sentiment emotions is the categorical model which classifies the sentiment of emotion as anger, disgust, fear, joy, sadness, and surprise [18].

Related work in sentiment analysis includes [10], where a combination of emotional orientation and logistical regression is used to analyze online comments from Amazon.com. By filtering the training dataset by text length, vocabulary complexity, text correlation with product, sentiment similarity, and transition words, the proposed model achieved 91.2% accuracy. Our HMM-based sentiment model will also be compared to the result generated from an XGBoost model based on user activity [13], which achieved 93% accuracy. However, from our review of the literature, it appears that only [12] applies sentiment analysis to the troll detection problem. Specifically, in [12] domain adaptation techniques are applied to a recursive neural tensor network (RNTN) sentiment analysis model to detect repetitive, destructive, and deceptive forum posts, achieving 78% accuracy. The results in [12] serve as a baseline for our research.

Sentiment analysis is widely used for mining subjective information behind online posts. In [19], Kim, et al. proposed to use Hidden Markov Model with syntactic and sentiment information for the microblogs sentiment analysis on Twitter data. Different than classic approaches using n-grams and polarity lexicons, they have proposed to group words with similar syntactic and sentiment roles (SIG) then build HMM on these SIGs. Zhao and Ohsawa in their later paper [2] proposed a 2-dimensional Hidden Markov Model in analyzing Amazon online reviews in Japanese which illustrate an important method of converting segmented Japanese words into word vector by using Word2Vec. Feng and Durdyev in their latest class research, implemented 3 types of 4-class classification models (SVM, XGBoost, LSTM) for the aspect-level sentiment analysis of restaurant customers reviews in Chinese [6].

LSTM yields better F-1 score and accuracy than SVM and XGBoost which may be added as a comparison model into our future research. Further research should be based the baseline results presented by Liu, et al. that use a modified version of HMM as self-adaptive HMM [20]. These researches suggest that the best models convert segmented text into categorical sentiment uses similar words vectoring technique which converting words into mathematical vectors in calculating the sentiment score. In Figure 3, We specify the process of Chinese sentiment analysis we are going to implement in this project.

2.4.3 Chinese Word Segmentation

In order to handle text into correct sentiment, the first job is to correctly segment Chinese sentences existing in the micro-blog posts into meaningful lists of words. Chinese word segmentation is a classic research topic in the natural language processing regime. Chinese sentence has a non-segmented characteristic with no space in between words in sentences. Therefore, different segmentation can create different meaning of an exactly same looking sentence. Figure 4 gives an example of state changes in segmenting a Chinese sentence.

Special Interest Group for Chinese Language Processing (SIGHAN) is an organization specializing in developing and organizing the competition for the best Chinese

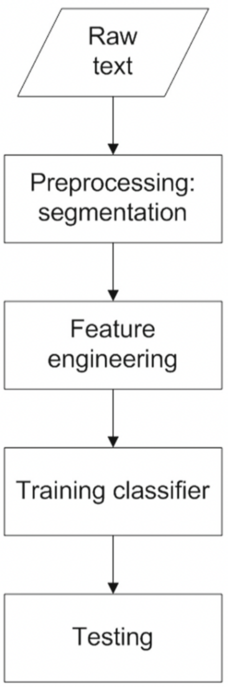
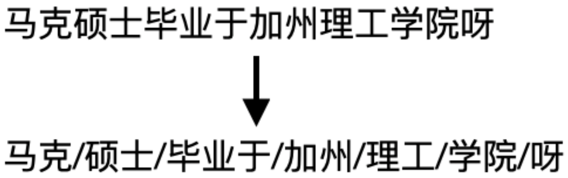


Figure 3: Sentiment analysis procedure for Chinese language



马克硕士毕业于加州理工学院呀

马克/硕士/毕业于/加州/理工/学院/呀

Figure 4: Sample Chinese sentence segmentation

segmentation model. As early as 2003, from the first SIGHAN bake-off event, Zhang, et al. proposed a word-based approach using Hierarchical Hidden Markov Model to form a Chinese lexical analyzer ICTCLAS [21]. Later in 2005, Masayuki, et al. presented three word segmentation modules including character-based tagging classifier technique based on Support Vector Machine as classifying method, maximum expropriation Markov model and conditional random fields [22]. All the modules are based on previously proposed methods with a different combination of out-of-vocabulary (OOV) extraction techniques being used.

Character-based models perform better than word-based modes and soon been adopted by a lot of researchers in the study of Chinese segmentation. Wang, Zong, and Su [23] highlighted the out-of-vocabulary (OOV) technique for word extraction performs poor for in-vocabulary (IV) words which has been recorded before. They proposed a generative model that performs well over both OOV and IV words and achieved comparable results on the SIGHAN datasets which previous researches have tested their models upon. Chen, Chang, and Pei in their latest research [24] reported the use of Gibbs sampling in the combination of both word-based hierarchical process model and character-based Hidden Markov Model. Their solution achieves better performance (F-score) than the state-of-the-art models at the time and further research can be conducted to handle the more simplified sentence people encounter over the Weibo platform with multiple languages and hash-tagging situation.

2.4.4 Word2Vec

Word2Vec [1] is one of the word embedding technique in natural language processing field. Word embedding is all about project a word into a mathematical space so that we can easily quantify the correlations be multiple words. Figure 5 [25] provides basic analogies that word embedding between a few different words.

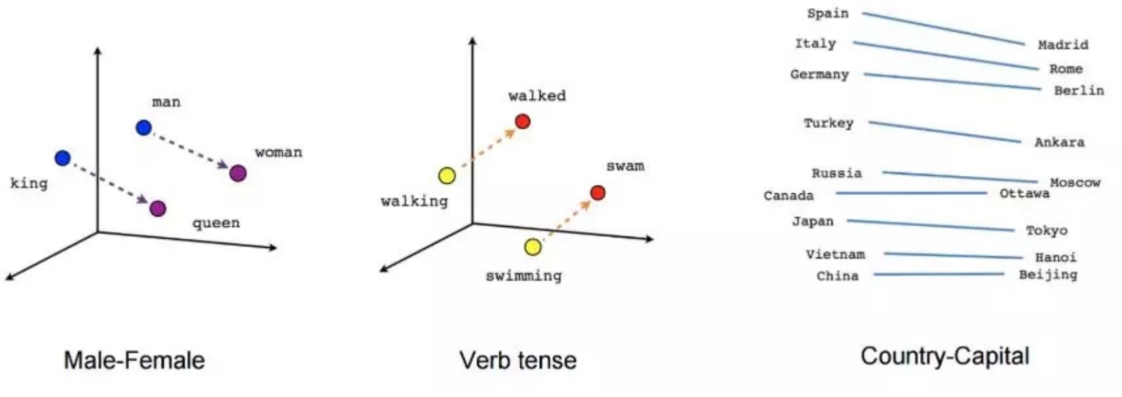


Figure 5: Illustration of Word2Vec [1]

Word2Vec model creates a mapping from word to its assigned property. It could be part of speech or its associated emotion or context. Word2Vec is based on skip-gram neural network, where we have one word as a input and underlying multiple context as output as illustrated in Figure 6.

