Real-Time Twitter Emotion Classification by Topic

Christian Norrie
Filotas Theodosiou
(Group 1)

March 22, 2020

Abstract:

The project has attempted to determine general public opinion about any topic in terms of the emotions inferred from relevant tweets. Determining how people feel about a trending topic can be a useful metric for evaluating advertising efficacy and projecting profits, among other uses. A dataset of approximately 50000 tweets was queried through the Twitter developer API manually and then preprocessed by classifying them as belonging to one of four classes of emotions: happiness, sadness, fear, and anger. Machine learning models, specifically single layer LSTM, two layer LSTM, singer layer bi-LSTM, and two layer bi-LSTM, were trained to classify the emotion and topic of tweets. The model with the both the highest accuracy and F1 score, and smallest difference between training set accuracy and validation set accuracy was two layer bi-LSTM. After the models were trained, classification of public opinion of several topics in terms of emotion was attempted on streamed tweets on a fine-tuned two layer bi-LSTM model and the results were visualized through a Kibana dashboard.

1. Introduction

Online social media platforms have become spaces where people express their opinion and feelings about events around the world. Big data programming and machine learning models provide tools to analyze this data and determine how people feel about various topics. Sentiment analysis is a collection of techniques that determine if social media users feel positively, neutral, or negatively about a topic or event. The resulting sentiment can then inform future decisions related to the topic or event. For example, if a marketing agency wants to determine the efficacy of their advertising of a product, they can run sentiment analysis techniques on social media posts relating to that product. If public opinion about the product is determined to be positive, the marketing agency can claim that their advertising was successful, and conversely if public opinion is negative, the marketing agency may seek to make changes to their advertising campaigns. Sentiment analysis can be quantified in such a way that public opinion can be expressed through a numerical value, such as ‘social media users feel 92% positive about our product’.

This project has attempted to quantify public opinion about topics or events based on four discrete classes of emotions: happiness, anger, sadness, and fear. These classes, which are considered to be four of the most basic human emotions[3], were chosen based on prior research of Twitter data and how different emotions are represented and classified on social media[1]. Identifying if the public feels anger or fear about a trending topic can inform analysts about future decisions when it comes to fields other than business, such as politics. For example, if analysts are attempting to determine how to respond to a disease outbreak, the government response will be different if public opinion of the disease outbreak is anger as opposed to fear. Correctly identifying public opinion in such cases can avoid an unfavourable government response, and could avoid catastrophe. This differs from sentiment analysis - in this case, sentiment analysis would only imply that public opinion is negative (which might be

considered a rather obvious conclusion to make about a disease outbreak), rather than identifying the specific emotion.

Twitter currently provides around five hundred million tweets per day on a wide variety of topics. Most of the time, these tweets are followed by a number of hashtags which include both the topic the user is writing about and some indication of their feelings on the topic. For example, "*I still don't know how @realDonaldTrump is the president... #Trump #anger*" identifies both the topic (Trump, presumably the American president) and the emotion (anger) in the hashtags alone. Another example might be "*going to the hospital for a #coronavirus checkup #nervous"*, which again, through the hashtags alone, identifies the topic (coronavirus, presumably COVID-19 at the time of writing) and the emotion (fear).

We have attempted to create a neural network trained on tweets such as the above examples that identify the emotion of the general public. This model can then be used to classify new tweets obtained via streaming and identify current public opinion (in terms of emotion) about new topics. In this iteration of the neural network, tweets belong to exactly one class of emotion. For example, the tweet "*I can't believe the #WHO isn't doing anything about #coronavirus! #fear #anger*" can be classified as 'fear' or 'anger', but not both. Candidate models trained were single layer LSTM, two layer LSTM, singer layer bi-LSTM, and two layer bi-LSTM, each with varying architectures outlined in the model descriptions section.

There are many uses for the outlined system. Other than the examples already specified, some uses might include informing health analysts of public opinion of a disease outbreak, guiding what sort of information and healthcare campaign they might wish to run to counter said outbreak. Another example might be guiding marketing decisions - if a horror movie releases a trailer and the public opinion is not generally fearful, the marketing team may wish to delay the release until they can offer a more frightening trailer.

2. Method:

Approximately 50000 tweets were queried through the Twitter API to create a dataset for training machine learning models. The decision to query our own dataset instead of using an existing dataset was made because Twitter restricts publicizing datasets queried by researchers and as such many existing datasets are simply not large enough for the purposes of this project - many datasets available online only contain 1000-1500 tweets. With that in mind, the steps taken from data collection to classifying streaming data are as follows.

1. Query Twitter and retrieve a dataset of tweets. More information on how the data was queried can be found in the data collection section.
2. Apply heuristics to validate the quality of the retrieved tweets. More information on these heuristics can be found in the data collection section.
3. Clean dataset (stemming, tokenizing, applying class labels, removing hashtags). Specifics can be found in the data cleaning section.
4. Train models on cleaned dataset. In this case, four different model architectures were trained, including single LSTM and bi-LSTM.
5. Evaluate models based on accuracy and F1 score, and selecting the highest performing model. Accuracy was selected as our first evaluation method to show us which model had more correctly classified tweets, and F1 was selected as a second evaluation metric as it gives us information about the incorrect classified tweets.
6. Stream new data from Twitter (applying the cleaning pipeline from earlier) based on a hashtag defining a topic, then use the model chosen earlier to classify public opinion (based on emotional state) of a trending topic.

