Sentiment Analysis of Chinese Microblog Posts on Attitude Towards Vaccination Since the Distribution of COVID-19 Vaccines in China

Zhaoqing Bai (baizq871), Yifei He (yifeihe), Sylvia Yuan (luyuan)

Abstract

By comparing performance of different sentiment analysis methods such as Naive Bayes, n-gram, k-means clustering with word embeddings, logistic regression, SVM, and long short-term memory (LSTM), we found two models with comparable performance on the validation dataset of 1000 posts, a LSTM model with bigram tokenization and a SVM model with text2vec. We then used the two models to predict the percentage of positive posts and analyzed the trend of attitude towards vaccination from December 31, 2020 to mid-April 2021 in China. We found that during this time frame, the percentage of positive posts remains predominant, but decreases over time likely due to experience of side effects of the vaccines.

1 Introduction

In light of the current situation with COVID-19, attitudes towards vaccination on social media platforms are important in assessing whether a population would be willing to receive vaccination. With large amounts of information online surrounding COVID-19 and the recent vaccine distribution, we hope to find a way to analyze posts about vaccination on social media platforms. To understand the Chinese population's attitude towards COVID-19 vaccination, we conduct sentiment analysis on Chinese social media posts, specifically on the Weibo (microblog) platform. This project applies different existing natural language processing algorithms to Weibo posts collected online.

2 Related Works

Existing research had tried to analyze attitudes towards vaccination on English social media platforms. For example, many assessed various vaccination sentiment trends (for vaccinations such as measles vaccination, human papillomavirus vaccination, or COVID-19 vaccination) with Twitter data [1][2][3]. Some previous studies have also been done on sentiment analysis in the Chinese language [4]. [4] introduces a variety of studies on Chinese sentiment analysis and their respective performance on different datasets. For example, Zhai et al. explores different features in Chinese sentiment analysis, and found that character bigram features outperform other n-gram features [5]. Also according to [4], the best performing algorithm is the weakly supervised method by Zhang and He, which applies a self-training cycle after obtaining pseudo-labelled training set from applying classification methods on two partition splits of the test dataset and training initial classifier on the pseudo-labelled training set [6].

3 Dataset and Features

We collected a dataset with weibo-search[1] to get over 60000 posts from January to April 2021[7]. Upon examining the datasets, we found that there are a significant amount of media posts, and most of them contain the characters "【】". Hence, we discarded the posts containing "【", hoping this will reduce the percentage of media posts in the datasets. Then we concatenated all of the posts, randomly selected the training set (around 2100 posts), validation set (around 850 posts), and test set (around 850 posts), and hand-labelled the posts. For error analysis, we also labelled whether each post is an opinion post or news post. We separated the remaining dataset according to the months in which they are posted, for later trend analysis purposes.

[1] Available at: https://github.com/dataabc/weibo-search

As preprocessing, we decided to try four tokenization methods of the posts, tokenization by the python jieba library, unigram, bigram, and trigram. Jieba is a Chinese text segmentation tool that separates strings into meaningful words[2][8]. We used and modified a list of stopwords from Stopwords Chinese (ZH), a collection of Chinese stopwords[3][9] to remove stop words from posts.

4 Methods

4.1 Naive Bayes

We chose Naive Bayes with Laplace smoothing as the baseline method. The prediction probability functions are as follows [10]:

$$\phi_{k|y=1} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i} 1\{x_j^{(i)}=k \wedge y^{(i)}=1\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)}=1\} n_i + |V|} \quad (1)$$

$$\phi_{k|y=0} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i} 1\{x_j^{(i)}=k \wedge y^{(i)}=0\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)}=0\} n_i + |V|} \quad (2)$$

where $\{(x^{(i)}, y^{(i)})\}$is the training set with m training examples, each training sample is of length n_i, and $|V|$ is the size of the vocabulary.