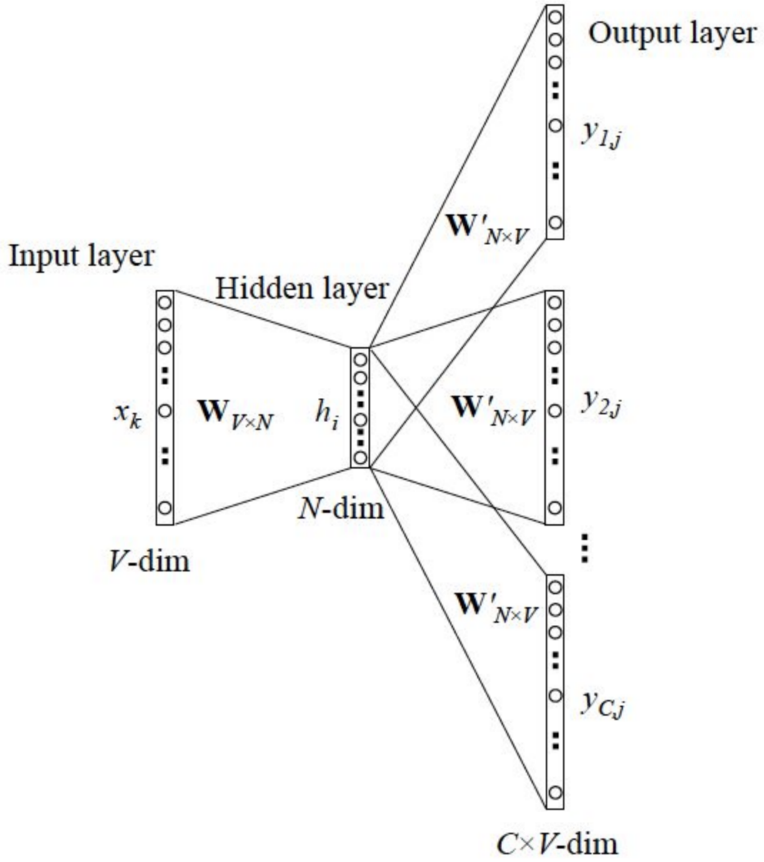


Figure 6: Skip-gram network for word X map to multiple Y context

CHAPTER 3

Dataset

We acquired and generated multiple datasets for different parts of this project such as Chinese segmentation dataset, sentiment analysis dataset, Weibo comment and user dataset for troll detection model training.

3.1 Chinese Segmentation Dataset

For Chinese sentence segmentation part, we acquired the state-of-the-art Chinese segmentation dataset from SIGHAN 2005 Competition for Chinese sentence processing [26]. This dataset includes training, testing corpora as well as a golden standard validation corpora. The training corpora includes approximately 860,000 segmented Chinese sentences. Most of the source sentences are from newspapers and published books. The test set includes about 22,000 sentences from similar source but not segmented. The validation set contains the standard segmentation of all the sentences from test set. Table 2 gives basic statistics for the source and number of sentences from training, testing and validation in the SIGHAN 2nd Bakeoff corpora.

Table 2: Chinese segmentation dataset statistics from SIGHAN 2nd Bakeoff 2005

Source	Training	Testing
Academia Sincia	708,953	14,432
Peking University	19,056	1,944
City University of Hong Kong	53,019	1,492
Microsoft Research Asia	86,924	3,985

3.1.1 Features and Extraction

Based on the Character-based generative model proposed in [3], the features from training data are the positions of each character located in each segmented word. Beginning characters are marked as state B, middle characters are marked as state M and ending characters are marked as state E. One character words are marked as state S. The training data already includes all the segmented sentences line by line by new-line character and words in the sentence segmented by empty space. Therefore, every character in every segmented word followed by space is considered as the beginning character. All the characters following beginning character are middle characters. The character followed by space is considered ending character. Proper segmentation should appear between ending and beginning character as well as between single character words and other words. Table 6 demonstrate the proper state mark to the sentence segmented in Figure 4.

3.2 Sentiment Analysis Dataset

For the sentiment analysis part, we acquired the sentiment training dataset from Python SnowNLP package [27] to get started, which includes 16,548 sentences with positive sentiment 18,574 sentences with negative sentences. The source for the SnowNLP sentiment analysis dataset are from Chinese online shopping, movie and book reviews. However, those more formally composed corpora as seen on online shopping reviews might not accurately represents the tweets and comments content appearing in Weibo. Also, since there is no public datasets of Weibo posts and comments realted to emotions, we managed to crawl 5 million Sina Weibo posts to train the sentiment model better with speech and slang more commonly seen on Weibo.

Lastly, we created a corpus of total of 2,325,644 sentences in positive sentiment training set and 960,899 sentences in negative training set after removing duplicated tweets and tweets with emoji or other contents unable to be categorized.

From all the Weibo corpora source we crawled, we manually filtered out 500 tweets for each of the 6 emotions representing happiness, surprise, fear, anger, disgust and sadness [18] for the training HMM on each emotion. For each comment entry, we implement content processing functions with Pandas package in Python to remove stopping words, numbers, any nonsense emoji and single word. The language detection method was implemented to detect non-Chinese comments and translate English comments into Chinese using Translator package by Google Translate. We abandoned all the other comments composed in languages other than Chinese and English, which contribute to a very small portion of the complete dataset. Also, we removed the special tagging for pure re-posting and replying from comments since comments with special tagging contribute to a large portion of comments data. The special tagging can only reveal constant sentiment score which will not contribute to accuracy of final result.

3.2.1 Features and Extraction

For the positive and negative sentiment analysis, we used words embedding method called Word2Vec to calculate the words sentiment scores after segmenting the Weibo comments sentences into list of word. For sentiment analysis based on 6 basic emotions, we adopted three features introduced in [20] to feed in the HMM sentiment classification model listed below:

- Mutual Information (MI). Mutual Information can provide correlation between two items. In this research, MI is used to provide relevance between words and sentences and certain emotion. The formula for MI representing correlation between emotion e and text t is written as

$$\mathrm{MI}(t, e) = \log \frac{P(t \,|\, e)}{P(e)}.$$

- Chi-Square (CHI). In this project, CHI measures the dependence between emotion e and text t. The higher the CHI value means text t is more dependent on emotion e. Expression for CHI is written as

$$\mathrm{CHI}(t, e) = \frac{(AD - BC)^2 \cdot N}{(A + B)(C + D)(A + C)(B + D)},$$

where A as presence of word t in a comment with e; B as presence of word t in a comment with emotion not as e; C as absence of word t in a comment with e; D as absence of word t in a comment with emotion not as e; N as total number of comments.

- Term Frequency Inverse Document Frequency (TF-IDF). With TF-IDF, if a word has high frequency in one emotion but low frequency in other emotion, then this word can be a feature word to determine this emotion. Expression for TF-IDF is written as

$$\text{TF-IDF}(t, e) = \frac{N_{e,t}}{\sum_k N_{k,t}} \log\left(\frac{N}{n_e} + 0.01\right),$$

where $N_{e,t}$ as number of times word t appears in emotion e; N as total number of comments; n_t as number of comments that t appears.

3.3 Troll Detection Dataset

After a few weeks of experiment, we started to realized the multi-prospect of emotion troll comments on Weibo and the limitation based on [12] and [20]. Therefore we started to find out more user information related features that we can obtain during mining for the Weibo comments data. Since we used JSON packet returned from REST call to Weibo mobile site [28], there are user related information returned as well as contents of comments. We managed to get the following user information listed in Table 3. We also used a small dataset of 673 normal users and 75 troll users from a Kaggle data source [29].

All the user information with comments are grouped by original tweets ID where we stored as CSV format. One CSV file contains all the comments regarding one tweet. Each entry contains all the information listed in Table 3. We selected 8 tweets with a total of 31,980 comments from Sina Weibo accounts belong to business owners, celebrities in entertainment and influencers for model training. The detailed tweets

Table 3: List of user related information from comment data

Field Name	Dataset Name	Meaning
uid	UID	Unique User ID for User Account in Weibo
screen_name	Username	Displayed User Nickname
followers_count	Follower	User's follower count
follow_count	Following	User's following count
status_count	Original_post	User's original composed tweets count
urank	User_rank	User's rated rank by user activity in Weibo
verified	Verified	Whether user is verified celebrity or business
description	Description	User's own description in headline
like_count	Like_count	Like count of this comment
floor_number	Floor_number	Location where the comment is at
text	Comment	Comment content

information and statistics are listed in Table 4.