2.1. Data Collection

The training dataset contains tweets and their class label (happiness, sadness, fear, or anger). One of the difficulties in acquiring such a dataset is automatically classifying a tweet as belonging to one of these emotional classes. Querying a large volume of tweets is fairly trivial but classifying them manually is not. The solution proposed was to only query tweets with specific hashtags that imply a class label. This methodology is supported by studies performed by Saif Mohammed in 2012[2] which demonstrates that when Twitter users self-label their tweets as an emotion, that emotion is accurately reflected in the tweet. For example, if a user uses a hashtag of “#happy”, it can be classified as having the label “happy” without the need to scrutinize the tweet further. This may seem obvious, but removes any ambiguity in that there is no need to analyze tweets wherein the emotional state cannot be immediately indicated through its hashtags. This methodology is further supported by another study performed in 2012[5] in which seven emotional states were specified and a dataset of 2.5 million tweets were each classified as indicating one of the specified emotional states. These seven emotional states are considered to be the ‘most important’ and distinct enough from the others[3], but for the sake of time constraints and model’s performance (multiclass classification problems with seven classes may be more ambiguous than with four classes), our study has pared it down to four of these

states: joy, sadness, anger, and fear. Our research further differs from prior research in that we are attempting to classify public opinion, in terms of emotion, of various topics, whereas previous research uses the information to find patterns in terms of network usage (i.e.: what times of day are people more likely to tweet in various emotional states?).

As noted earlier, tweets with hashtags specifically defining an emotion usually indicate the emotion of the user. Therefore, tweets were queried based not exclusively on the four words of class labels (i.e.: #joy, #sadness, #anger, #fear), but synonyms of the class labels as well (Appendix A). Synonyms were retrieved based on the most commonly used words for each class label, supported by studies[3][1] as to which words most strongly correlate to a specific human emotion. Once a list of commonly used hashtags for each emotion was defined, the tweets were queried using Tweepy and the Twitter developer API. Only tweets containing one of the hashtags on the list of synonyms were queried, which were then automatically classified as indicating a specific emotion corresponding to the synonym (such as '*#scared'* or '*#nervous*' both being automatically classified as 'fear'). The tweets were saved in a csv with three columns: the tweet number, the body of the tweet (which has been preprocessed as well, details of which will be provided below), and a class label automatically retrieved from the hashtags present in the tweet. For example, the tweet "*Going to the hospital for a #coronavirus test… #nervous*" was automatically assigned to the class "fear" based on the hashtag of "#nervous".

After querying a week's worth of tweets, the dataset was then filtered based on a set of heuristics proposed by Wang, Chen, and Thirunarayan[5]. The heuristics used are as follows.

1. Removing tweets with an excessive number (in this case, five or more) of hashtags as these might be bots or generally be less indicative of a specific emotion
2. Removing tweets where the hashtags are not at the end of the tweet body as they are less likely to clearly indicate the writer's emotion[1].
3. Removing tweets with three or less words. These tweets generally do not indicate an emotional state of any kind.
4. Removing tweets containing non-English characters, as the trained model can only process English language tweets.
5. Removing retweets as they may have already been queried.
6. Removing replies as determining emotion depends on the context of the tweet being replied to.

To ensure the model is not biased, the number of tweets belonging to each class should be roughly the same. Interestingly enough, most tweets were classified as happy and the category far outweighed the size of the other categories, containing over 15000 tweets, whereas each other class only contained around 5000. In this case, more data is assumed to result in a more accurate model, so each other category was supplemented with other previously mined datasets used for competitions on similar tasks[21][22]. The Twitter API only supports querying a maximum of 2500 tweets per 15 minutes and only contains tweets from at most one week prior, so supplementing the training datasets with other publically available datasets was a necessity

to achieve a ‘sufficient’ data volume, along with querying for two additional weeks. An additional complication lies in defining how many tweets are a sufficient data volume. Intuition implies that more data would lead to a better-performing model, but due to time constraints, a final dataset of only 48815 tweets was compiled.

Class	Anger	Fear	Happiness	Sadness
Number of Tweets	11181	9536	14296	13802

2.2. Data Cleaning and Model Selection

To justify choices for how the dataset was cleaned, candidate models must first be considered in order to identify what kind of data they can process.

When it comes to Natural Language Processing (NLP), most methods used in the past were machine learning models trained on hand-crafted features such as n-grams. In addition, because of the high dimensionality of the BOW (Bag of Words) representations, the “curse of dimensionality” is inevitable.The extensive feature engineering needed, especially in language used on social media (ie: slang, abbreviations)[6], was the main reason we did not choose a traditional machine learning model (such as support Support Vector Machines, Bayesian Networks, Maximum Entropy). Furthermore, word embedding techniques utilizing deep learning models generally outperform shallow machine learning models[7], such as Naive Bayes, Support Vector Machines (SVM), or logistic regression.

