

COVID Vaccine Stance Classification

Sk. Aftab Aman¹, Meghna Chakraborty²

¹University of Engineering and Management, New Town Kolkata, West Bengal, India

²University of Engineering and Management, New Town Kolkata, West Bengal, India

Abstract

This paper discusses the work submitted by us for IRMiDis FIRE 2021 Task[2]. The goal of this task was to classify tweets related to COVID19 vaccines into three different sentiment classes. Our approach is based on using machine learning techniques to complete this 3-class sentiment classification problem. The evaluation scores of the submitted runs are reported in terms of accuracy and macro-f1 score. The accuracy reported for our classification was 0.448 and the macro-f1 score came out as 0.442.

Keywords

sentiment analysis, micro blogs, machine learning, 3-class classification

1. Introduction

The only way to stop the COVID-19 pandemic is through vaccines. However many people have a negative view on the vaccines. Politics and the fact that the vaccines have been rushed into production has led to numerous rumors and distrust among the citizens. It increases the necessity to understand the public's sentiment towards the vaccines. Social media platforms like twitter is one of the most popular places where users express their opinions. These tweets if exploited properly can be used to understand the sentiment of the public regarding COVID-19 vaccines.

Sentiment Analysis is a popular tool used to understand the sentiment of the people towards a particular topic. This can be achieved using labeled data and machine learning algorithms as well as different natural language processing techniques. In our approach we have used labeled micro-blogs (or tweets) for training and testing. We have used machine learning algorithms to find the stance of the people towards the COVID 19 vaccines.

2. Tasks

Our work discusses an automated approach to classify the sentiment of tweets into three classes. We explore the use of several machine learning algorithms and preprocessing techniques to design our code in order to do the following task.

Build an effective classifier for 3-class classification on tweets regarding their stance towards COVID-19 vaccines.

The 3 classes are described below:


Forum for Information Retrieval Evaluation, December 13-17, 2021, India

✉ aftabaman2000@gmail.com (Sk. A. Aman); meghnachakraborty12@gmail.com (M. Chakraborty)

🆔 0000-0002-0877-7063 (Sk. A. Aman); 0000-0001-7116-9338 (M. Chakraborty)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

Antivax - The tweet is against the use of vaccines.

ProVax - The tweet supports / promotes the use of vaccines

Neutral - The tweet does not have any discernible sentiment expressed towards vaccines or is not related to vaccines.

Below are samples of tweets showing various sentiments.

Tweet 1 : *Coronavirus: Some Canadians hesitant to take a COVID-19 vaccine* – Global News

Tweet 2 : *More good news!!! I could get used to this Covid-19 vaccine candidate is 90 percent effective, says manufacturer* <https://t.co/wtpyAh71pU>

Tweet 3 : *Moderna on track to report late-stage COVID-19 vaccine data next month.*

Tweet 1 is an AntiVax tweet , Tweet 2 is a ProVax tweet while Tweet 3 is a neutral tweet. Tweet 1 shows how hesitant some Canadians are to take the vaccine while Tweet 2 shows how the vaccines are a good news as its 90 percent effective. Tweet 3 gives us only facts about the vaccines and does not show any distinguishable sentiment. The tweets are perfectly matched.

3. Dataset

The data used for this task was gathered from twitter. The tweets were taken in the year 2020 and are based on COVID-19 vaccines. The entire data was made available in two phases :

- The training tweets were taken from the dataset provided by article [1]. It had the stances regarding COVID-19 vaccines taken between November-December 2020. We used 2792 tweets from this dataset for training and validation.
- The dataset taken for testing comprises of 1600 unlabelled tweets annotated by three crowdworkers and enjoy a majority agreement.

The dataset was slightly skewed as the count of the Neutral and ProVax tweets were more than that of the AntiVax tweets which could potentially bias the classification model.

4. Methodology

4.1. Preprocessing

This phase is the first and most important step for any text based problem.

For both Run1 and Run2 this process remains same. First we removed all the URLs i.e. words starting with https. The hash symbols are removed as they are common and appear in many tweets with hashtags. All the words starting with ' @ ' are pruned from every tweet. We then removed all the retweets to remove duplicates and thus remove some biasness. Next we divided the CamelCased words into independent words. CamelCase words are words whose first letter of the second word in a closed compound is a capital letter (example PayPal, iPhone etc.). Hashtags generally have such words as seen in the example below.

16km ENE of Nagarkot, Nepal: DYFI? - ITime2015-04-27 21:27:41 UTC2015-04-28 03:12...#Earth-Quake

Following this we converted the sentences into lowercase and removed all the emoticons, symbols, flags, pictographs, transport and map symbols because we deal with only textual data and the

unicode characters of these symbols are treated as random numbers and punctuations and do not help to detect sentiment. Next we dealt with some contractions and converted words like “haven’t”, “shouldn’t” into “have not” and “should not”.

