

Milestone Report for Analysis of Covid-19 Vaccine Sentiment

Yuming Chang
Georgia Institute of Technology
Atlanta, USA
ychang394@gatech.edu

1 INTRODUCTION

Social media has played more and more important roles in our daily life. People can freely share their opinions and comments and share real-time information on social media. The government, policymakers, and companies must understand the sentiment of users based on their posts to make better decisions. Because of the free expression on social media, that results in tons of good real-time news and information. Twitter is one of the largest social media, which has a huge amount of active users every day, which means it is important to analyze users' sentiments. Past few years, Covid-19 is a hot-spot topic for the entire society. The Vaccine of Covid-19 has also aroused controversy, which caused heated discussions and provided sufficient amounts of data to do sentiment analysis to analyze the opinions on the Covid-19 Vaccine. In recent years, many researchers have put effort into using machine learning models such as Support Vector Machine (SVM), Decision Trees, and CNN to analyze the sentiment of the text, which can also provide useful insights for misinformation detection. In this project, several machine learning and deep learning models will be implemented for misinformation detection and prediction.

Based on the sentiment analysis. Using the Natural Language Process (NLP) method, sentiment analysis can automatically extract the information from the text and classify the sentiment from those texts[1]. In order to use the machine learning model to analyze and predict sentiments, we need to convert the text into numeric values using Natural Language Process methods. So the problem in this project will be to convert the text information into a numeric value and train and test models using those data.

2 CHANGE FROM THE PROPOSAL

Compared with the original project proposal solely focused on general sentiment analysis of Twitter data, the modification is that I will focus on the Covid-19 vaccine misinformation detection. Based on the feedback on the project proposal from Professor, the project needs to be more related to urban computing problems. So in order to achieve this goal, I will adopt a new twitter dataset, which contains the tweets on the Covid-19 vaccine information to train and test the models I will implement. The detail of the new dataset will be discussed in Section 5.

3 RELATED WORK

In recent years, plenty of researchers have focused on using different methods to analyze the text's sentiment. Dey[2] has developed a Naive Bayes classifier to do the sentiment analysis. Assuming that each word is independent, the Naive Bayes classifier they developed is based on the Bayes theorem by using the frequency of the words and the text category as the feature to build the joint probability. The strength of the naive Bayes classifier is compared

with other machine learning models, and the naive Bayes classifier requires fewer data to train. However, the weakness of the naive Bayes model they proposed is that in this model, they assume each word appears independently, which in some cases is not valid, and each word may have a different meaning when they appear in the word combination. Astri[3] uses Multinomial Logistic Regression which uses the softmax activation function to do the logistic regression. The strength of this method is that it requires a small amount of data to train the model and does well even though the input features has a relationship of multi-linear. Moreover, the weakness of those methods is that this model assumes the input variable has a linear relationship with each other, which may not be accurate in the text classification. Tripathi[4] develop the random forest model to do the sentiment analysis on the Twitter text. The random forest model produces the output by combining the results of the different decision trees in the random forest model. Furthermore, in their research, they picked up some features from the dataset as the input feature, then based on the input features selected to determine the best number of nodes in each decision tree and built multiple decision trees. The strength of the random forest model is that this model can often generate good classification results and deal with missing data. However, the weakness of the random forest model is that sometimes they need lots of features to fit the model, but in the text classification task, we need to avoid the overfitting issue during the training period by carefully choosing several features. Also, we need to address that the training time of the random forest model may be longer compared with the primary decision tree model or other machine learning models. Rahman[5] developed the Support Vector Machine (SVM) model to do the sentiment analysis. In their research, they use the Principle Component Analysis method to find the essential features using the matrix build by calculating the weights of different features; then, they use grid research to find the best parameter in the SVM models. The advantage of the SVM model is that when the dataset is highly dimensional, the performance of the SVM model will perform well. However, the weakness of their work is that they use a small number of data to train their model in this research, but the problem is that the SVM model may perform worse when the dataset is too large, so we need to see how the performance of SVM model when dataset become large. Malarvizhi[6] has developed the Convolutional Neural Network (CNN) model to analyze sentiment. They first convert the different lengths of text data into the matrix with the same dimension and pad those matrices together. After that, they put the padded matrix into the embedding layer of the CNN model. After generating the embedding results, the embedding results will go through the convolutional layer to get the classification results. The strength of this method is that the CNN model has multiple layers, which enable the model to capture some vital information from the text. However, the weakness of the CNN model is that train the CNN model