Table 4: Statistics of troll detection tweets and comments crawled from Weibo

Num	Tweet ID	Tweet Detail	Number
1	44275283	LeEco CEO YT Jia declared bankrupt	812
2	44317480	Actress Yiyan Jiang volunteered teaching in rural	829
3	44564209	Yong actress Zi Yang suspected done plastic surgery	335
4	44718878	Reporting fraud in singer Hong Han Foundation receiving donation	1210
5	44651702	Singer Hong Han Foundation donation to Wu Han Coronavirus battle	3379
6	44650056	Criticism of multiple celebrities' donation to Coronavirus battle	814
7	43961306	Suspected breakup of Han Lu and Xiaotong Guan (Actor/ress)	8371
8	43961306	Han Lu and Xiaotong Guan (Actor/ress) showoff their same sweatshirt	16,230

3.3.1 Feature Extraction

We worked on the data listed in Table 3 with Python Pandas and Numpy in order to get most useful features from them. Meanwhile there are features that appears to be not useful at all so we drop it before feature extraction and passing them into model training. For example, most users are not certified as celebrities account or company account , therefore most of the data entries have verified field as "-1".

Moreover, in order to have the XGBoost model better understand data as feature. We need to engineer some of the features into better understanding features for the model. For example, for users with self description, I'm not going to analyze the content of the description but going to mark it as "1" for users with description and "0" vise versa.

Features such as follower count, following count and original composed tweets count have a higher importance in the analysis but we found out building model with quantitative numbers from the follower, following and original posts count will bias the model due to large difference between those numbers across all the entries. Normal Weibo (other social media platform) users usually only follow accounts in their favor. Troll users, on the other hand, follow a large number of accounts with their special goal but has fewer followers due to less value and attractiveness in troll users' original composed post. Therefore, we dropped follower and following count in the raw dataset and came up with the "Following/Follower" ratio as a feature. We also created a feature of "Original Post/Follower" to identify the fact that troll users could post a large number of posts in a short period of time without an equal amount of increase in their follower count. With this feature, we dropped the originally composed post feature from raw data.

During crawling for the training dataset from Weibo as well as manually identifying emotions and troll users from selected Weibo comments, we found out some users frequently commented on same tweet rather than replying others' comment under a tweet. We first selected users who has more than one comments under a tweet. And then get a count of comments each of them made. Then we can have a median number from all the count of comments. A frequent comment feature can be made by getting records for users who made comments more than this median number under a tweet.

3.3.2 Features

We used Python and Pandas package to process all the crawled comments and user information presented in Table 3, then to extract all the features from the comments file. At the very beginning, we wanted as many features as possible from the raw data. Then we performed multiple data manipulation with Pandas and NumPy in order to engineer the feature to reveal most of the facts.

With the original troll detection dataset together with sentiment score and emotion score calculated for each of the six emotions from a scale of 0 to 1, we have a total of 19 features to feed into the XGBoost model. One of the engineered features related to sentiment score is the `diffOriginalSenti`, which is the score for this comment subtracted by sentiment score of the original tweet. The goal of this project is to maximize the troll detection accuracy with minimum source data available. One of the important jobs is to perform feature analysis in order to rate importance among the features and drop features that could affect the quality of the model as we will illustrated in Section 4 Methodology. Table 4 lists the complete set of features we got by combining sentiment analysis result and user information data in Table 3 before the process of feature elimination at the feature ranking step.

3.3.3 Labels

We manually labeled 1, 2, 3, 6 from Table 4 by examining the each comment content and analyzing the user behavior of its Weibo account who commented. Troll users are labeled 1 and normal users are labeled 0. Combining with fake account data from [29]. We have about 3500 comments entries as initial training and testing data for the troll detection model. The manual labeling is tedious and need tremendous amount of job. We created a bot based on Selenium to help me open each user's Weibo page based on the UID we feed in from dataset to help me expedite this process.

Table 5: Features considered for troll detection model

	Feature	Description	Source
F0	`follower`	Follower count	Crawled Weibo dataset
F1	`following`	Following count	Crawled Weibo dataset
F2	`original_post`	Number of original tweets	Crawled Weibo dataset
F3	`urank`	Rank by user activity in Weibo	Crawled Weibo dataset
F4	`verified`	User certified or not	Crawled Weibo dataset
F5	`like_count`	Like count for a comment	Crawled Weibo dataset
F6	`floor_number`	Comment location	Crawled Weibo dataset
F7	`description`	Self description (1 or 0)	Engineered feature
F8	`freqComment`	Frequent comments	Engineered feature
F9	`ffRatio`	`following` divided by `follower`	Engineered feature
F10	`foRatio`	`original_post` divided by `follower`	Engineered feature
F11	`sentiment`	Comment sentiment score (0 to 1)	Engineered feature
F12	`diffOriginalSenti`	`sentiment` minus sentiment of original	Engineered feature
F13	`happy`	Happiness score (0 to 1)	Engineered feature
F14	`sad`	Sadness score (0 to 1)	Engineered feature
F15	`anger`	Anger score (0 to 1)	Engineered feature
F16	`disgust`	Disgust score (0 to 1)	Engineered feature
F17	`fear`	Fear score (0 to 1)	Engineered feature
F18	`surprise`	Surprise score (0 to 1)	Engineered feature

CHAPTER 4

Methodology

We experiment with several machine learning techniques. For the Chinese sentences segmentation part, we used Hidden Markov Model. For the sentiment analysis part, we used Naïve Bayes and Word vectoring to conduct calculating sentiment score of a word based on positiveness and negativeness of correlation to the both sentiment corpora. For the emotion score calculation part, we used HMM to map the comment content into a three dimensional vector using features introduced in Section 3.2.1. For the troll detection part, we used XGBoost model aggregating both sentiment and user information to identify troll users from a large amount of Weibo comment dataset. Next we will discuss the usage of each machine learning technique in the part it participate and introduce the evaluation method used in this project.

4.1 Hidden Markov Model

Hidden Markov Model are largely used in unsupervised machine learning field. HMM can be used to predict current state by state transition probabilities. The scenarios using HMM usually contain the cases that we cannot obtain state sequence but other observation related to hidden state is accessible. In this project, HMM are used in following two parts.

4.1.1 HMM used in Chinese sentences segmentation

As introduced in Section 3.1.1, there are four transition states (B, M, E, S) in the HMM model for sentence segmentation. The observation set is all the Chinese characters in the training dataset introduced in Section 3.1. We can then construct a 4×4 matrix as transition probability matrix to denote from all the transition probabilities from one state to another. The transition probability matrix is expressed as

$$\begin{bmatrix} \mathrm{B} \to \mathrm{B} & \mathrm{B} \to \mathrm{E} & \mathrm{B} \to \mathrm{M} & \mathrm{B} \to \mathrm{S} \\ \mathrm{E} \to \mathrm{B} & \mathrm{E} \to \mathrm{E} & \mathrm{E} \to \mathrm{M} & \mathrm{E} \to \mathrm{S} \\ \mathrm{M} \to \mathrm{B} & \mathrm{M} \to \mathrm{E} & \mathrm{M} \to \mathrm{M} & \mathrm{M} \to \mathrm{S} \\ \mathrm{S} \to \mathrm{B} & \mathrm{S} \to \mathrm{E} & \mathrm{S} \to \mathrm{M} & \mathrm{S} \to \mathrm{S} \end{bmatrix}.$$

The central theme of character based model in Chinese sentences segmentation is to tag different words in sentence by giving them different state. And by going through training data with my HMM model, we can build a transition probability matrix denoting transition probabilities between four states. Then we have emission probability which denote the probability of each observed Chinese word/character with certain state $P(\text{Observed}[i] \,|\, \text{State}[j])$. When finish training, we also will have a emission probability matrix denoting the probabilities of each Chinese character with certain state. By back tracking the emission probability and find the largest state among 4 states for each character, we can form the state transition by character.