The models used for these tasks were recurrent neural networks (RNN); more specifically, long short-term memory (LSTM) models. This family of neural networks was suitable for our task since they excel in tasks involving sequential data such as text. LSTM models are variants of RNNs that overcome the vanishing gradient problem of RNNs[8] and have achieved great results on NLP tasks.

Single LSTM and bidirectional-LSTM models were trained on the queried Twitter data and their results were compared. As word embeddings are necessary for LSTM models, multiple approaches were considered, such as training embeddings based on our dataset, using a pre-trained embedding layer like GloVe[12], or training an embedding layer on a pre-existing Twitter dataset. Additionally, pretrained models for NLP tasks such as GTP-2[9] can be fine-tuned through transfer-learning for our tasks and can then be compared with our own networks. The rationale for utilizing these techniques was that despite their original purpose of text generation, they contain all necessary information for NLP which can be repurposed for our models. With this in mind, GloVe was imported pre-trained for the word embeddings layer[12], reducing the training time of our models and hopefully increasing model accuracy. GloVe has previously been trained on millions of tweets, so it should provide a strong basis for word embeddings. Initially, we used a 3 layer neural network, consisting of the embedding layer, a

single LSTM layer, and a fully connected output layer (4 outputs, each corresponding to the appropriate class). From there, multiple models were iterated upon.

To facilitate model training, the training data needed to be cleaned further. Below is a list of tasks completed for data cleaning. These steps were done using the ekphrasis library[11], a github dictionary[19], and emoji library[20].

- Replacing URL's with "<url>"
- Replacing usernames with "<username>"
- Spelling correction
- Stemming:
    - Converted contractions to their complete form (ie: "I've" to "I have", "could've" to "could have")
    - Replacing elongated words with their grammatically correct form (ie: "maaaaaad" to "mad")
    - Removed hashtag symbol (ie: "#example" to "example")
    - Segmented the words produced after a hashtag in the step before, by splitting them into actual words(ie: "#nojobinbiology" to "nojobsinbiology" to "no jobs in biology")
    - Transforming emojis and emoticons into meaningful phrases (ie: ":)" to "smiley_face").

- Replacing slang words and commonly used internet abbreviations with their grammatically correct form (ie: "gr8" to "great", "lol" to "laughing out loud")
- Removing hashtags that indicated an emotion (on the list of synonyms in appendix A) so the model does not just learn the hashtags and is instead forced to learn based on the tweet content

Below are some examples of what the tweets look like when cleaned according to these outlined steps.

Tweet Body	**Class**
<user> oh my gosh shit is getting real fast ! game changer !	**Fear**
so i have to say bye to <number> people i really care about all in one day <sad> and	**Sadness**
the sun is shining and the skys are blue . its going to be a good day . sun_with_face sun grinning_face good day its friday	**Happy**
<user> i thought you guys were cleaning the trains now ? mold all over the air vents on car <number> am train <url>	**Anger**

The models were then evaluated using accuracy and F1 as the evaluation metrics. Ideally, 10-fold cross validation would have been applied but due to the excessive training time, this proved to be unrealistic.

3. Results:

3.1. Model Descriptions

Multiple LSTM networks were trained with the following architectures.

Model	**LSTM Layer Type**	**Number of LSTM Layers**	**Number of Nodes Per Layer**	**Optimizer**	**Loss Function**	**Modifications**
Single layer LSTM	Single LSTM	1	256	adam	Categorical Cross Entropy	
Single layer LSTM w/ attention mechanism	Single LSTM	1	256	adam	Categorical Cross Entropy	Attention Mechanism
Stacked LSTM	Single LSTM	2	1. 256 2. 128	adam	Categorical Cross Entropy	
Stacked LSTM w/ attention mechanism	Single LSTM	2	1. 256 2. 128	adam	Categorical Cross Entropy	Attention mechanism
Single layer bi-LSTM	Bidirectional LSTM	1	256 (each way)	adam	Categorical Cross Entropy	
Single layer bi-LSTM w/ attention mechanism	Bidirectional LSTM	1	256 (each way)	adam	Categorical Cross Entropy	Attention Mechanism
Stacked bi-LSTM	Bidirectional LSTM	2	1. 256 2. 128 (each way)	adam	Categorical Cross Entropy	
Stacked bi-LSTM w/ attention mechanism	Bidirectional LSTM	2	1. 256 2. 128 (each way)	adam	Categorical Cross Entropy	Attention mechanism

Despite initial promising results, the first model trained (single layer LSTM) overfitted on the training set after the 15th epoch and the accuracy on validation data was not as high as we expected. As a result, we added a second LSTM layer with fewer nodes and increased the dropout rate to prevent the model from overfitting.

Next, we constructed more complex architectures by training models with a bi-LSTM[13] layer and later with 2 bi-LSTM layers. Finally, we added an attention mechanism, similar to other LSTM models[11][15], on each of the trained models. In using the attention mechanism, we wanted to test our assumption that aggregating the results of every hidden state on the LSTM layer (based on their importance) would give us better results.