needs a large amount of data, and it is time-consuming to train the CNN models. Swathi[7] proposed the Long-short Term Memory (LSTM) networks to investigate the sentiment of the tweet data. After cleaning the tweet text data by removing the special characters, usernames, and hashtags, they put the processed text dataset into the tokenization module and fed those data into the LSTM model they built. The strength of the LSTM model is that the LSTM model can handle the long sequence of text well compared with other models, but the weakness of the LSTM model is that it needs more data to train. There are also some research on Covid-19 information detection. Hayawi[8], et al. collected over 15000 tweets on Covid-19 vaccine and used several learning methods: XGBoost, LSTM, BERT transformer model to find the best perform model. In their research, they found BERT transformer model have the best performance based on their experiments, which achieve 0.98 F1-score and 0.97 precision. Mulahuwaish[9] et al. also collect 1375592 tweets about the Covid-19 vaccine and build the CNN+Bi-GRU and the performance of the CNN+Bi-GRU is better than Bi-LSTM model based on the data they collected, which achieve over 0.92 accuracy. Reshi[?] et al used different Lexicon-Based Methods: TextBlob, Valence Aware Dictionary for Sentiment Reasoning (VADER), AFINN to generate the sentiment based on tweet dataset about the Covid-19 vaccine and build some learning models to do the sentiment prediction. Based on their results, using TextBlob method to assign the sentiment of tweet data had the best performance.

4 TEXTBLOB

Based on Reshi[10] et al., the results using the TextBlob method to assign the sentiment have the best performance in their experiments. So in the project, I will also use TextBlob methods to assign the sentiment of the Covid-19 vaccine Twitter.

TextBlob is a lexicon-based method for tasks with raw text data in NLP. Textblob has 2918 lexicons, and they calculate the polarity score based on personal opinions or facts[10]. There is a library in Python, TextBlob can help us to calculate the polarity score based on the TextBlob methods, which return the corresponding scores. In this project, I will use the polarity score to assign the sentiment. The classification criteria are shown in Table 1, and algorithm is shown in Algorithm 1

Polarity Score	Sentiment	Numerical Value
<0	Negative	0
=0	Neutral	1
>0	Positive	2

5 DATA


In this project, COVID-19 All Vaccines Tweets[11] will be used to generate the sentiment, train, validate and test the models built. This dataset was stored in a CSV file—the detail of the dataset provided in Figure 1. From Figure 1, we can know there are 228207 data in this dataset. Moreover, have 16 features in this dataset. However, in this project, I analyze and predict the sentiment of tweets. So we do not need so many features. I will keep the id and text columns in this project in this dataset. The processed dataset is shown in Figure 2.

Algorithm 1 Assign Sentiment for tweet data

```

1: Using TextBlob compute the polarity score of each tweets
2: for iteration = 1, 2, ... do
3:     if polarity score<0 then
4:         Sentiment = 'negative'
5:         numeric_value=0
6:     if polarity score==0 then
7:         Sentiment = 'neutral'
8:         numeric_value=1
9:     if polarity score>0 then
10:        Sentiment = 'positive'
11:        numeric_value=2