The HMM observation sequence for the above segmented sentence is

BEBEBMEBEBEBES,

which has 4 states, also visualized in Table 6

Table 6: HMM observation sequence for Chinese sentence

马	克	硕	士	毕	业	于	加	州	理	工	学	院	呀
B	E	B	E	B	M	E	B	E	B	E	B	E	S

4.1.2 HMM used in Emotion classification

For each word in a tweet/comment, we can calculate a three dimensional vector based on its MI, CHI and TF-IDF with respect of training corpora introduced in Section 3.1. Then for a tweet or comment, we can get a mean value for each feature over all words for a certain emotion. Table 7 gives an example of a sentence with “A”, “B” and “C” three words and their features (mocked) with emotion respectively.

Table 7: Features of terms A, B and C in a sentence on all the emotions

Word	Emotion	MI	CHI	TD-IDF
A	Happiness	0.0012	0.0247	0.0009
	Anger	0.0012	0.0247	0.0070
	Sadness	0.0015	0.0100	0.0450
	Surprise	0.0080	−0.0050	0.0220
	Disgust	0.0020	0.0470	0.0117
	Fear	0.2200	0.0700	0.0009
B	Happiness	0.0167	0.0064	0.1045
	Anger	−0.0012	0.0247	0.0009
	Sadness	0.0200	−0.1416	0.0009
	Surprise	0.0012	−0.0247	−0.0009
	Disgust	−0.0012	0.0247	0.0009
	Fear	0.0012	0.0247	−0.0009
C	Happiness	0.0012	0.0247	0.0009
	Anger	0.0012	0.0247	0.0070
	Sadness	0.0015	0.0100	0.0450
	Surprise	0.3693	0.0820	−0.0119
	Disgust	0.0526	0.0247	0.0008
	Fear	0.0012	0.0247	0.0007

After calculating the feature vectors for the dataset for each emotion, we have a mean value of each feature for all tweets labeled as each emotion. This mean value is considered observation state. The transition probability

$$P(S_k = s_p \,|\, S_{k-1} = s_q) = \begin{cases} 1 & \text{if } p = q + 1 \\ 0 & \text{otherwise} \end{cases}$$

is considered as whether feature vectors of test tweets are close enough to those in training set when emotion changes.

The emission probability

$$P(y_k \,|\, S_k^{e_i}) = J(y_k \,|\, S_k^{e_i}) = \frac{M_{11}}{M_{11} + M_{10} + M_{01}}$$

can be calculated by Jaccard similarity [30] measuring correlation between feature vector y_k and state S_k, where M_{11} as total number of tweets containing feature vector

y_k and state S_k in emotion e_i; M_{10} as number of tweets containing only state S_k in emotion e_i; M_{01} as number of tweets containing only feature vector y_k in emotion e_i.

By compiling the transition and emission probability matrices, each of 6 emotions has its own HMM model. While testing with new tweet, we can select the largest probability calculated by each of the 6 HMM model and assign the emotion to this tweet.

4.2 Word2Vec in Sentiment Analysis

For the training set introduced Section 3.2, we constructed Word2Vec model based on 35,124 online shopping reviews with balanced positive and negative entries. And we inference 3,286,543 tweets from Weibo to construct positive and negative sentiment dataset to feed into Naïve Bayes model for sentiment score calculating.

4.3 Naïve Bayes

By using Naïve Bayes for sentiment score calculation, the goal is to sum up the frequency each training word's similarity to words in positive or negative dictionary. After finishing counting for the word frequency, we can then get a probability reflecting its positiveness or negativeness in the inference part by

$$P(c_1 \mid w_1, \ldots, w_n) = \frac{P(w_1, \ldots, w_n \mid c_1)P(c_1)}{P(w_1, \ldots, w_n \mid c_1)P(c_1) + P(w_1, \ldots, w_n \mid c_2)P(c_2)}.$$

The output from this Naïve Bayes model is the probability of positive test word. Then we can get sentiment score by using either the output probability or $1 - P(\text{positive})$ to get a sentiment score range from 0 to 1. Figure 7 below shows a sample sentiment score distribution of one of the training comments dataset.

4.4 XGBoost

XGBoost is a modified, advanced boosting technique to group multiple classifier into one better classifier [31]. The core of XGBoost is similar to decision tree. That is why we modified my raw data into entries that also could easily feed into decision

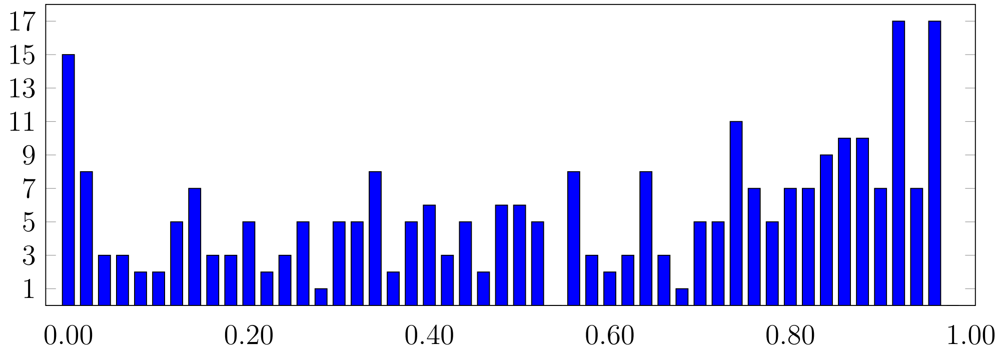


Figure 7: Sentiment score distribution for comments in Table 5, Row 1

tree. Also, use features with XGBoost package in Python, it is easier to analyze the importance among all the features and make decision to eliminate ineffective features. XGBoost can take all the weak classifier. With this advantage and sufficient numbers of feature, we can construct a relatively strong XGBoost classifier over other machine learning techniques.

Boosting is one of the ensemble methods used in supervised learning. Ensemble methods construct a set of classifiers from training data and combine the previous result from classifier set to help better predict class label. Each classifier stay independent. The basic idea for using ensemble method is by dividing dataset into multiple subsets then train each subsets with different classifier and combine the classifier. Boosting technique attentively change distribution of training data by focusing more on previously mis-classified records. At the beginning, all the classifier and record has equal weight, while weight is changing at the end of each boosting round.

The result of utilizing XGBoosting is promising, by dropping all the non-quantitative features and feed the XGBoost model with 9 features listed in Table 8. The model achieves 80% accuracy with a 5 fold training-test validation with each classifier accuracy of 53% to 72%. It is obvious to me that this some feature are irrelevant such as F8 and some are redundant such as F12 for the model training, so we rated the feature importance for the beginning model as below. Figure 8 shows the feature importance distribution and ranking, giving guidance to conduct feature reduction. Note that F12 is not shown in the Figure 8 since “diffOriginalSenti” contribute nothing to classification and received a 0 importance score from feature ranking.

Table 8: Troll detection statistics crawled from Weibo

	Feature	Description
F0	`follower`	Follower count
F1	`following`	Following count
F2	`original_post`	Number of original tweets
F3	`urank`	User activity rank in Weibo
F4	`verified`	User certified or not
F5	`like_count`	Like count for comment
F6	`floor_number`	Location of comment
F7	`description`	User’s self description (1 or 0)
F8	`freqComment`	User comments frequently or not
F9	`ffRatio`	`following` divided by `follower`
F10	`foRatio`	`original_post` divided by `follower`
F11	`sentiment`	Sentiment score of the comment (0 to 1)
F12	`diffOriginalSenti`	`sentiment` minus sentiment of original

Also, we experimented with threshold feature reduction with XGBoost, we had model drop the lowest importance rated feature one at a time and reevaluate the result for comparison. However, there were no substantial accuracy improvement found after this feature reduction as shown in Figure 9. We conducted future manually feature tuning in order to raise the accuracy of the model.