To reduce overfitting, we also attempted to combine classification results of bigram and trigram tokenization with linear interpolation on probability found with bigram and trigram [11]:

$$log(P(Y = 1|X)) = \lambda log(P_{bigram}(Y = 1|X)) + (1 - \lambda) log(P_{trigram}(Y = 1|X)) \quad (3)$$

4.2 Support Vector Machine (SVM)

To compare with our baseline method, we first attempted to classify using the support vector machine method as implemented in the open-source machine learn package scikit-learn [12]. A support vector machine constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. Given training vectors $x_i \in R^d$, i=1,…, n, in two classes, and a vector $y \in \{-1, 1\}^n$, our goal is to find $\theta \in R^d$ and $b \in R$ such that the prediction given by sign($\theta^T \phi(x_i) + b$) is correct for most samples. In particular, SVM solves the following primal problem [13]:

$$\min_{\theta,b,\zeta} \frac{1}{2}\theta^T\theta + C\sum_{i=1}^{n}\zeta_i \text{ , subject to } y_i(\theta^T\phi(x_i) + b) \geq 1 - \zeta_i, \quad (4)$$

where we allow some samples to be at a distance $\zeta_i \geq 0,\ i = 1,\ldots, n$ from their correct margin boundary.

4.3 Logistic Regression

We used the logistic regression approach implemented with the scikit-learn library [12] as our classifier. Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. The logistic regression minimizes the the following regularized cost function [14]:

---

[2] Available at: https://github.com/fxsjy/jieba

[3] Available at: https://github.com/stopwords-iso/stopwords-zh.git

$$\min_{\theta,c} \ \frac{1}{2}\theta^T\theta + C\sum_{i=1}^{n} \log(\exp(-y_i(X_i^T\theta + c)) + 1) \ , \tag{5}$$

In order to use logistic regression, we have to convert the input text to a vector using word embedding. To that end, we have attempted two methods: raw count matrix and text2vec[4].

4.4 Bigram Model

We attempted using the bigram model to classify [11]. Denote the text as x (where x_i is the i-th word (jieba or unigram), and b and e are start and end token respectively) and its sentiment as y. We first calculate $p(x_i \mid x_{i-1}, y)$ for the training set with add-1 (laplace) estimate. Then we can calculate the possibility of the new text x having a sentiment y by

$$p(y \mid x) \propto p(e \mid x_n, y) \cdot p(x_n \mid x_{n-1}, y) \cdot \ldots \cdot p(x_2 \mid x_1, y) \cdot p(x_1 \mid b, y) \cdot p\,(y) \tag{6}$$

4.5 Long Short-Term Memory

We also attempted a long short-term memory (LSTM) recurrent neural network to improve accuracy of classification. LSTM is a recurrent neural network architecture that can learn long term dependence among features and the objective [13]. The recurrent network is formed by a series of LSTM units. In each LSTM unit, calculation is performed as follows:

$$\bar{c}^{<t>} = tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c) \tag{7}$$

$$\Gamma_u = \sigma(W_u[W_c[a^{<t-1>}, x^{<t>}] + b_u) \tag{8}$$

$$\Gamma_f = \sigma(W_f[W_f[a^{<t-1>}, x^{<t>}] + b_f) \tag{9}$$

$$\Gamma_o = \sigma(W_o[W_o[a^{<t-1>}, x^{<t>}] + b_o) \tag{10}$$

$$c^{<t>} = \Gamma_u * \bar{c}^{<t>} + \Gamma_f * c^{<t-1>} \tag{11}$$

$$a^{<t>} = \Gamma_o * c^{<t>} \tag{12}$$

where $a^{<t-1>}$, $a^{<t>}$ represent hidden states, $x^{<t>}$ is input, c is cell state, W is weight, b is bias, and Γ_u, Γ_u, Γ_u are the update gate, forget gate, and output gate respectively [15]. Then, backpropagation is performed through time to update the parameters [15].