```

2	133785819140118533		Your Bed	hai, hydra 	2020-06-25 23:00:28	10	88	155	False	2020-12-12 20:34:45
3	133785573991883577		Charles Adler	Vancouver, BC - Canada "CharlesAdlerSinger"	Hosting Global News Read... 2008-09-10 11:28:53	49165	3933	21853	True	2020-12-12 20:23:59
4	133785406484966912		Citizen News Channel	N/A	Citizen News Channel bringing you an editorial... 2020-04-23 17:58:42	152	580	1473	False	2020-10-12 19:17:18
...
228202	1460170772399865408		VaidLR	Bangaluru, India	Hourly updates on FREE and PAD 18+ and 45+ vs... 2021-09-21 08:44:34	31	0	0	False	2021-11-15 09:00:15
228203	1460163628626051841		VaidLR	Bangaluru, India	Hourly updates on FREE and PAD 18+ and 45+ vs... 2021-08-21 08:44:34	31	0	0	False	2021-11-15 08:30:26
228204	1460163204221851655		VaidLR	Bangaluru, India	Hourly updates on FREE and PAD 18+ and 45+ vs... 2021-09-21 08:44:34	31	0	0	False	2021-11-15 08:30:15
228205	1460156376995573795		Gati Valentini	Southern Africa	Entrepreneur, self taught cook & @Chloeas @Fer... 2019-08-28 10:31:43	8103	3113	45726	False	2021-11-15 08:03:03
228206	1460155671140134912		VaidLR	Bangaluru, India	Hourly updates on FREE and PAD 18+ and 45+ vs... 2021-09-21 08:44:34	31	0	0	False	2021-11-15 08:00:15

228201 rows × 11 columns

Figure 1: screenshot of dataset

	id	text
0	1340539111971516416	Same folks said daikon paste could treat a cyt...
1	1338158543359250433	While the world has been on the wrong side of ...
2	1337858199140118533	#coronavirus #SputnikV #AstraZeneca #PfizerBio...
3	1337855739918835717	Facts are immutable, Senator, even when you're...
4	1337854064604966912	Explain to me again why we need a vaccine @Bor...

Figure 2: processed datasets

5.1 Clean the tweets

Figure 1 shows some links, special characters like @ and hashtags, etc. In order to avoid the impacts of those meaningless special characters, we need to remove them from the data process. The code in this process is shown in Figure 3, and the results of the processed dataset are shown in Figure 4. From Figure 4, we can see the use of the code described in Figure 3. I have successfully removed all the special characters.

[illegible]

Figure 3: screenshot of code for remove special characters

	id	text	preprocess_tweet
0	1340539111971516416	Same folks said daikon paste could treat a cyt...	Same folks said daikon paste could treat a cyt...
1	1338158543359250433	While the world has been on the wrong side of ...	While the world has been on the wrong side of ...
2	1337858199140118533	#coronavirus #SputnikV #AstraZeneca #PfizerBio...	coronavirus sputnikv astraZeneca pfizerBioTec...
3	1337855739918835717	Facts are immutable, Senator, even when you're...	Facts are immutable Senator even when youre no...
4	133785406404966912	Explain to me again why we need a vaccine @Bor...	Explain to me again why we need a vaccine wher...

Figure 4: screenshot of processed dataset

5.2 Convert All The Words to Lowercase

Some research has also proved that the case of words may also influence the sentiment analysis results. So to avoid those influences, I am going to convert all the words to lowercase. The results of the converted dataset are shown in Figure 5. From Figure 5, we can see from the preprocess_tweet column that all the tweet text has been converted into lowercase

	id	text	preprocess_tweet
0	1340539111971516416	Same folks said daikon paste could treat a cyt...	same folks said daikon paste could treat a cyt...
1	1338158543359250433	While the world has been on the wrong side of ...	while the world has been on the wrong side of ...
2	1337858199140118533	#coronavirus #SputnikV #AstraZeneca #PfizerBio...	coronavirus sputnikv astraZeneca pfizerbiontec...
3	1337855739918835717	Facts are immutable, Senator, even when you're...	facts are immutable senator even when youre no...
4	133785406404966912	Explain to me again why we need a vaccine @Bor...	explain to me again why we need a vaccine wher...