4.5 Support Vector Machine

The essential idea behind support vector machine is to find the boundary in order to maximize the margin between two differentiating classes. In [32], SVM is described as “a rare example of a methodology where geometric intuition, elegant mathematics,

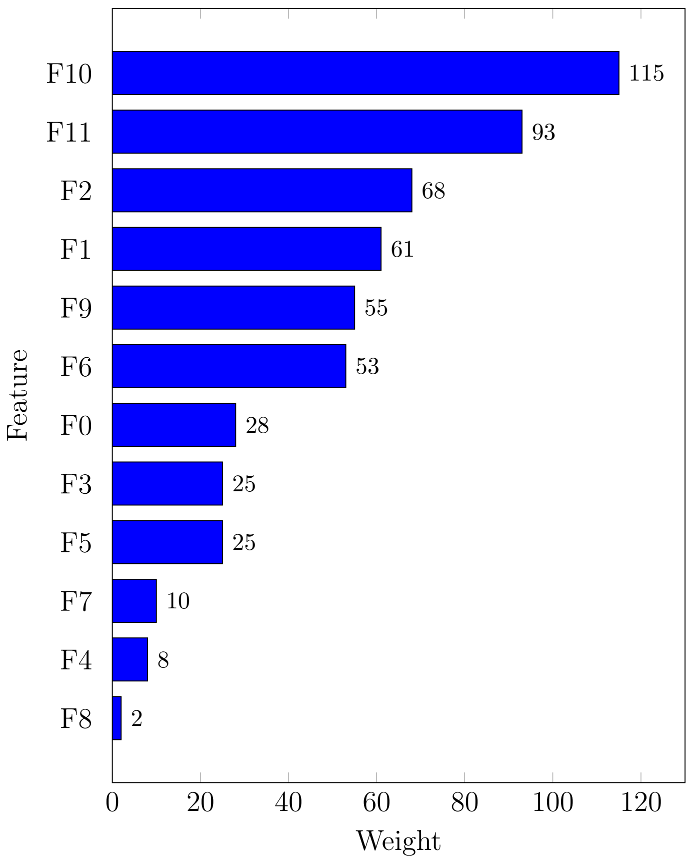


Figure 8: Initial XGBoost features ranking

theoretical guarantees, and practical algorithms meet."

In this project, SVM is utilized as a comparison to XGBoost method in this troll detection binary classification problem. We used SVM to separate hyperplane in training data between troll users and normal users. We also need to reduce the dimension by minimizing the features feeding into SVM to reduce the"curse of dimentionality". At last, we need to maximize the margin between the two hyperplane we created for troll users and normal users. We trained SVM for troll detection by mapping the training data into high dimensional feature spaces where we can find the separating hyperplane that may maximize the margin as we desired.

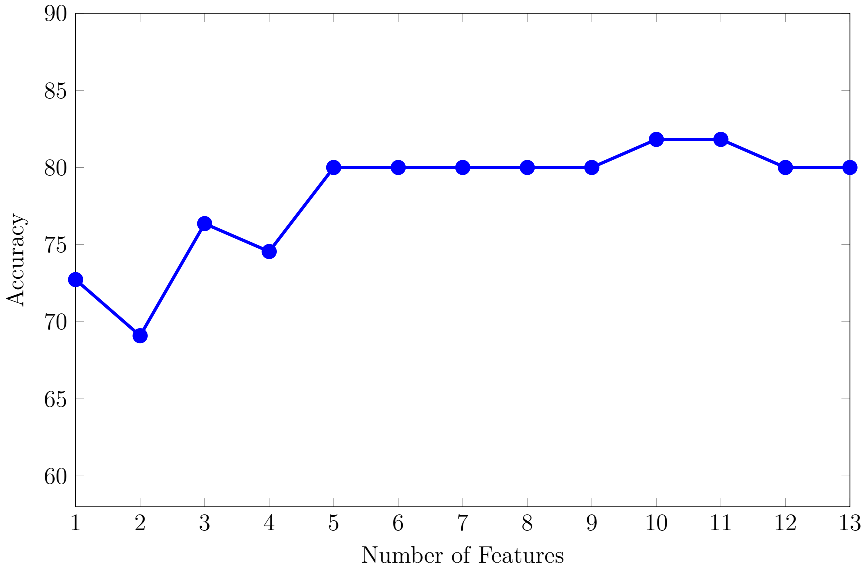


Figure 9: XGBoost model accuracy vs numbers of features

4.6 Cross Validation

Cross validation is a popular technique of augmentation with limited amount of data. By partitioning dataset into n equal size subsets then train on n-1 portion of data and test on 1 portion rest of data. For example, for 5-fold cross validation, we used in XGBoost model training. Dataset is divided into 5 portions and first 4 portion as training, last 1 portion as testing. Second round 1,3,4,5 as training, 2 as testing. Using cross validation minimize the bias of training dataset and maximize test independence based on original dataset.

4.7 Evaluation Metrics

We used accuracy as evaluating measurement for all classification methods used in this project. Giving an test with labeled dataset, accuracy is expressed as

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}},$$

where TP as true positive number, TN as true negative number, FP as false positive number, FN as false negative number. Accuracy is defined as ratio of number of correct classifications divided by total classifications.

We also implemented receiver operating characteristic curve (ROC) in analyzing the experiment results [33]. For the troll detection model as a binary classifier, ROC curve is constructed with true positive rate (TPR) and false positive rate (FPR) as threshold in the result data. The area under curve (AUC) ranging between 0 and 1 is the measurement of separation indicating whether false positive or false negative result exists. An AUC larger than 0.5 indicating better binary classification than flip a coin.

CHAPTER 5

Experiments and Results

5.1 Weibo Crawler

In order to obtain enough training and testing data raw from Internet. We composed a crawler for both post and comment text content as well as user information from Weibo platform. The crawler consists of three parts: posts crawler, comments crawler and user info crawler.

Posts crawler specifically crawl certain number of tweets under one certain Weibo account. Noted that similar to Twitter tweets, one can also retweet/repost others' post to their own Weibo account. My crawler specifically disregard these repost, only keeps account's original posts. The posts crawled from certain accounts are used to analyse certain user's sentiment across the account in help of analyzing its sentiment regarding of certain comment is typical or not.

Comments crawler accounts for the majority of the work since most of the comments contains repetitive replying information, username and hash-tagging. Removing them with Python Panda `Dataframe` function is a must before pipeline the comments into the word segmentation stage. Also, it is very common to see bi-lingual comments, most of which is Chinese-English mixed replies. Therefore, we incorporate a language detection module extended from Google language detection library using naïve Bayes filter to detect and translate the comments with English words embedded to all Chinese sentences.

The comment crawler works for Weibo mobile site [28] where the tweets and comments page are simplified for easier access of the content. The crawler makes HTTP request to the endpoint such as

```
https://m.weibo.cn/comments/hotflow?id=TWEETID&mid=TWEETID&max_id=
```

and gets JSON data from specific Weibo platform regarding a certain amount of comment data from specified tweet. Then we parse the comment data JSON packet to extract all the raw data including user information and comment content extracted by BeautifulSoup package. One obstacle we met was the change of the end point of Weibo mobile site in the beginning of 2020. We was forced to modify the crawler and find the property called max_ID for the current comment page in order to crawl for large amount of comments without getting blocked. After all the entries being collected, they are saved into a CSV file for each tweet. The feature extraction and model training are based on those CSV files as well in the later steps.

5.2 Chinese Word Segmentation

In this project, we implemented the Character-based Generative model[3] with Hidden Markov Model. For each line in training file. A python HMM model script was created to read each line of the training dataset introduced in Section 3.1 as a list of strings.