Language is a form of sequence data, hence LSTM can be applied to conduct sentiment classification. We used the python Tensorflow library [16] to implement the model. The model architecture is adapted and modified from Chinese sentiment analysis[5] with Gated Recurrent Units (GRU). Binary cross-entropy loss function is used [10]:

$$l(\theta) = \sum_{i=1}^{n} y^{(i)} log\, h(x^{(i)}) + (1 - y^{(i)}) log\,(1 - h(x^{(i)}))\tag{13}$$

5 Experiments / Results / Discussion

We were able to find the following results with different classifiers on the training set preprocessed with different tokenization methods:

Table 1: Performance on Training and Validation Set

Classifier	Tokenization	Training Acc	Dev Acc	Dev Precision	Dev Recall	Dev F1
Naive Bayes	**Unigram**	0.7712	0.6935	0.7804	0.7250	0.7517

[4] Available at: https://github.com/shibing624/text2vec
[5] Available at: https://github.com/Tony607/Chinese_sentiment_analysis.git

	Bigram	0.9541	0.7226	0.7381	0.8780	0.8020
	Trigram	0.9732	0.7156	0.7488	0.8361	0.7900
	Jieba	0.8769	0.7168	0.7602	0.8142	0.7863
SVM	**Unigram**	0.9695	0.6422	0.6424	0.9945	0.7806
	Bigram	1.0000	0.6434	0.6441	0.9891	0.7802
	Trigram	0.9995	0.6399	0.6399	1.0000	0.7804
	Jieba	1.0000	0.6399	0.6415	0.9909	0.7788
	text2vec	0.8339	0.7599	0.7727	0.8852	0.8251
Logistic Regression	**Unigram**	0.9756	0.7145	0.7235	0.8962	0.8007
	Bigram	0.9981	0.7319	0.7250	0.9362	0.8172
	Trigram	1.0000	0.6993	0.6974	0.9362	0.7994
	Jieba	0.9930	0.7331	0.7286	0.9290	0.8167
	text2vec	0.7884	0.6853	0.6755	0.9781	0.7991
Bigram Model	**Unigram**	0.8827	0.6608	0.6554	0.9909	0.7890
	Jieba	0.9615	0.6772	0.6696	0.9781	0.7950
LSTM	**Unigram**	0.9466	0.7389	0.7543	0.8780	0.8114
	Bigram	0.9862	0.7541	0.7485	0.9271	0.8283
	Trigram	0.9970	0.7226	0.7244	0.9144	0.8084
	Jieba	0.9852	0.7436	0.7423	0.9180	0.8208

From the performance of Naive Bayes, we deduced that there may be overfitting in Naive Bayes with trigram. By applying linear interpolation and looping through different values of λ, we found when $\lambda = 0.8$, we were able to achieve an accuracy of 0.7238 on the validation set. However, the run time is heavily affected by needing to predict with two different tokenization methods, and improvement in performance is limited. Hence, we choose not to apply linear interpolation in trend analysis, and established a baseline accuracy of 0.7226 with Naive Bayes method with bigram tokenization.

For logistic regression, we attempted different regularization strength and used the C that yields the best result on the dev set for each tokenization method (e.g., $C = 2$ for jieba and $C = 1$for bigram).

We experimented with LSTM recurrent neural networks. We chose to have an embedding layer with word embedding dimension of 256, and one LSTM layer with 64 units, followed by a dense layer with sigmoid activation function. Adam optimizer was used with learning rate 0.0005, beta 1 value 0.9, and beta 2 value 0.999. Batch size for training was 16. The number of layers and hyperparameters were chosen according to validation accuracy after repeated experiments with the training set tokenized with the jieba tokenization method. We were able to achieve a validation accuracy of 0.7541 and an F1 score of 0.8283 with the LSTM model with bigram tokenization.

From predictions by the LSTM model with bigram tokenization, we extracted the 171 false positive and 40 false negative posts for error analysis. One excerpt from a false negative post is “做事情不要犹豫，要敢做，敢去做，就像疫苗，一犹豫没了”, the post repeatedly uses the word “犹豫”, which means hesitation, but the overall meaning is not to hesitate when deciding whether to take the vaccine. Such double negatives seem to be one of the causes for errors.