3.2. Performance

To evaluate the goodness of each model, validation set and test set accuracy are used as a baseline metric. To achieve a deeper understanding, we also used the aggregated F1 score for each class. F1 score is the harmonic mean between precision and recall and is a better measure for the incorrectly classified tweets.

Model	**Training Accuracy**	**Validation Accuracy**	**F1 score**
Single layer LSTM	0.865	0.686	0.6741096
Single layer LSTM w/ attention	0.846	0.699	0.6718419
Stacked LSTM	0.842	0.690	0.6822054
Stacked LSTM w/ attention	0.809	0.689	0.6989982
Single layer bi-LSTM	0.868	0.684	0.674109
Single layer bi-LSTM w/ attention	0.901	0.684	0.68310016
Stacked bi-LSTM	0.699	0.699	0.7034837
Stacked bi-LSTM w/ attention	0.844	0.691	0.68978685

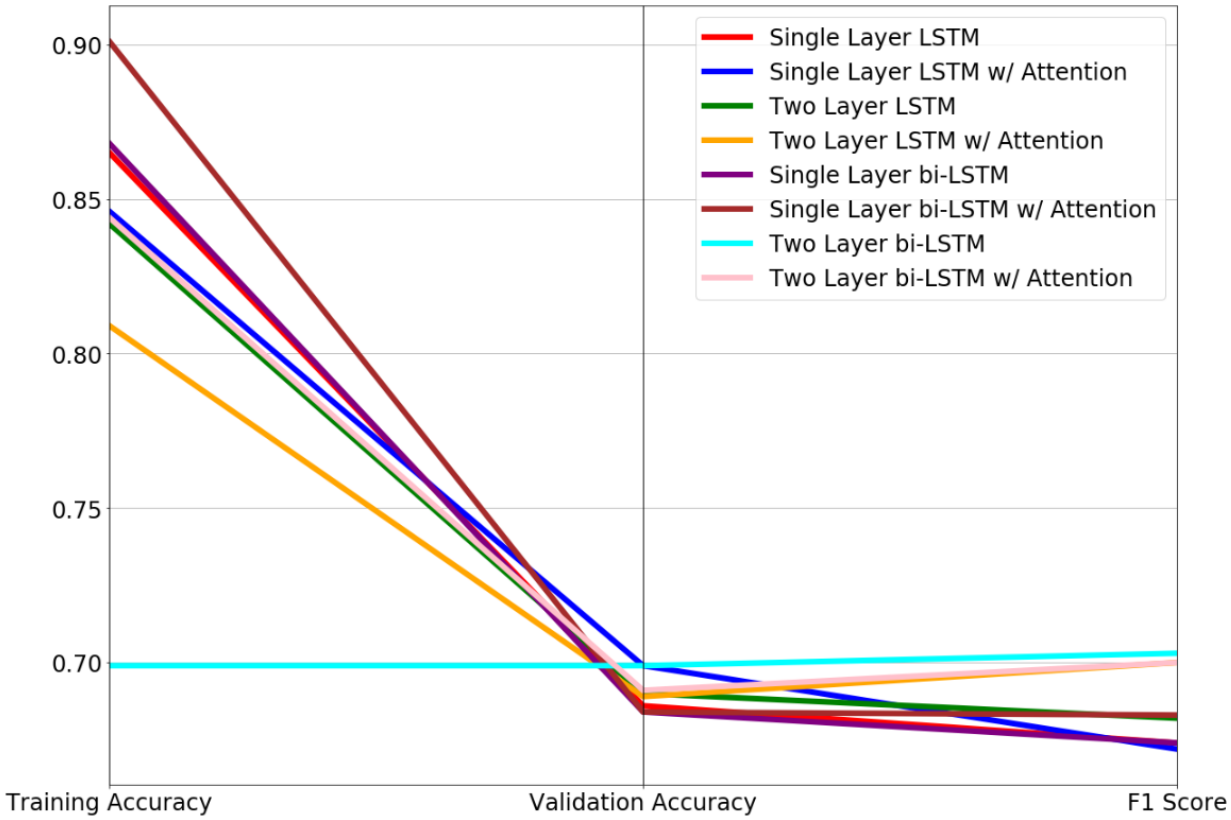

Fig. 1: Comparison of model evaluation metrics

To further understand how each model performed and identify which classes our model failed to distinguish, we calculated the confusion matrices of each model and we observed how models classified tweets from every class.