Figure 5: screenshot of processed dataset

5.3 Using TextBlob to Generate the Sentiment of Tweet

After obtaining the processed tweet text, we need to use the TextBlob method to assign the sentiment of each Tweet. The code used to generate the sentiment is shown in Figure 6, and the results of processed data are shown in Figure 7. From Figure 7, we can see that the sentiment for each tweet has been successful. After that, I explore the distribution of sentiment in this dataset. The results are shown in Figure 8. The results show that the neutral class has the most number in this dataset, and the positive class has the lowest number.

5.4 Using TF-IDF to convert tweet into vector

In this project, I will build several machine learning and deep learning methods to predict the sentiment of the tweet on the Covid-19 vaccine. There we need to vectorize the tweet text. In the Natural Language Process, TF-IDF is the widely used statistical method. The equation of TF-IDF is shown below. In this project, I will use the TfidfVectorizer function from sklearn to vectorize the tweet text. The code used is shown in Figure 9. From the code, in this TF-IDF function, I currently set the max feature to 1000.

$$TF = \frac{\text{number_of_times_the_word_shown_in_documents}}{\text{number_of_words_in_documents}}$$

$$IDF = \log\left(\frac{\text{number_of_documents_in_the_corpus}}{\text{number_of_documents_in_corpus_contain_the_word}}\right)$$

$$TF - IDF = TF * IDF$$

```
from textblob import TextBlob

al_list=[]
nu_list=[]

for i in range(len(data2)):
    sent=TextBlob(data2['preprocess_tweet'].iloc[i])
    polarity = sent.sentiment.polarity

    if polarity < 0:
        al= 'negative'
        nu=0
    if polarity ==0:
        al='neutral'
        nu=1
    if polarity >0:
        al='positive'
        nu=2

    al_list.append(al)
    nu_list.append(nu)

data2['Sentiment']=al_list

data2['numeric']=nu_list
```

Figure 6: screenshot of code for assigning sentiment to tweet using TextBlob

	id	text	preprocess_tweet	Sentiment	numeric
0	1340539111971516416	Same folks said daikon paste could treat a cyt...	same folks said daikon paste could treat a cyt...	neutral	1
1	1338158543359250433	While the world has been on the wrong side of ...	while the world has been on the wrong side of ...	negative	0
2	1337858199140118533	#coronavirus #SputnikV #AstraZeneca #PfizerBio...	coronavirus sputnikv astraZeneca pfizerbiontec...	neutral	1
3	1337855739918835717	Facts are immutable, Senator, even when you're...	facts are immutable senator even when youre no...	negative	0
4	133785406404966912	Explain to me again why we need a vaccine @Bor...	explain to me again why we need a vaccine wher...	neutral	1

Figure 7: screenshot of processed dataset after assigning Sentiment

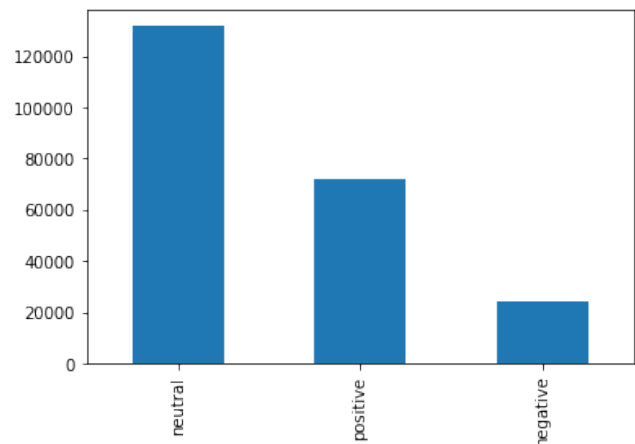


Figure 8: Distribution of each sentiment in this dataset

```
tfidf=TfidfVectorizer(ngram_range=(1, 2),min_df=2,max_features=1000)
tfidf.fit(data2['preprocess_tweet'])
tfidf_idf=tfidf.transform(data2['preprocess_tweet']).toarray()
```