According to performance on the validation set, we selected the LSTM model with bigram tokenization and the SVM model with text2vec to conduct trend analysis on the remaining dataset. Running the two models on the test set, we get the following results in Table 2. The two models have comparable and reasonable performance on the test set.

Table 2: Performance on Test Set

	Accuracy	**Precision**	**Recall**	**F1**

LSTM (bigram)	0.8164	0.8452	0.9045	0.8738
SVM (text2vec)	0.8185	0.8576	0.8896	0.8733

6 Application: Trend Analysis

Observing the collected posts, we discovered a heated discussion surrounding the promised free distribution of COVID vaccination on December 31, 2020. Hence, from December 31, 2020, we separated the remaining dataset based on month. Using the two models selected based on performance on the validation set, we predicted the percentage of positive posts by month (the latest posts are from April 15). Table 3 shows the result. We discovered that though with differences, the two models show a similar trend. Both predict a high percentage on December 31 2020, maintain a largely stable trend through January, February, and March, then show a drop in April. We plotted the sentiment trend against the COVID-19 vaccine doses received in China (Figure 1). As there is a sharp increase in doses administered, we observe an abrupt drop in percentage of positive posts. Observing the content of the posts in April, we believe this drop may not be due to drop in confidence towards the vaccine, but due to experience of side effects of the increasing portion of population that has received the first or both doses.

Table 3: Percentage of Positive Posts by Month

		December 31	**January**	**February**	**March**	**April**
Total Number of Posts		3588	11647	5136	5388	4205
Percentage of Positive Posts	**LSTM (bigram)**	0.9512	0.7921	0.8033	0.7564	0.6326
	SVM (text2vec)	0.9445	0.7696	0.7841	0.7793	0.5705

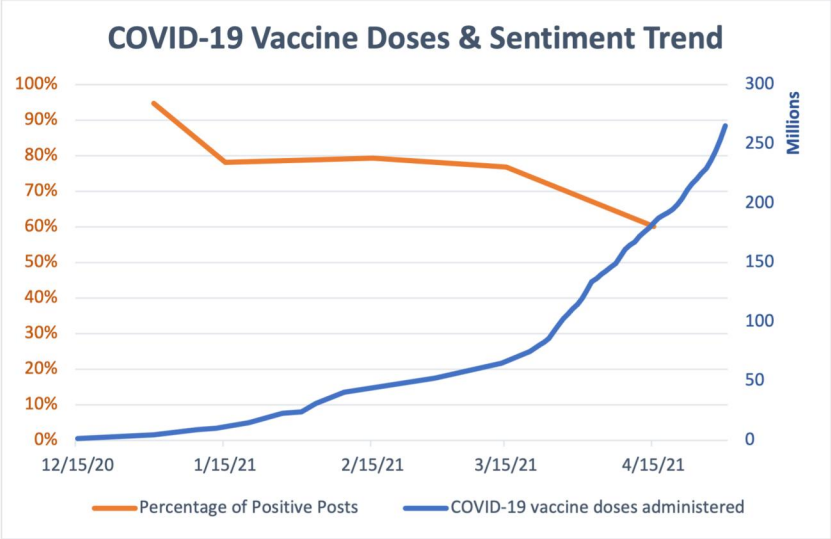

Figure 1: Predicted Sentiment Trend and Vaccine Doses Administered in China[17]

7 Conclusion / Future Works

After exploring different methods of sentiment analysis, we found two models, a LSTM model with bigram tokenization, and a SVM model with text2vec, that has comparable performance on the validation set. Then, we conducted predictions by these two models on the remaining dataset and analyzed the sentiment trend by month.

There is still space for improvement in our models. Part of existing error may also be due to vagueness in attitude towards vaccines in many posts and possible human errors when labelling. Adding a neutral class to the current binary classification method may help reduce the error. In addition, we can experiment with other methods, such as training with Bidirectional Encoder Representations from Transformers (BERT) [18], a model that can reduce error due to double negatives.