In the training method, first character of each segmented word is marked as beginning state; second to second last characters are marked as middle state; last character is marked as ending state. After calculate log value of emit probability, log value of transfer probability and log value of initial state then update the probability matrix for each state.

The segment method uses the matrices from training dataset to segment Weibo posts and comments corpora introduced in Section 3.2 and 3.3 line by line and place segment character space into sentence and output text-file as result. We modified this python segmentation script later to fit in the sentiment analysis script.

5.3 Word Vector Sentiment Score

In implementation of Word2Vec sentiment score calculation model. First step is to used the segmentation script in Section 5.2 to segment all the training corpora introduced in Section 3.1 and stored them as positive dictionary or negative dictionary according to the their source. Then we took reference of GenSim package in Python [34] to create the Word2Vec model based on both positive and negative dictionary. By using the Word2Vec model calculating the word similarity to words in both positive and negative dictionary we can assign a sentiment score from 0 to 1 representing the positiveness or negativeness of a word in the data that we want to inference where 0 represents definitive negative word and 1 represents definitive positive word.

We used Weibo corpora as inference dataset and SnowNLP [27] original sentiment corpora from online shopping review as training dataset to build a new training set solely based on Weibo corpora for better accuracy on more casual wording on Weibo platform for future inference Weibo comments data.

5.4 Troll Detection by XGBoost Classification

We implemented the troll detection model using Extreme Gradient Boosting (XGBoost) using Python with Jupyter Notebook environment. First we need to define the proper training data source range with numerous of input features. In the discuss in Section 4, we figured out simply dropping features by its importance rating did not make substantial improvement on the accuracy of the model. Therefore, we first dropped several redundant features including `following`, `follower`, and `original_post` since we already put in `ffRatio` and `foRatio` as two engineered feature. The accuracy got up to 81.82% accuracy, similar to feature reduction we conducted previously. Then we conducted another round of feature analysis, found out the `diffOriginalSentiment` feature has no contribution to the model at all. So we droped that feature as well. The rest of features and their importance rank are shown in Figure 11.

5.5 Troll Detection based on Support Vector Machine

We utilized Python Sci-kit learn package as our SVM classification tool. We used the 3 best features as we used in XGBoost model into C-Support Vector Classification method in SVM to created 3 sets of support vectors. The classification used a 5-fold cross validation and 5 splits with 3 random starts for the ROC-AUC calculation.

5.6 Chrome Extension for Troll Detection Model

The usecase for the Weibo troll detection extension is simple. Troll detection extension runs in the background as Chrome web browser. When user browses specific Weibo tweets with comments attached to them, the URL for specific Weibo tweets page will match the manifest file in Chrome extension so the plug-in starts injecting Javascript which intercept all HTTP requests this page received with comments and users data. By sending out this data back from Chrome extension to the backend server, the backend return back the troll detection result and make the corresponding comment on the page blurred in order to hide away from users.

Since most of the troll detection mechanism is composed in Python. We made the front-end Chrome extension using HTML and JavaScript to pass the HTTP request packet with Weibo comment information to backend built by Flask framework with troll detection model logic for the real-time troll detection on Sina Weibo mobile site. The workflow for the plug-in is:

- Inject Javascript inside plug-in to Weibo tweet page DOM in order to get HTTP request which Weibo mobile site gets comments data from and send it to background javascript inside plug-in;
- Background javascript from plug-in send a cross domain request to backend server with the intercepted request from Weibo site;
- Server side runs the crawling script using given HTTP request URL and returns

JSON packet containing comments text and user information;

- On the server-side, sort out essential user information and comment text from returned JSON packet from crwaler;
- On the server-side, run the sentiment analysis script against comment text and acquire text sentiment score;
- On the server-side, aggregate sentiment score and other user information into the troll detection model and return troll detection result to client side plug-in;
- Plug-in modify the CSS style sheet for certain comments classified as troll comments by blurring those comments.

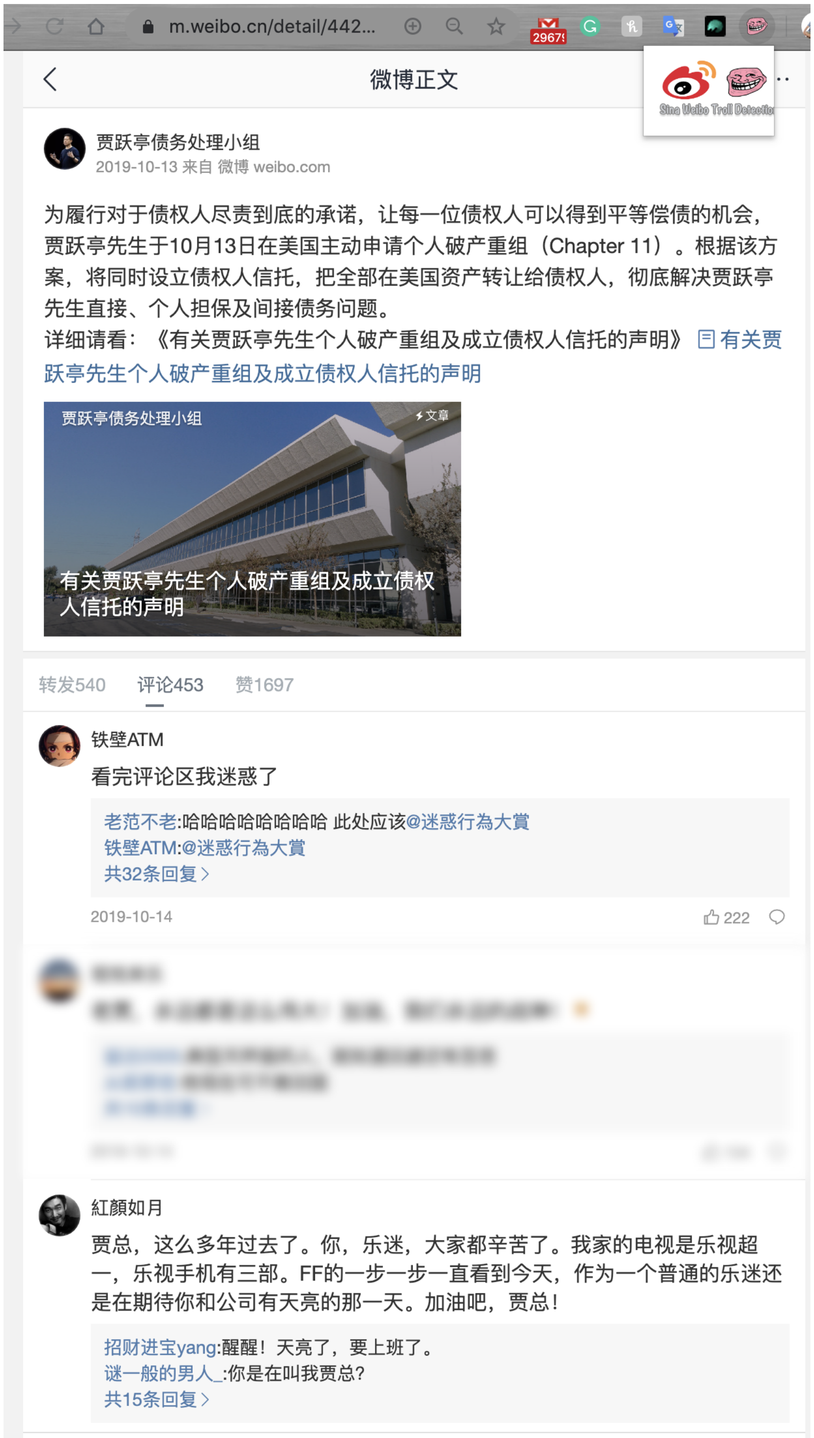
A screenshot for the plug-in in working condition with identified troll comments blurred is shown in Figure 10.