After this we removed all the punctuations and all the stop words i.e. the words that occur in high frequency like a, the etc. (except no and not as they give us some knowledge about the sentiment of a tweet). After this we decided to lemmatize as stemming often gives us words that are not part of the vocabulary but a lemma always belongs to the language. The lemmatized tweets were then ready to be converted into vectors to be fed in our classifier.

4.2. Model Selection

After cleaning our data we had to transform it into a type understandable by the machine learning model. We used tf-idf vectoriser to transform each tweet to a vector. We considered unigrams as well as bigrams. We found through experimentation that this particular arrangement gave us the best result.

Then we tried feeding this into three different models that can classify each tweet into any one of the three classes. Naive Bayes, SVM and CNN. We worked with 2753 number of training data.

RUN 1 : We experimented with several learning algorithms. We checked the different models by checking the validation accuracy. We split the data into a test size of 30%. We got a validation accuracy of 0.733 with SVM and 0.724 with Naive Bayes. The confusion-matrix during validation for SVM is given in Fig 1

We picked Support Vector Machine to be used for our classification problem RUN 1 as the validation accuracy for SVM was more. We used rbf as our kernel function because this is not a linear classification problem. Then the model is trained using the preprocessed training data. The accuracy and Macro f2-score for the test data is shown in Table 1.

RUN 2 : For this run we experimented with Convolutional Neural Network (CNN). The pre-processing was same as done in RUN1. The maximum length of a preprocessed sentence was found to be 33, so we set maximum length of each tweet to 40. We padded each tweet. The train data was split into a test size of 30% during the training phase.

We used a sequential model and added an embedding layer. We followed that by adding a 1D CNN. Then we used a GlobalMaxPooling1D layer to down sample the input representation. We used a Dropout layer next to deal with some level of overfitting. At the very end a Dense layer with “sigmoid” as the layer activation function was used. We trained this model for 100 epochs. A visual representation of the model is shown in Fig 4. We can get an idea about the kind of fit and accuracy this gives us from Fig 2 and Fig 3.

Our neural network model was overfit as we can see from the graphs in Fig 2 and Fig 3. When the training data loss is very less and validation data loss is high (as in Fig 2) it means our model is suffering an overfit. The same inference can be gained from Fig 3. The categorical accuracy of the training data is much higher than that of the validation data. This also indicated an overfit. We used this and trained our model on our test data and found the result as shown in Table 2.

As CNN model in RUN 2 suffers from overfit and the accuracy is also lower as seen from Table 2 we discarded this run and considered RUN 1 as our primary run.

Table 1
Result using SVM

Accuracy	Macro f1-score
0.448	0.442

Table 2
Result using CNN

Accuracy	Macro f1-score
0.414	0.401

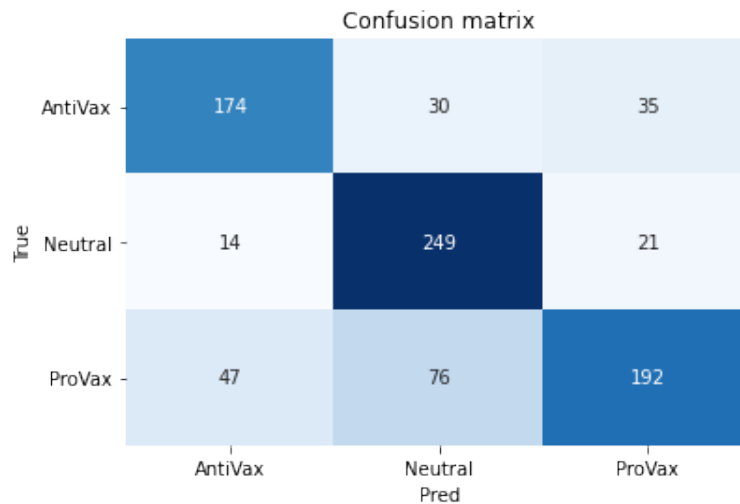


Figure 1: Confusion Matrix using SVM

5. Evaluation

The gold-standard for the classification is generated using manual runs. As mentioned in the IRMiDis Track, three crowdworkers are supplied with the tweets. The tweets have a majority agreement i.e. 2 out of 3 or all 3 agree annotate the tweet in a certain class. This proves that some of the tweets are subjective and thus likely to be falsely classified automatically. The run submissions are evaluated against the overall accuracy and the macro-F1 score. The macro-F1 score was the main judging factor.

The results of our submitted automated run are shown in Table 1 and Table 2. We were allowed to submit more than one runs and our primary automatic run submission got 8th place and we got 5th place as a team. We managed an accuracy of 0.448 and a macro-f1 score of 0.442 with the RUN1. With RUN2 we managed an accuracy of 0.414 and a macro f1-score of 0.401.