Figure 9: TF-IDF Code

6 MODELS

6.1 CNN model

Through this milestone report, I have implemented a basic CNN model. CNN model has convolutional layers, which enables it to do text classification. In this project, I used TensorFlow to build the CNN model. The structure of the CNN model is shown in Figure 10. The CNN model I built currently has 5 convolutional layers based on Figure 10. I also use adam optimization function and a 0.001 learning rate.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 998, 1, 32)	128
max_pooling2d (MaxPooling2D)	(None, 998, 1, 32)	0
batch_normalization (Batch Normalization)	(None, 998, 1, 32)	128
conv2d_1 (Conv2D)	(None, 996, 1, 64)	6208
max_pooling2d_1 (MaxPooling2D)	(None, 996, 1, 64)	0
batch_normalization_1 (Batch Normalization)	(None, 996, 1, 64)	256
conv2d_2 (Conv2D)	(None, 994, 1, 64)	12352
max_pooling2d_2 (MaxPooling2D)	(None, 994, 1, 64)	0
batch_normalization_2 (Batch Normalization)	(None, 994, 1, 64)	256
conv2d_3 (Conv2D)	(None, 992, 1, 64)	12352
max_pooling2d_3 (MaxPooling2D)	(None, 992, 1, 64)	0
batch_normalization_3 (Batch Normalization)	(None, 992, 1, 64)	256
conv2d_4 (Conv2D)	(None, 992, 1, 64)	4160
flatten (Flatten)	(None, 63488)	0
dense (Dense)	(None, 64)	4063296
dense_1 (Dense)	(None, 3)	195
Total params: 4,099,587		

Figure 10: CNN model built currently

6.2 Graph Convolutional Network Model

The advantage of the Graph Convolutional Network Model is that it can deal with the structure dataset compared with CNN and RNN. In order to explore how the Graph Convolutional Network performance on this project. I also built a Graph Convolutional Network using Pytorch. The structure is shown in Figure 11. Currently, I used 4 graphs of convolutional layers in the GCN model. The current learning rate is 0.1. The optimization function is Stochastic Gradient Descent (SGD).

7 EVALUATION METHOD

In this project, the models will be evaluated based on the accuracy and F1 score. And based on the accuracy and F1 score to get the

```
hidden_channels = 16
dropout_probability = 0.5

class GCN(nn.Module):
    def __init__(self, dataset, hidden_channels):
        super(GCN, self).__init__()
        torch.manual_seed(98765)
        self.dataset = dataset

        self.conv1=GCNConv(dataset.num_node_features,64)

        self.conv2=GCNConv(64,32)

        self.conv3=GCNConv(32,hidden_channels)

        self.conv4=GCNConv(hidden_channels,data.num_classes)

    def forward(self, data):
        x, edge_index = data.x, data.edge_index
        x=self.conv1(x,edge_index)
        x=F.relu(x)
        x=F.dropout(x,p=dropout_probability,training=self.training)
        x=self.conv2(x,edge_index)
        x=F.relu(x)
        x=F.dropout(x,p=dropout_probability,training=self.training)
        x=self.conv3(x,edge_index)
        x=F.relu(x)
        x=F.dropout(x,p=dropout_probability,training=self.training)
        x=self.conv4(x,edge_index)
        out=F.log_softmax(x,dim=1)

        return out

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
gcn_model = GCN(data, hidden_channels=hidden_channels)
```

Figure 11: structure of proposed GCN model

best performance model through the project. The formula of the accuracy and F1 score is shown in Figure 5

Evaluation Method	Equation
Accuracy	$\frac{\text{Total Number of Correct Prediction}}{\text{All Number of prediction}}$
Recall	$\frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$
Precision	$\frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$
F1-score	$2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$

Figure 12: number of data of each sentiment

Also in the model modification process, the confusion matrix of each model will also be used to evaluate performance of each model

8 FUTURE WORK

Currently, I have built a basic CNN and GCN model. I will build a simple Support Vector Machine(SVM) model using sklearn. Furthermore, train, and test those models using the vector generated using TF-IDF based on tweet texts. Moreover, I will use the Evaluation method discussed above to evaluate the performance of each model. I will also do serval experiments, such as testing different numbers of convolutional layers in the CNN model and different graph convolutional layers in the GCN model. Different learning rates in each model to see the performance of each model and find

the best combination of parameters of each model and compare the performance.