Contributions

Zhaoqing Bai:

- Did data cleaning to part of the training set
- Divided part of the training set into opinion/non-opinion categories
- Labelled 2200+ posts for training set and test set
- Contributed to part of the scripts that run logistic regression classifier and SVM classifier
- Contributed to the report

Yifei He:

- Retrieved part of dataset with weibo-search source code online
- Labelled around 600 posts for training set and test set
- Contributed to part of the code in preprocessing.py and util.py
- Analyzed different sub-portions (opinions only vs. non-opinions only) of the dataset as well as precision and recall
- Coded the bigram model
- Coded the logistic regression classifier and SVM classifier
- Coded k-means clustering
- Contributed to the report

Sylvia Yuan:

- Retrieved part of dataset with weibo-search source code online
- Labelled 1300+ posts from validation set and part of test set
- Contributed to code in preprocessing.py and util.py
- Coded the Naive Bayes classifier and linear interpolation
- Coded the LSTM model
- Contributed to the report

Reference

[1] Raghupathi V, Ren J, Raghupathi W. Studying Public Perception about Vaccination: A Sentiment Analysis of Tweets. *International Journal of Environmental Research and Public Health*. 2020; 17(10):3464. https://doi.org/10.3390/ijerph17103464

[2] Erika Bonnevie, Allison Gallegos-Jeffrey, Jaclyn Goldbarg, Brian Byrd & Joseph Smyser (2021) Quantifying the rise of vaccine opposition on Twitter during the COVID-19 pandemic, Journal of Communication in Healthcare, 14:1, 12-19, DOI: 10.1080/17538068.2020.1858222

[3] Du, J., Xu, J., Song, H. et al. Optimization on machine learning based approaches for sentiment analysis on HPV vaccines related tweets. J Biomed Semant 8, 9 (2017). https://doi.org/10.1186/s13326-017-0120-6

[4] Peng, H., Cambria, E., & Hussain, A. (2017). A review of sentiment analysis research in Chinese language. Cognitive Computation, 9(4), 423-435.

[5] Zhai, Zhongwu, Hua Xu, Bada Kang, and Peifa Jia. "Exploiting effective features for chinese sentiment classification." Expert Systems with Applications 38, no. 8 (2011): 9139-9146.

[6] Zhang, Pu, and Zhongshi He. "A weakly supervised approach to Chinese sentiment classification using partitioned self-training." Journal of Information Science 39, no. 6 (2013): 815-831.

[7] weibo-search. https://github.com/dataabc/weibo-search, accessed: April 20, 2021

[8] Jieba Chinese Text Segmentation. https://github.com/fxsjy/jieba, accessed: April 28, 2021

[9] Gene Diaz 2016. Stopwords Chinese (ZH). https://github.com/stopwords-iso/stopwords-zh.git, accessed: April 28, 2021

[10] Ng, Andrew. "CS229 Lecture notes." CS229 Lecture notes 1, no. 1 (2000): 1-3.

[11] Jurafsky, D., & Martin, J. H. (2014). N-gram Language Models. In Speech and language processing. Upper Saddle River, NJ: Prentice Hall, Pearson Education International.

[12] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

[13] "1.4. Support Vector Machines¶," scikit-learn. [Online]. Available: https://scikit-learn.org/stable/modules/svm.html#svm-mathematical-formulation. [Accessed: 02-Jun-2021].

[14] "1.1. Linear Models¶," scikit-learn. [Online]. Available: https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression. [Accessed: 02-Jun-2021].

[15] Ng, Andrew. "Recurrent Neural Networks." CS230 Lecture Slides, (2021).

[16] Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin et al. "Tensorflow: A system for large-scale machine learning." In 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16), pp. 265-283. 2016.

[17] Ritchie, Hannah, Esteban Ortiz-Ospina, Diana Beltekian, Edouard Mathieu, Joe Hasell, Bobbie Macdonald, Charlie Giattino, Cameron Appel, Lucas Rodés-Guirao, and Max Roser. "Coronavirus Pandemic (COVID-19)." Our World in Data (2020).

[18] Cui, Yiming, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. "Pre-training with whole word masking for chinese bert." arXiv preprint arXiv:1906.08101 (2019).