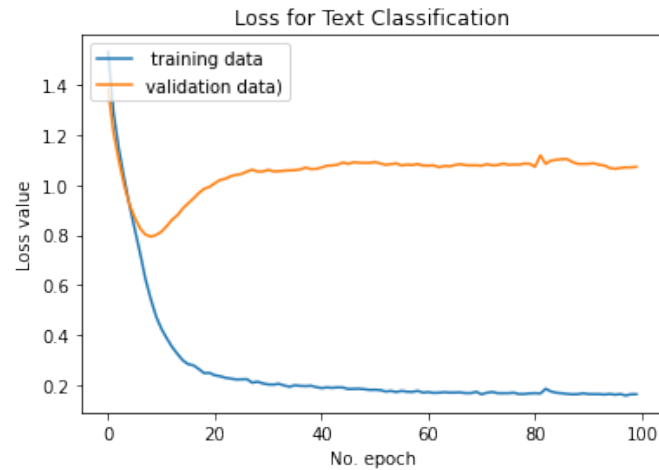


Figure 2: Loss vs No. of Epoch

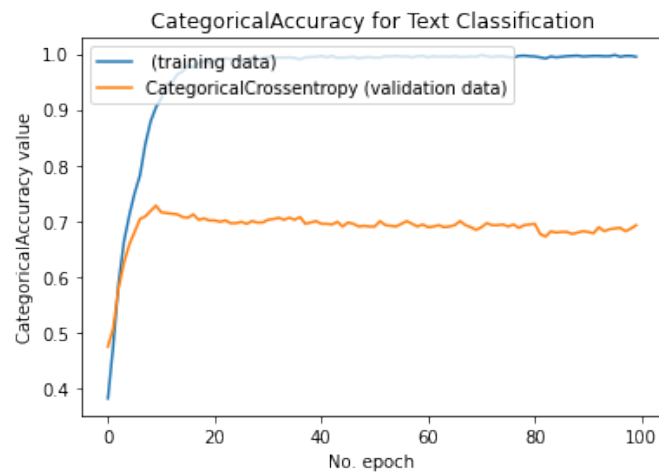


Figure 3: Categorical Accuracy vs No. of Epoch

6. Conclusion

In this work for IRMiDis FIRE2021 we used natural language processing preprocessing techniques and machine learning models to perform a three-class classification problem. We have tried various learning models and found the one that gives the best result. As a future extension of this work we plan to extend our knowledge of natural language processing and understand the relative sequence of words and the POS-tags to improve the performance of the model. The overfitting problem can be dealt with by tuning the hyperparameters. We can remove tweets which have 80% or more similarity to decrease biasness. We can also use methods to help deal with the imbalanced class problem which we ignored in our study.

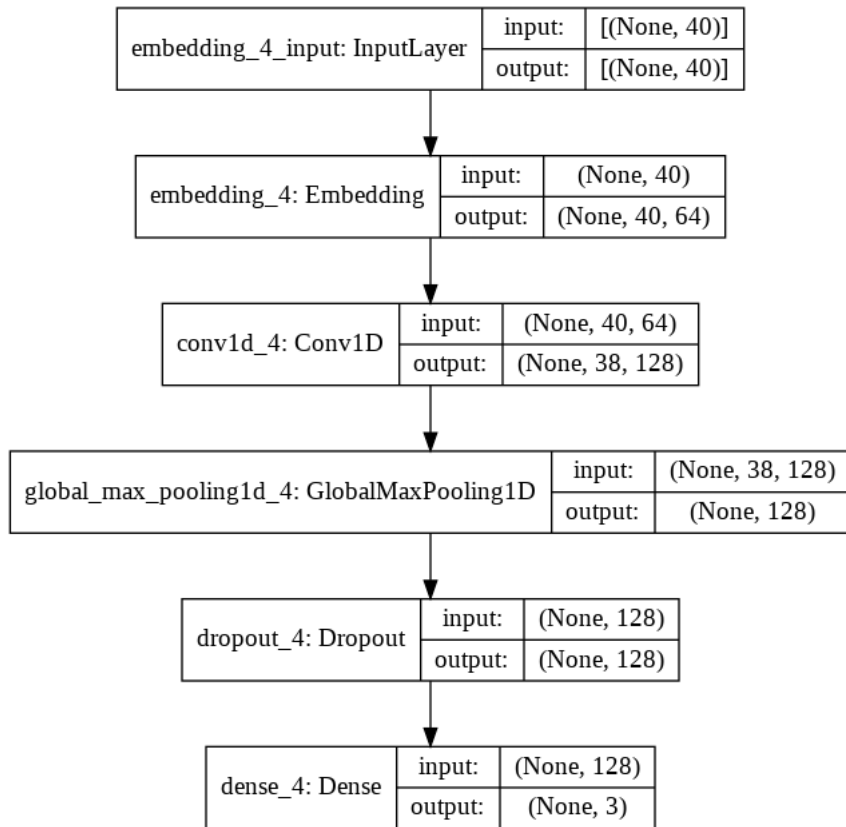


Figure 4: CNN model

References

- [1] L.-A. Cotfas, C. Delcia, D. S. Gherai, C. Ioanăș, I. Roxin, F. Tajariol, The longest month: Analyzing covid-19 vaccination opinions dynamics from tweets in the month following the first vaccine announcement, IEEE access (2021) 33203–33223. doi:10.1109/ACCESS.2021.3059821.