5.7 Results and Discussions

By utilizing XGBoost model as classification method, we achieved the troll detection accuracy of 83.64% by only using 3 features (F9, F10, F11 from Figure 11 and Table 8), which correspond to `Following/Follower Ratio`, `Original Post/Follower Ratio`, and `Sentiment Score` contribute the most to the classification. The final chosen features are marked yellow in the feature importance distribution.

By utilizing SVM as classification method, we achieved 87.27% accuracy using `Following/Follower Ratio`, `Original Post/Follower Ratio`, and `Sentiment Score` as features. The SVM model achieved mean AUC-ROC of 0.67409 in a 3 runs 5 splits ROC test.

Another experiment we conducted with XGBoost model is by replacing the Weibo corpora with SnowNLP sentiment dataset for the sentiment score calculation. By using this training dataset for sentiment analysis score feature and with addition of F3 (user rank) and F6 (floor number) from Figure 11 and Table 8, the final accuracy got



微博正文

贾跃亭债务处理小组
2019-10-13 来自 微博 weibo.com

为履行对于债权人尽责到底的承诺，让每一位债权人可以得到平等偿债的机会，贾跃亭先生于10月13日在美国主动申请个人破产重组（Chapter 11）。根据该方案，将同时设立债权人信托，把全部在美国资产转让给债权人，彻底解决贾跃亭先生直接、个人担保及间接债务问题。
详细请看：《有关贾跃亭先生个人破产重组及成立债权人信托的声明》 有关贾跃亭先生个人破产重组及成立债权人信托的声明

贾跃亭债务处理小组 文章
有关贾跃亭先生个人破产重组及成立债权人信托的声明

转发540 评论453 赞1697

铁壁ATM
看完评论区我迷惑了
老范不老:哈哈哈哈哈哈哈 此处应该@迷惑行為大賞
铁壁ATM:@迷惑行為大賞
共32条回复 >
2019-10-14 222

[redacted]

紅顏如月
贾总，这么多年过去了。你，乐迷，大家都辛苦了。我家的电视是乐视超一，乐视手机有三部。FF的一步一步一直看到今天，作为一个普通的乐迷还是在期待你和公司有天亮的那一天。加油吧，贾总！
招财进宝yang:醒醒！天亮了，要上班了。
谜一般的男人_:你是在叫我贾总?
共15条回复 >

Figure 10: Screenshot of working Chrome extension employing troll detection model

improved to 89% with a mean AUC-ROC of 0.7498 shown in Figure 13. The accuracy comparison for all the experiments is shown in Figure 12.

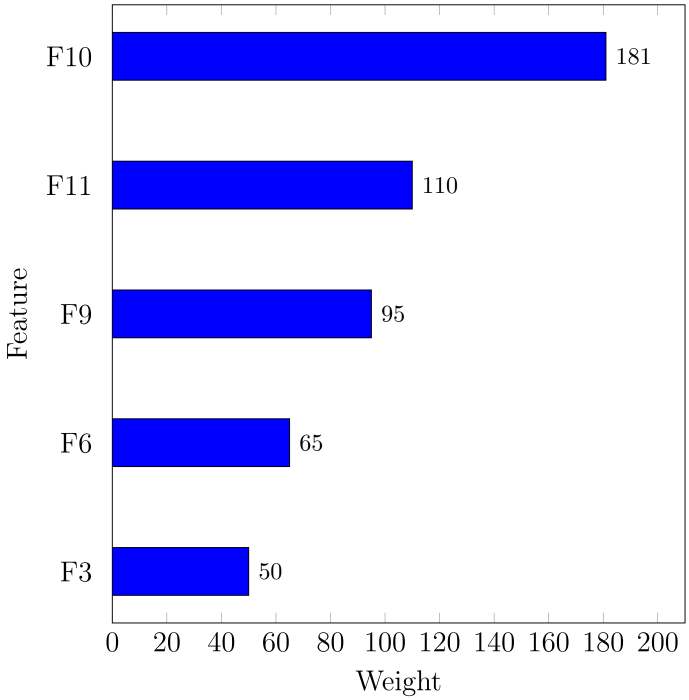


Figure 11: Final XGBoost feature ranking

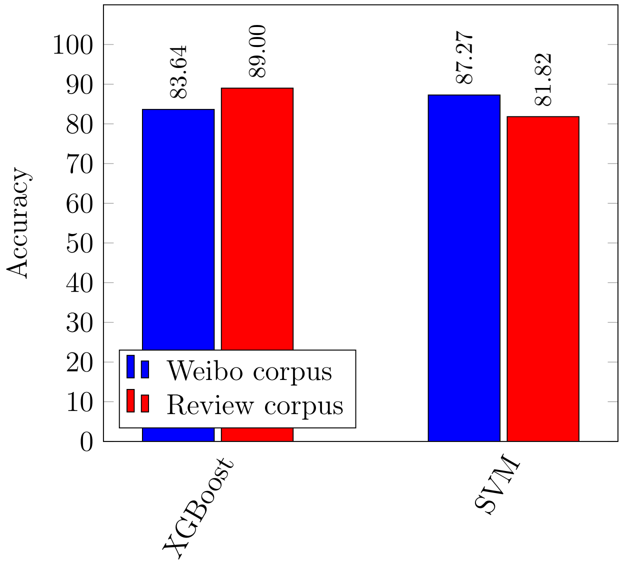


Figure 12: Accuracy comparison of XGBoost and SVM

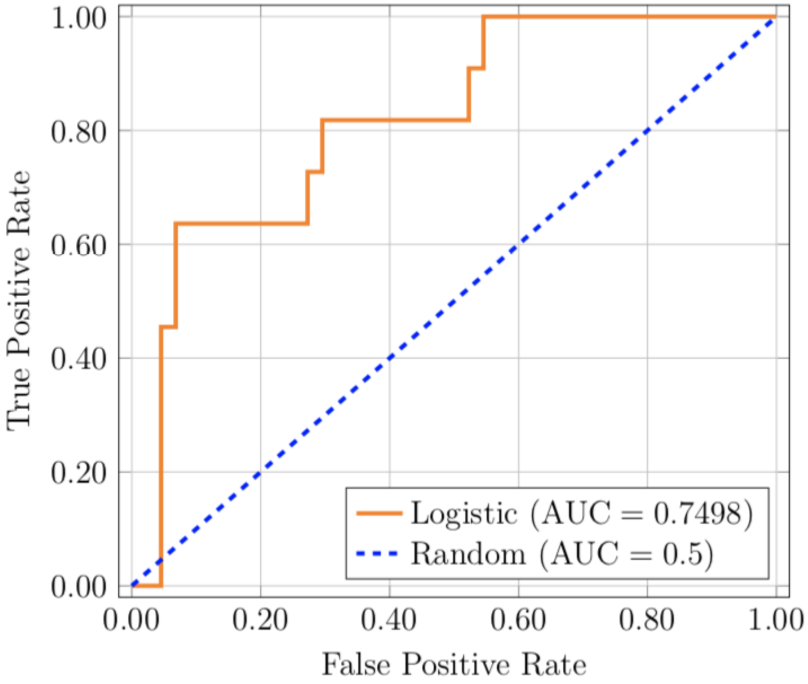


Figure 13: Area under ROC curve for XGBoost

In conclusion, the experiment result showing above by combining both sentiment analysis result and user relationship data for classification of troll activity achieved higher accuracy of 89% than accuracy of traditional troll detection method 78% only by analyzing content sentiment suggested in Section 2.3.1. Also, the proposed troll detection method uses the minimum number of features (5 total feeding features) and raw data needed for troll detection than any other referenced method in this research, which makes room more efficiency less overhead for the real-time troll detection application.

