Identification of Informative COVID-19 Tweets

John A. Oliver, B.S. Environmental Sciences[1], Advanced Certificate, (Data Science)[2]
[1]University of Virginia, Charlottesville, Virginia; [2]Virginia Commonwealth University, Richmond, Virginia

Abstract

With COVID-19 still present in our society, people are forced to make decisions that will protect themselves from the virus and prevent the spread. Obtaining the information to make these decisions is critical .WNUT-2020 Task 2 focuses on this issue, and provides a challenge to solve it. In this paper, I demonstrate the effectiveness for Convolutional Neural Networks in the identification of Informative COVID-19 Tweets. Several word embedding methods are compared for accuracy and impact on the model.

Introduction

COVID-19 has changed how our world is operating today. So many aspects of the day to day lives of people all across the globe, have been altered to prevent the spreading and contraction of COVID-19. This virus has headlined the news in America, since all the way back in January of this year. The United States has been especially hit hard by the virus, having over 10 million confirmed cases as of November, 2020[1]. Part of what is making this virus so difficult to combat is that it is new to us. There was no prior research or understanding of this virus and we were left to try to learn about it as quickly as possible while it was spreading around the world.

After nine months of this virus wreaking havoc on the world, we are beginning to see hopes of vaccines in the near future[2], however, no vaccine has yet to be approved and there is no telling when that will happen[3]. The Information we know about COVID-19 continues to change as evident by the updated reports from the Center of Disease Control (CDC)[4]. People of the world are relying on information about COVID-19 to go about their lives safely. However, COVID-19 is being discussed by many people and sometimes discerning information can be difficult.

The goal of this project is to develop a system that can automatically detect informative tweets about COVID-19. Of the active Twitter users, around 71 percent use the platform to obtain news[5]. The identification of informative COVID-19 tweets would allow for citizens to find quick information on the virus that is communicated on Twitter. It would also allow for citizens and authorities to track the spread of the disease through communities. This will aid people in finding more information about COVID-19 that is available which may ultimately help people stay safe and make more informed decisions. Examples of Informative and Uninformative Tweets that this system would detect is shown in Figure 1.

Figure 1: Examples of COVID-19 Tweets with class.

No.	Tweet	Class
1.	“#COVID19 Updates. #SothSudan's fist positive patient is a 29 yo Dutch, who arrived in the country from Netherlands via Addis Ababa on 28 February. #SSOT HTTPURL"	Informative
2.	"@USER Carma & her family in San Jose, CA, are possibly infected with #coronavirus since 6 Feb in Seattle. Way way more than 1,000 cases in Seattle now... HTTPURL"	Informative
3.	"As #coronavirus spreads, Democratic Republic of Congo discharges last #Ebola patient HTTPURL via @USER"	Uninformative
4.	"The corona virus cases in Ghana is moving like the Eve people's dance.... It moves forward, comes back and move forward again ... 19 #Covid_19 cases now ehii"	Uninformative

Methods for this system were heavily influenced by by the recent success displayed by Deep Learning with Convolutional Neural Networks (CNN) for text classification[6][7]. Deep learning methods have been shown to perform well with various tasks such as object recognition and tweet classification. They provide several advantages compared to traditional machine learning algorithms as well. Deep learning models are able to train and work with higher-level hierarchical data which eliminates the need for engineered features that are usually implemented in traditional machine learning algorithms[8].The basis of this task is classification. Convolutional Neural Networks have been shown to perform well, and comparatively better, than traditional machine learning algorithms. For this reason, Convolutional Neural Networks have been used in this experiment.

In addition to the task of identifying informative tweets, I am also experimenting with different designs for the CNN. Neural Networks are built in layers and the number of possible layer combinations is large. Specifically, I am experimenting with the embedding layer of the CNN. I have created three different embedding layers for the model in attempt to find if one performs significantly better than another. With exception to the embedding layer, the three models are very similar in design.

Task and Data

The task and data from this project come from WNUT-2020 Task 2: Identification of informative COVID-19 English Tweets[9]. The task is to create a system that will automatically identify informative English Tweets, relating to COVID-19. Informative tweets are those that "provide information about recovered, suspected, confirmed and death cases as well as location or travel history of the cases." A data set of 10k tweets is provided, including 4719 tweets labeled as informative or uninformative. The tweets were hand annotated by three annotators with a inter-annotator agreement score of Fleiss' Kappa at 0.818. The data is split into training, validation , and testing data as described in table 1.

Table 1: Data set splits and class.