Single Layer LSTM				
	Anger	**Fear**	**Happy**	**Sad**
Anger	**0.56**	0.1	0.05	0.27
Fear	0.12	**0.62**	0.06	0.18
Happy	0.02	0.04	**0.84**	0.08
Sad	0.14	0.1	0.08	**0.66**

Single Layer LSTM w/ Attention				
	Anger	**Fear**	**Happy**	**Sad**
Anger	**0.53**	0.09	0.05	0.31
Fear	0.12	**0.59**	0.06	0.21
Happy	0.02	0.02	**0.85**	0.09
Sad	0.12	0.08	0.08	**0.69**

Stacked LSTM				
	Anger	**Fear**	**Happy**	**Sad**
Anger	**0.62**	0.1	0.04	0.22
Fear	0.15	**0.6**	0.06	0.17
Happy	0.03	0.03	**0.85**	0.64
Sad	016	0.1	0.07	**0.64**

Stacked LSTM w/ Attention				
	Anger	**Fear**	**Happy**	**Sad**
Anger	**0.63**	0.09	0.05	0.23
Fear	0.14	**0.63**	0.06	0.17
Happy	0.03	0.03	**0.84**	0.08
Sad	0.18	0.09	0.08	**0.64**

Single Layer bi-LSTM				
	Anger	**Fear**	**Happy**	**Sad**
Anger	**0.59**	0.1	0.05	0.24
Fear	0.13	**0.63**	0.07	0.16
Happy	0.03	0.03	**0.85**	0.07
Sad	0.15	0.1	0.08	**0.54**

Single Layer bi-LSTM w/ Attention				
	Anger	**Fear**	**Happy**	**Sad**
Anger	**0.62**	0.11	0.06	0.21
Fear	0.15	**0.63**	0.04	0.18
Happy	0.04	0.04	**0.84**	0.08
Sad	0.20	0.11	0.08	**0.61**

Stacked bi-LSTM				
	Anger	**Fear**	**Happy**	**Sad**
Anger	**0.60**	0.09	0.06	0.25
Fear	0.12	**0.60**	0.07	0.20
Happy	0.03	0.04	**0.85**	0.09
Sad	0.14	0.08	0.08	**0.70**

Stacked bi-LSTM w/ Attention				
	Anger	**Fear**	**Happy**	**Sad**
Anger	**0.57**	0.12	0.05	0.26
Fear	0.12	**0.64**	0.06	0.18
Happy	0.04	0.04	**0.84**	0.08
Sad	0.15	0.11	0.08	**0.66**

As the dataset is relatively small, it is also important to check for over- or underfitting of models. To indicate this, plotting training accuracy and validation accuracy along with their loss, for each epoch and noting which models have the lowest difference should indicate which model is the least overfit. These comparisons were only done for the two models with the highest F1 score (two layer bi-LSTM, two layer LSTM w/ attention) (fig. 1 and fig.2) and the two models with the lowest F1 score (single layer LSTM, single layer bi-LSTM) (fig. 3 and fig. 4).