CHAPTER 6

Conclusion and Future Work

6.1 Conclusion

The widespread usage of social media makes our sight broader as well as makes the information transfer faster than ever before. In the mean time, such development boom from social media inevitably creates the troll activities all over the Internet. Trolls are driven by profits, publishing deceptive and meaningless information. Information published by trolls often time misguide general public. In this project, we utilized different machine learning techniques to adequately analyzing comment post content and user information from Sina Weibo platform. By conducting sentiment analysis and troll user data aggregation, we can quickly identify troll comments on Sina Weibo. As result, we can achieve 89% accuracy for troll detection on a small number of sample data we crawled from Weibo.

6.2 Future Work

There are still a long way to go in fighting with troll users existing on Internet and further prevent the negative influence by Water Army. We need to further utilize the emotion categorization data we get by HMM model and find useful feature from emotion from comments. We can still acquire more user data such as Weibo credit from Sina Weibo desktop site for each user as a potential more effective and accurate feature in troll detection. More precious time of publishing comment can be also gathered to analyse user activity for posting comments under a tweet to feed in to troll detection model. A categorical model can be proposed to sub-divide Weibo into different category for training and testing to rule out potential fans group for posts belong to entertainment celebrities that can bias the model. Lastly, we can also try to use more nuance machine learning technologies such as recurrent neural network, long short term memory as comparison for the current classification methods.

The Chrome extension should support Weibo desktop site in the future for better user experience and should allow user to list out parameter they want to rule out for the troll detection setting. Also, user should be able to reset the page content to see all the comments. Since Webkit is widely used in multiple popular Internet browsers as browser core, we might be able to derive the troll detection plug-in for Chrome to be compatible to other Webkit browsers such as Firefox. We believe the use of machine learning can finally conquer the challenges posing by troll activities in the future.

LIST OF REFERENCES

[1] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," http://arxiv.org/abs/1301.3781, 2013.

[2] X. Zhao and Y. Ohsawa, "Sentiment analysis on the online reviews based on hidden Markov model," *Journal of Advances in Information Technology*, vol. 9, pp. 33–38, May 2018.

[3] K. Wang, C. Zong, and K.-Y. Su, "Which is more suitable for chinese word segmentation, the generative model or the discriminative one?" *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*, pp. 827–834, 2009.

[4] M. Chen, B. Chang, and W. Pei, "A joint model for unsupervised chinese word segmentation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 854–863, 2014.

[5] H. Peng, E. Cambria, and A. Hussain, "A review of sentiment analysis research in chinese language," *Cognitive Computation*, vol. 8, p. 423–435, May 2017.

[6] S. Feng and E. Durdyev, "Fine-grained sentiment analysis of restaurant customer reviews in Chinese language," http://cs229.stanford.edu/proj2018/report/195.pdf, 2018.

[7] Indiana University, "What is a troll?" https://kb.iu.edu/d/afhc, 2018.

[8] K. Dilanian, "Russian trolls who interfered in 2016 U.S. election also made ad money," https://kb.iu.edu/d/afhc, 2019.

[9] Sina-Entertainment, "Weibo of Zi Yang dominated by troll," http://ent.sina.com.cn/s/m/2016-11-03/doc-ifxxneua4008428.shtml, 2016.

[10] J. Zhao and H. Wang, "Detection of fake reviews based on emotional orientation and logistic regression," *Journal of CAAI Transactions on Intelligent Systems*, vol. 13, pp. 336–342, June 2016.

[11] Y. Huang, M. Zhang, Y. Yang, S. Gan, and Y. Zhang, "The Weibo spammers' identification and detection based on Bayesian-algorithm," in *2016 2nd Workshop on Advanced Research and Technology in Industry Applications*, ser. WARTIA-16, 01 2016.

[12] C. W. Seah, H. L. Chieu, K. M. A. Chai, L.-N. Teow, and L. W. Yeong, "Troll detection by domain-adapting sentiment analysis," in *18th International Conference on Information Fusion*, ser. Fusion 2015, 2015, pp. 792–799.

[13] Y. Liu, X. Wang, and W. Long, "Detection of false Weibo repost based on XGBoost," in *IEEE/WIC/ACM International Conference on Web Intelligence*, ser. WI '19. ACM, 2019, pp. 97–105.

[14] M. Stamp, "A revealing introduction to hidden Markov model," https://www.cs.sjsu.edu/~stamp/RUA/HMM.pdf, 2004.

[15] R. Vobbilisetty, F. D. Troia, R. M. Low, C. A. Visaggio, and M. Stamp, "Classic cryptanalysis using hidden Markov models," *Cryptologia*, vol. 41, no. 1, pp. 1–28, 2017.

[16] M. Stamp and S. Venkatachalam, "Detecting undetectable metamorphic viruses," in *Proceedings of 2011 International Conference on Security & Management*, 07 2011, pp. 340–345.

[17] D. Dhanasekar, F. Di Troia, K. Potika, and M. Stamp, "Detecting encrypted and polymorphic malware using hidden Markov models," in *Guide to Vulnerability Analysis for Computer Networks and Systems: An Artificial Intelligence Approach*, S. Parkinson, A. Crampton, and R. Hill, Eds. Springer, 2018, pp. 281–299.

[18] R. Calvo and S. Kim, "Emotions in text: Dimensional and categorical models," *Computational Intelligence*, vol. early view, 01 2012.

[19] N.-R. Kim, K. Kim, and J.-H. Lee, "Sentiment analysis in microblogs using hmms with syntactic and sentimental information," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 17, pp. 329–336, 12 2017.

[20] L. Liu, D. Luo, M. L. Liu, J. Zhong, Y. Wei, and L. Sun, "A self-adaptive hidden Markov model for emotion classification in Chinese microblogs," *Mathematical Problems in Engineering*, vol. 2015, no. 987189, 2015.

[21] 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.* Association for Computational Linguistics, July 2003, pp. 184–187.

[22] M. Asahara, K. Fukuoka, A. Azuma, C.-L. Goh, Y. Watanabe, Y. Matsumoto, and T. Tsuzuki, "Combination of machine learning methods for optimum Chinese word segmentation," in *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, 2005.

[23] K. Wang, C. Zong, and K.-Y. Su, "Which is more suitable for Chinese word segmentation, the generative model or the discriminative one?" in *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*. City University of Hong Kong, 2009, pp. 827–834.

[24] M. Chen, B. Chang, and W. Pei, "A joint model for unsupervised Chinese word segmentation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP 2014. Association for Computational Linguistics, 2014, pp. 854–863.

[25] Google-Developers, "Embeddings can produce remarkable analogies," https://developers.google.com/machine-learning/crash-course/embeddings/translating-to-a-lower-dimensional-space, 2020.

[26] T. Emerson, "The second international Chinese word segmentation bakeoff," https://pdfs.semanticscholar.org/65e9/0d9f6754d32db464f635e7fdec672fad9ccf.pdf, 2005.

[27] R. Wang, "SnowNLP Python package," https://github.com/isnowfy/snownlp, 2018.

[28] Sina-Weibo, "Weibo mobile site," https://m.weibo.cn/, 2009.

[29] L. Liu, Y. Lu, Y. Luo, R. Zhang, L. Itti, and J. Lu, "Detecting "smart" spammers on social network: A topic model approach," in *Proceedings of the NAACL Student Research Workshop*. San Diego, California: Association for Computational Linguistics, 2016, pp. 45–50.

[30] P. Jaccard, "Nouvelles recherches sur la distribution florale," *Bulletin de la Societe Vaudoise des Sciences Naturelles*, vol. 44, pp. 223–70, 01 1908.

[31] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," http://arxiv.org/abs/1603.02754.

[32] K. P. Bennett and C. Campbell, "Support vector machines: hype or hallelujah?" *SIGKDD Explorations*, vol. 2, pp. 1–13, 2000.

[33] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145 – 1159, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320396001422

[34] R. Řehůřek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 2010, pp. 45–50.