Data Set	Informative	Uninformative
Training	3303	3697
Validation	472	528
Testing	944	1056

Methodology

A CNN is made up of layers, including at least one convolutional layer, and each layer is able to detect increasingly more complex patterns in the data set. Rather than generated features that are used in traditional Machine learning algorithms, the CNN takes an embedding layer as an input. The embedding layer is generated from vectorized words. The layer is like a statistical representation of words. There are two ways of generating the embedding layer. The layer can be pre-trained before it is put in the model, or the words can be trained while the model is being trained. I am experimenting with two pre-trained methods and one train with the model method.

Three different embedding methods have been implemented for the CNN used. One word embedding was created using the Word2Vec model trained using the Tweets in the training data. The Word2Vec word embedding was created with the Gensim Word2Vec package, version 3.8.3[10]. Word2Vec works by taking the corpus of words given, in this case the COVID-19 training data, and it then creates a vector space of vectors. Each unique word in the corpus is assigned a vector in the space. The Word2Vec model is made up of layers. The first layer is the input layer where a vector with the length of the vocabulary filled with zeros except at the index of the word is put in. From there it is passed to a hidden layer of linear neurons and then to the output layer. Word2Vec helps in the classification of tweets because it creates similar outputs for words in similar context. In this case, it finds similarity amongst the words in informative and uninformative tweets. After the data has been converted to Word2Vec, it is inserted in the CNN as a pre-trained word embedding.

The second method is not a pre-trained word embedding. This method trains the word embedding during the training of the neural network. Here, every word in the tweet was tokenized which vectorizes the text corpus into a list of integers. To counter the problem of different length sequences, zeros were added to the length of sequences to match the length of the longest one. The embedding layer is built into the model with the addition of an added Flatten layer

that prepares the input for the Dense layer. The parameters of the embedding layer are the size of the vocabulary, the size of the dense vector, and the length of the sequence. Finally, a GlobalMaxPoolingID layer is added after the embedding. This takes the maximum value of all the features in the pool for each dimension. This essentially reduces the size of incoming feature vectors.

The final method is another pre-trained embedding called GloVe, which stands for Global Vectors. GloVe works similar to Wrod2Vec in that it obtains vector representations for words. It is different from Word2Vec in that it uses global statistics in addition to the local, where Word2Vec just uses local statistics. This could provide a possible advantage given some Natural Language Processing tasks. The GloVe embedding used here was created by the Stanford NLP group[11].

All data was preprocessed before its input into the models. In the processing of the data the stopwords were removed, hashtags and other twitter symbols were removed, emojis were removed, and user mentions and URLs were removed. The purpose of this data processing was to remove noise from the data. After the data was cleaned, the labels and the tweets were split apart. The tweets were tokenized using the Keras tokenizer. The labels of Informative and Uninformative were converted to binary 0s and 1s.

Experimental Details

The Convolutional Neural Network was developed with the Keras 2.3 package[12] on top of Tensorflow 2.3.1. For each word embedding, the CNN model is hyperparameterized to find the ideal number of filters and kernel size. Each model is comprised of the embedding layer, a convolutional layer, a Max Pooling layer, a flatten, and two dense layers. The GloVe embedding chosen has an embedding dimesion of 50. The Word2Vec embedding created has a dimension of 100.

Evaluation Metrics

To evaluate the quality of each model, the accuracy, precision, recall, and F1 score has been recorded. Precision will be the proportion of positive predictions that are actually correct. Recall, also known as sensitivity, will be the proportion of actual positives that were identified correctly. In this case, positive predictions are when the model predicts the tweet is Informative and negative predictions are for Uninformative tweets. Accuracy, precision, and recall were all produced using the Keras Metrics package. The F1 score was calculated using precision and recall.

Results and Discussion

Table 2: Data set splits and class.

Word Embedding	Accuracy (%)	Precision	Recall	F1 Score
Word2Vec	52.08	-	-	-
Keras Embedding	81.78	0.8362	0.8084	0.8220
Glove	79.41	0.8007	0.8051	0.8029

The Keras Embedding that trains as the model trains returned the highest accuracy, precision, recall, and F1 score (Table 2.). The GloVe embedding performed nearly as well to the Keras embedding in each evaluation metric. The Word2Vec embedding had a significantly lower accuracy at 52 percent. It did not perform well enough for the keras metrics package to produce precision or recall.

The results of these models were greatly enhanced by fine tuning the parameters of each model. Each model started with poor performances similar to the final performance of the Word2Vec embedding model. Changing the parameters of the model optimizer especially improved the performance of the GloVe and Keras embedding models. The best parameters were never fully discovered out for the Word2Vec embedding model.