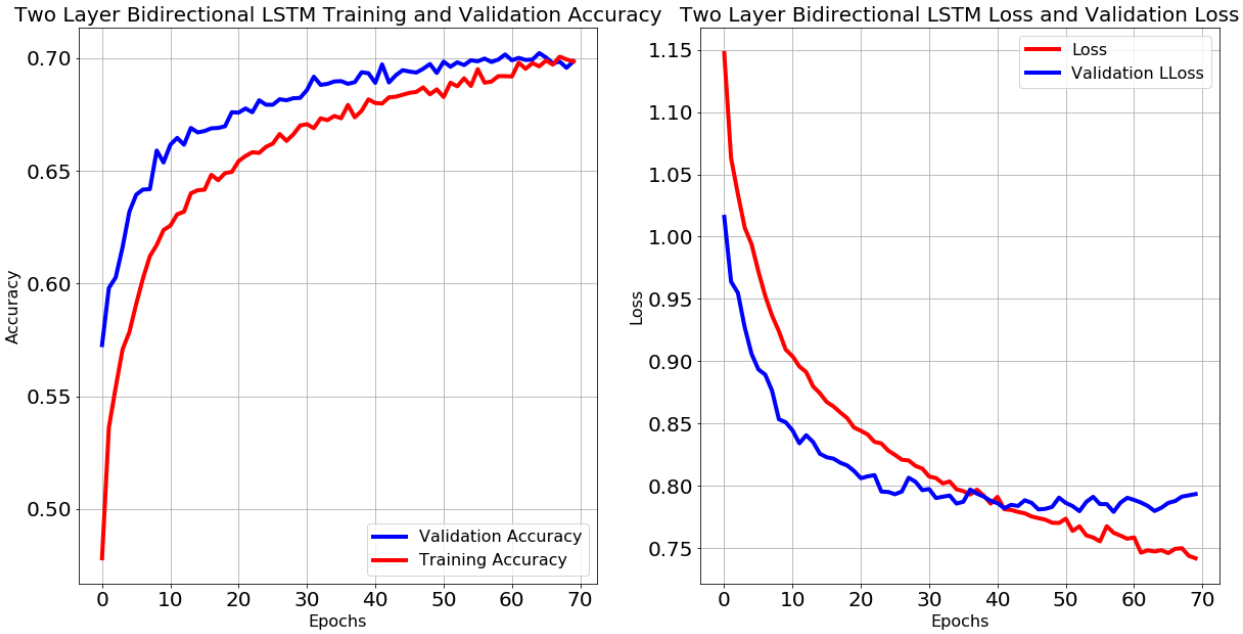

Fig. 2: Two Layer Bidirectional LSTM

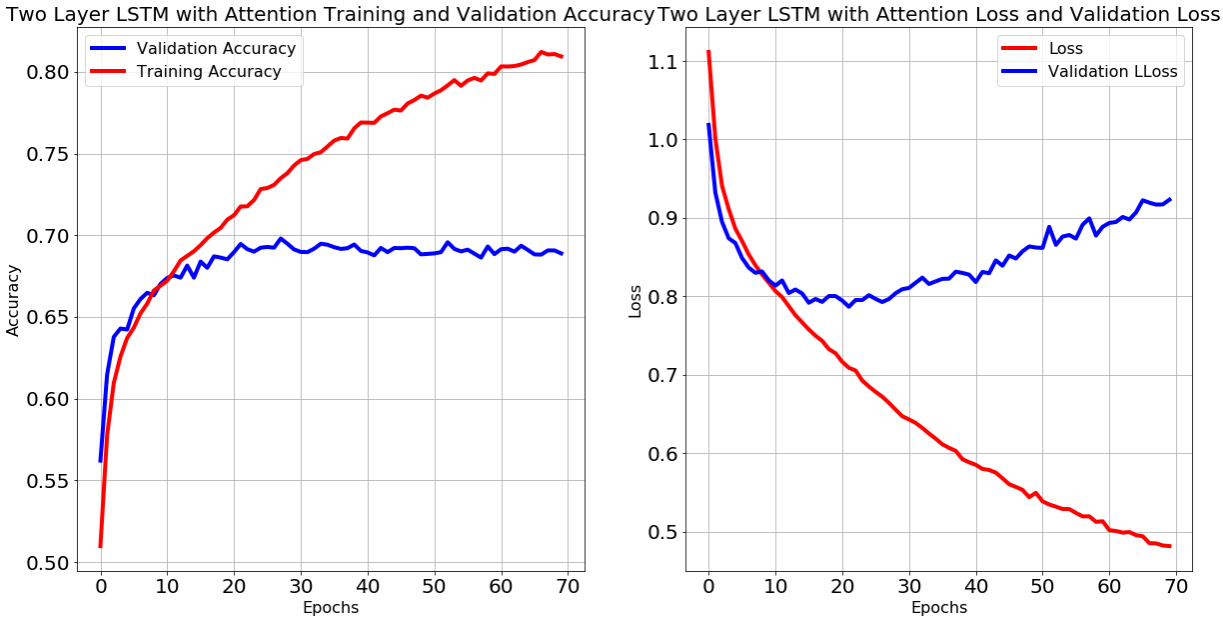

Fig. 3: Two Layer LSTM with Attention

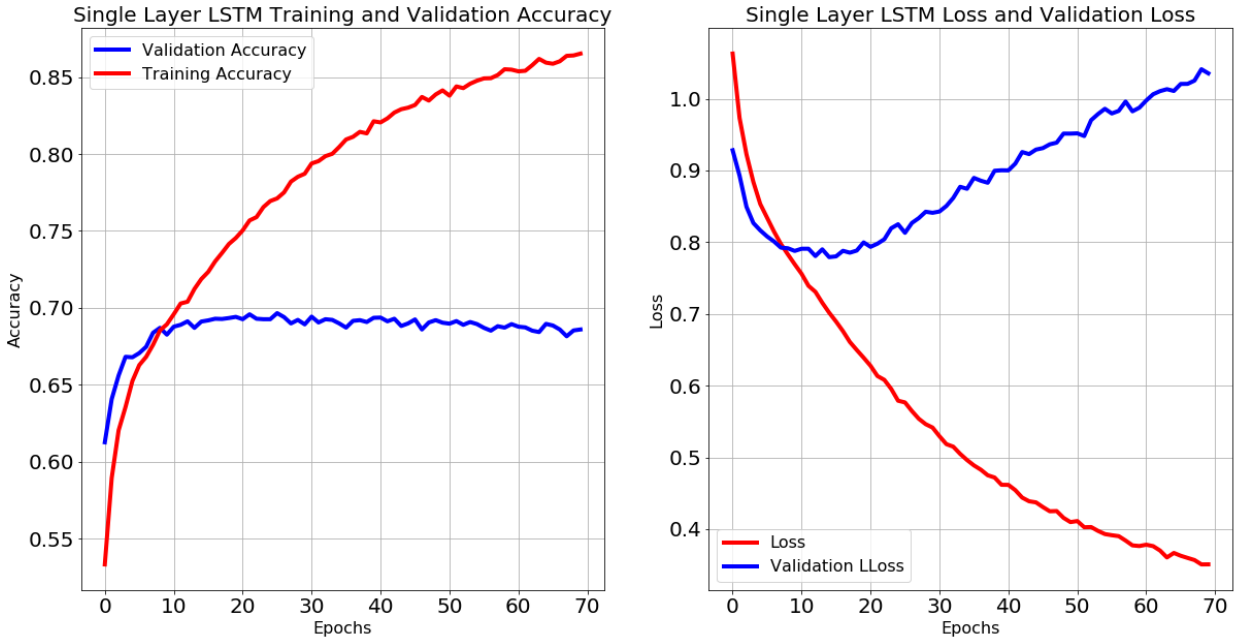

Fig. 4: Single Layer LSTM

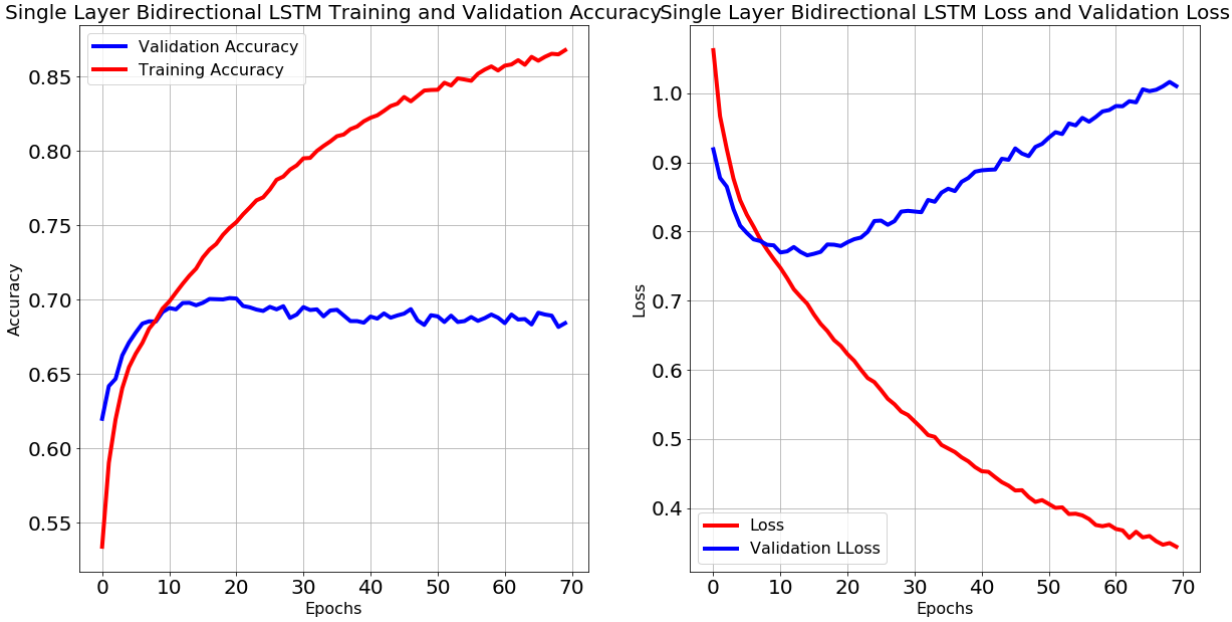

Fig. 5: Single Layer Bidirectional LSTM

3.3. Related Work

_____Twitter has many restrictions surrounding publishing a queried dataset. Additionally, most related research for this topic is concerned with sentiment analysis as opposed to classifying emotional state. Furthermore, even research that is closely related to our has

fundamental differences in the dataset or the models (i.e.: in the extra datasets added after our own queried data, five emotional classes were considered as opposed to four, as the model was intended to identify a 'neutral' emotional state as well). The winning team of similar competitions achieved an F1 score of nearly 0.71 using a much more complex model[17]. Different models trained on a dataset six times larger than our own queried dataset, queried the same way as ours, achieved a maximum F1 score of 0.71, which is only slightly higher than ours[15]. Considering these differences and similarities, it may be worthwhile to ask whether or not more data is beneficial to these models or if a sample size of approximately 10000 tweets per class is sufficient.

3.4. Model Selection and Streaming Results

After considering the performance of each individual model, we decided to pick 2-layer bi-LSTM without attention for streaming, as it had the highest F1 score. Additionally, its training and validation accuracy and loss were very close to each other, implying very little overfitting. To facilitate the model's integration with the streaming pipeline, hyperparameters like number of nodes and dropout rate on each layer were optimized. In order to achieve that, we used k-fold cross validation to fine-tune one hyperparameter at the time by trying different values, while keeping the other parameters unchanged. For each hyperparameter, we picked the value that gave the best F1 score while also preventing the model from overfitting. Below is the architecture of our selected model. Additionally, both bi-LSTM layers have dropout and recurrent dropout set to 0.5, to prevent overfitting. Analyzing the difference between training loss and validation loss (fig. 2) implied that 70 epochs did not reduce loss nor increased accuracy after 40th epoch. Thus, a more suitable 42 epochs was chosen.