9 DIFFICULTIES IN THIS PROJECT

The first difficulty in this project is that the dataset did not contain the ground truth label. So I need to use the TextBlob method to generate the sentiment of each model. Also, there are other methods in NLP to do sentiment analysis. So It will cost time to find the best methods.

The second difficulty in this project is constructing the vectors based on the tweet text using TF-IDF. How find the best parameters to generate the best vectors to ensure the best performance of each model is also time-consuming.

The third difficulty is that finding the optimal parameters of each model also requires more time.

REFERENCES

- [1] Doaa Mohey Hussein. A survey on sentiment analysis challenges. *Journal of King Saud University - Engineering Sciences*, 30(4):330–338, 2018. doi: 10.1016/j.jksues.2016.04.002.
- [2] Lopamudra Dey, Sanjay Chakraborty, Anuraag Biswas, Beepa Bose, and Sweta Tiwari. Sentiment analysis of review datasets using naïve bayes’ and k-nn classifier. *International Journal of Information Engineering and Electronic Business*, 8(4):54–62, 2016. doi: 10.5815/ijieeb.2016.04.07.
- [3] W.P. Ramadhan, S.T.M.T. Astri Novianty, and S.T.M.T. Casi Setianingsih. Sentiment analysis using multinomial logistic regression. *2017 International Conference on Control, Electronics, Renewable Energy and Communications (ICCREC)*, 2017. doi: 10.1109/iccerec.2017.8226700.
- [4] Jyotsna Singh and Pradeep Tripathi. Sentiment analysis of twitter data by making use of svm, random forest and decision tree algorithm. *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, 2021. doi: 10.1109/csnt51715.2021.9509679.
- [5] Mohammad Rezwanul, Ahmad Ali, and Anika Rahman. Sentiment analysis on twitter data using knn and svm. *International Journal of Advanced Computer Science and Applications*, 8(6), 2017. doi: 10.14569/ijacsa.2017.080603.
- [6] Usha Devi Gandhi, Priyan Malarvizhi Kumar, Gokulnath Chandra Babu, and Gayathri Karthick. Sentiment analysis on twitter data by using convolutional neural network (cnn) and long short term memory (lstm). *Wireless Personal Communications*, 2021. doi: 10.1007/s11277-021-08580-3.
- [7] T. Swathi, N. Kasiviswanath, and A. Ananda Rao. An optimal deep learning-based lstm for stock price prediction using twitter sentiment analysis. *Applied Intelligence*, 52(12):13675–13688, 2022. doi: 10.1007/s10489-022-03175-2.
- [8] K. Hayawi, S. Shahriar, M.A. Serhani, I. Taleb, and S.S. Mathew. Anti-vax: A novel twitter dataset for covid-19 vaccine misinformation detection. *Public Health*, 203: 23–30, 2022. doi: 10.1016/j.puhe.2021.11.022.
- [9] K. Gyorick M. Maabreh A. Gupta A. Mulahuwaish, M. Osti and B. Qolomany. “covidmis20: Covid-19 misinformation detection system on twitter tweets using deep learning models. *the 14th International Conference on Intelligent Human Computer Interaction (IHCI-2022)*, 2022.
- [10] Aijaz Ahmad Reshi, Furqan Rustam, Wajdi Aljedaani, Shabana Shafi, Abdulaziz Alhossan, Ziyad Alrabiah, Ajaz Ahmad, Hessa Alsuwailem, Thamer A. Alman-gour, Musaad A. Alshammari, and et al. Covid-19 vaccination-related sentiments analysis: A case study using worldwide twitter dataset. *Healthcare*, 10(3):411, 2022. doi: 10.3390/healthcare10030411.
- [11] Amartyanambiar. Covid-19 vaccine sentiment analysis, Sep 2021. URL <https://www.kaggle.com/code/amartyanambiar/covid-19-vaccine-sentimental-analysis>.