**Pandemic Tweet Challenge**

**Group Names:** Yudong Lu, Fengyi Zhao, Javan Reuto, and Bradley R. [Mascarenhas](https://umd.instructure.com/courses/1314706/users/4280535)

**Introduction:**

Social media has become the main source of communication and a method of extracting public opinions and certain topics related to the nature of the dataset. The problem we are trying to solve is deriving meaningful insight into public opinion regarding largely controversial issues. This dataset is regarding COVID-19 and is a controversial topic on different levels whether lockdowns are placed in action, vaccines, or mask mandates. It is difficult to analyze these opinions on a surface level. Building a model that gives insight into the source of controversy in these public opinions gives public officials the ability to speak to them with greater understanding on the topic, demographics, segments, and trends.

Sentiment analysis is a widely used method currently used to find accuracy of the application being used. Although widely used it faces challenges such as polarity, sarcasm, comparative sentences, emojis; our goal is to create a model which can learn from these issues and improve as the model is trained. We plan to use the LSTM model and Fully Connected Feedforward model because both allow for a training component to sentiment analysis for the different tweets. Both models implement the technique known as gradient descent over random forest and gradient boosting leading to successful predictions and higher accuracies.

**Data: description, characteristics and potentially some data exploratory analysis**

The data is a collection of thousands of tweets related to the coronavirus with corresponding sentiments, which is divided into training and testing data. We got our data from Kaggle. There are five categories for the sentiment of tweets, Neutral, Positive, Negative, Extremely Positive and Extremely Negative. From the visualization we can see that there are around 8000 neutral, 12000 Positive, 5900 Extremely Negative, 10000 Negative, and 7000 Extremely Positive tweet sentiments.

The tweets contain three different elements: text, @, and #. The text in tweets are the main object to be analyzed for the project, as a result, we need to clean the data to remove @ and # content to get the text for sentiment analysis.

**Describe some existing methods (can be considered as your baselines for comparison):**

1. Sentiment analysis can include unsupervised or supervised techniques. Unsupervised techniques are still existing methods used such as corpus of words associated with polarity and sentiment involved. Polarities play a role in this technique and the category is assigned based on the polarity.
2. Topic modeling is an NLP method which analyses topics in text. The documents that are present here are the tweets and amongst the COVID-19 there can be topics derived such as vaccine, quarantine, and mask mandates. The tweets are “documents” that include various topics; these topics can include multiple words that would be associated with the topic. This is valuable however using methods such as tokenization and tf-idf allow for the importance of words that could be in topics through the vector representation. Bi-grams and further can also pick up the importance of surrounding words and context.
3. WordCloud is a method of visual representation of various words in which greater prominence is given to words of higher frequency relative to others. Documents that are in the COVID-19 tweets can be analyzed for the most frequently occurring words. After filtering out common words, the importance of words can be determined through frequency because they are being represented a lot in the documents because it’s the topic rather than a common word in the language.
4. KNN classification of Word2Vec is another method which is used for popular representation of words by capturing the linguistic context of them. The algorithm leverages machine learning to determine word associations from a large corpus. This has application to COVID-19 tweets as we can understand the context behind popular words to get greater insight than what frequency can provide alone. Average accuracy 53.0% for tweet sentiment which is 4.02% less than the fully connected feedforward network and 21.90% less accurate than the LSTM model we created.

**Your method:**

Before applying the models, we preprocess the data by the following steps. First we take only the Tweet and Sentiment from the original dataset. Second, we convert sentiment into categorical labels, that is ‘Extremely Positive’ as 4, ‘Positive’ as 3, ‘Neutral’ as 2, ‘Extremely Negative’ as 0. Then we convert the text to lowercase and remove websites and non-text.

1. Fully connected feedforward network: We apply the tf-idf on the text to receive the vector representation of the top 2000 features or sequences.
   1. Tf-idf: We used this method to create the vectorization and transformation of sequences.
   2. First input layer has 3000 neurons with the input dimensionality of the 2000 max features with the activation function ‘relu’.
   3. Fully connected layer with dropout rate - 0.2
   4. Second layer has 2000 neurons; activation function ‘relu’.
   5. Fully connected layer with dropout rate - 0.2
   6. Output layer has 5 neurons for the 5 class classification of sentiment leading to the use of softmax. After this we compile the network using categorical crossentropy as the loss function.

We received an accuracy of 57.02%. There is some overfitting which leads to the reasoning of using the LSTM model below.

1. LSTM: We define the number of max features as 10000 and use a tokenizer to vectorize and to convert the text into sequences so the network can deal with it. Then we create the sequential model and add layers to compose the LSTM network. The layers are as follows:
   1. An embedding layer that converts word tokens into embedding of specific size, in our project we add an embedding layer with input dimensionality of 10000, and output dimensionality of 128
   2. An dropout layer with the spacial dropout rate of 0.4, which dropping entire sentence in batch
   3. We then add the LSTM layer, which defined by hidden state dims and number of layers, with dropout for input and neurons in hidden layers of dropout rate of 0.2
   4. We also add fully connected layers with 0.2 dropout rate, and by using softmax as the activation function. After this we compile the network using categorical crossentropy as the loss function.

Then the dataset is split into training and testing sets and we run the model. We received accuracy around 76.38%

**Results:**

A 2-class sentiment generally will have an 80 % accuracy rate. We do not have an exact baseline for a 5-class sentiment analysis, however we can compare the result to the 2-class sentiment analysis, since it is more difficult to have a higher level classification of sentiment.

In our project, the fully connected feedforward network gets a 57% accuracy and the LSTM model reaches a 76% accuracy in reaching. The fully connected feedforward network uses tf-idf as the input to analyze importance for each word but not consider the sequence and content of the tweet. This will result in an overfitting issue of having a high accuracy in training data but low accuracy in testing due to same word used but different content and meaning in sentences.

The LSTM model gets a 76% accuracy in categorizing 5-class sentiment. By taking word location in sentences, and also considering their order, the full meaning of the sentence and also the level of positive and negative can be analyzed in the model. Since the project is a 5-class sentiment analysis, LSTM seems to be the better choice to complete this task and get a higher accuracy.