Conclusion

In this experiment I was successful in creating a system to identify informative COVID-19 tweets with an accuracy of 81.78 percent. The Keras embedding and the GloVe embedding both performed at very similar levels. I have displayed

the abilities of simple Convolutional Neural Networks for text classification.

I am lead to believe that the poor performance of the Word2Vec embedding was likely due to the model not being built correctly for it. As mentioned, CNNs have the ability to be highly customized to individual tasks and the smallest of changes can make large differences in results. In creating the model for GloVe, I produced similar results to the Word2Vec model until I found parameters that suited it much better. This indicates to me that if I can find the best parameters for the Word2Vec embedding, my results could be much better. The other way that Word2Vec could have gone wrong for me was in the creating of the embedding. Unlike GloVe which I was able to download, Word2Vec I created using my data. Had the embedding been done incorrectly, the model would surely not perform well.

The development of the model using the word embedding that trains as the model seemed to be the easiest to implement and learn. This is likely the reason this model performed well. It was surprising to see it perform better than GloVe. Perhaps the GloVe model has more room for improvement that I was unable to achieve. In literature, models with pre-trained word embeddings were very popular and had great performances[7]. However, this does attest to the ability of Convolutional Neural Networks to perform well in text classification applications. These models created were relatively simple compared to other used in literature. I would describe these models as beginner models. I am confident that with more fine tuning and understanding of the building process of CNNs, these results could be further improved.

Other possible sources of error can be by both machine and human errors. Packages for data processing and model building may have been limited and not updated due to available operating system. This system was built using Python on a Windows Operating system. There were many struggles with downloading Tensorflow and making it operational. Some packages, such as Tensorflow, were unable to be fully updated to its most recent version. As for human error, the development, and implementation of a model was new to me. There was much to be learned and understood to develop models correctly and to tune them to their best performance. There is a possibility that the lack of knowledge or misunderstandings of research could have lead to models created that do not perform to their best, or that are built incorrectly. Additionally, there is always the possibility of bias, or patterns being built into the model due to the tweets being annotated. While the inter-annotator agreement was at a satisfactory level, there were only three annotators and patterns of bias is a possibility.

If this system were to be applied in the real world, there are several limitations that would need to be addressed. First of all, this model identifies informative tweets based on the language in the tweet. This system would have no way of identifying if the information is true or false and therefore could aid in the spread of false information. Secondly, this system using Deep Learning methods performs well in instances like this where there is an abundance of data, but if the amount of data was limited, this model may not perform as well. If this were to be applied to an event that is new and there is not a lot of data on it, traditional machine learning algorithms would be likely to perform better[6]. Finally, we would need much better results before we could use this system to extract information. With an accuracy in the 80s, this indicates the model does well, but for something as important as COVID-19, we would want a greater accuracy. Especially if you apply this accuracy to much larger data sets, there is still a large number of tweets missed or falsely identified as informative.

The results from these models are congruent with research by showing Convolutional Neural Networks are able to perform well in the classification of Tweets. This indicates that this model and other similar deep learning models could be beneficial to continue studying. Given the success of this experiment, a good avenue of future work would be to further develop this model with the CNN. As shown with this experiment, different embedding methods can greatly effect the model, suggesting more experiments with more embedding methods could be beneficial. In particular, experimenting with n-grams in the Convolutional Neural Network may provide interesting results given the ability of the CNN to use higher n-grams than most traditional machine learning algorithms.

References

1. The Atlantic. The covid tracking project. 2020.

2. Pfizer. Pfizer and biontech announce vaccine candidate against covid-19 achieved success in first interim analysis from phase 3 study. 2020.

3. New York Times. Coronavirus vaccine tracker. 2020.

4. Center For Disease Control and Prevention. Coronavirus disease 2019 (covid-19). 2020.

5. Elisa Shearer and Katerina Eva Matsa. In *Pew Research Center*, 2018.

6. Seyed Hossein Ghafarian and Hadi Sadoghi Yazdi. In *Information Processing Management*, 2020.

7. Venkata Kishore Neppalli, Doina Caragea, and Cornelia Caragea. In *Social Media Studies*, 2020.

8. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. In *Nature 521*, 2020.

9. Dat Quoc Nguyen, Thanh Vu, Afshin Rahimi, Mai Hoang Dao, Linh The Nguyen, and Long Doan. In *Proceedings of the 6th Workshop on Noisy User-generated Text*, pages 314–318, 2020.

10. Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

11. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. https://nlp.stanford.edu/projects/glove/, 2014.

12. François Chollet et al. Keras. https://keras.io, 2015.