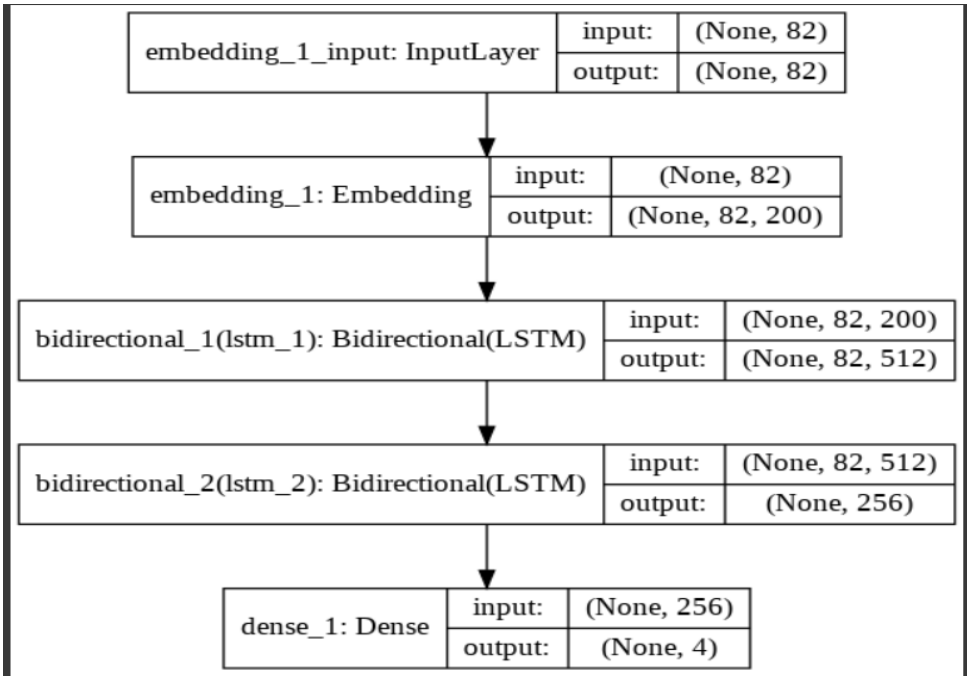

Fig. 6: Architecture of the 'best' model

To finally reach the initially defined goal of classifying public opinion of a topic in terms of emotion through streaming data, a streaming pipeline was created using tweepy. A Tweets Stream Listener object, tracking tweets containing one or more keywords,given by the user, was created. In addition, a preprocessing pipeline which included cleaning new tweets the same way we cleaned our training dataset was defined. Afterwards, they were fed into the model and their emotion was classified into one of the 4 classes. As a proof-of-concept exercise, geographic data was also recorded for integration with a visualizer.

Finally, the tweet's emotion, geolocation and timestamp were saved into an elasticsearch index. The architecture of the streaming pipeline is shown below.

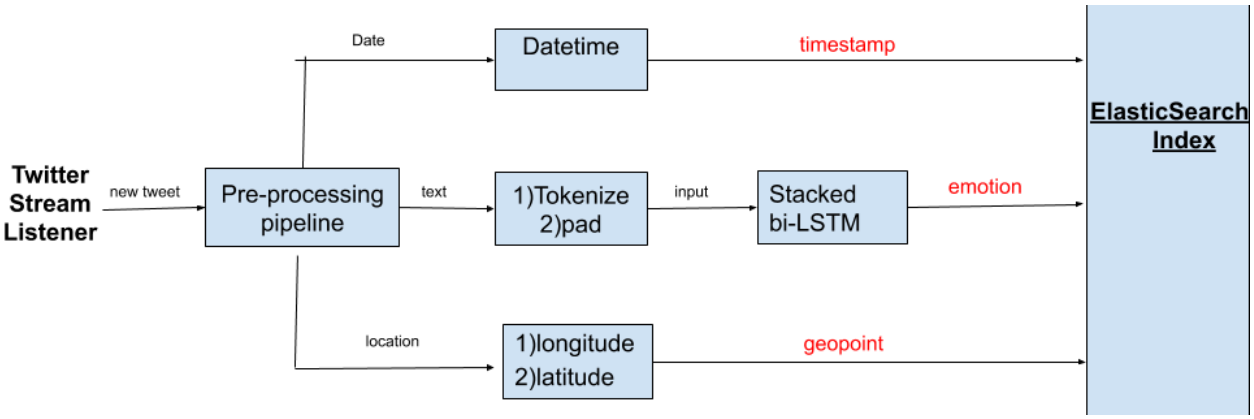

Fig. 7: Architecture of the streaming pipeline

A dashboard was created on Kibana, so we could further analyze the values we have generated from our streaming pipeline. The dashboard contains three different visualizations: a map to compare emotions of different users based on their geolocation and find patterns between countries, a line plot which analyzes how the total number of tweets regarding each emotion varies during time, and a pie chart to compare the total counts for each emotion. To demonstrate our results, we initialized our streaming pipeline and got it to track tweets containing the “#coronavirus” keyword. Streaming data was collected for 8 hours and the model classified over 100000 tweets (fig. 8).

An obvious first observation is that over half of the total tweets were classified as fear. Considering the obvious implication that the 2020 COVID-19 situation is fearful, this makes sense intuitively. Furthermore, a pattern around Europe seems to take place, most (but not all) tweets are either fear or sadness (which again makes sense intuitively. In addition, the regional percentage of happy tweets is very low. However, further analyzing distinct regions within the UK, a different pattern emerges.

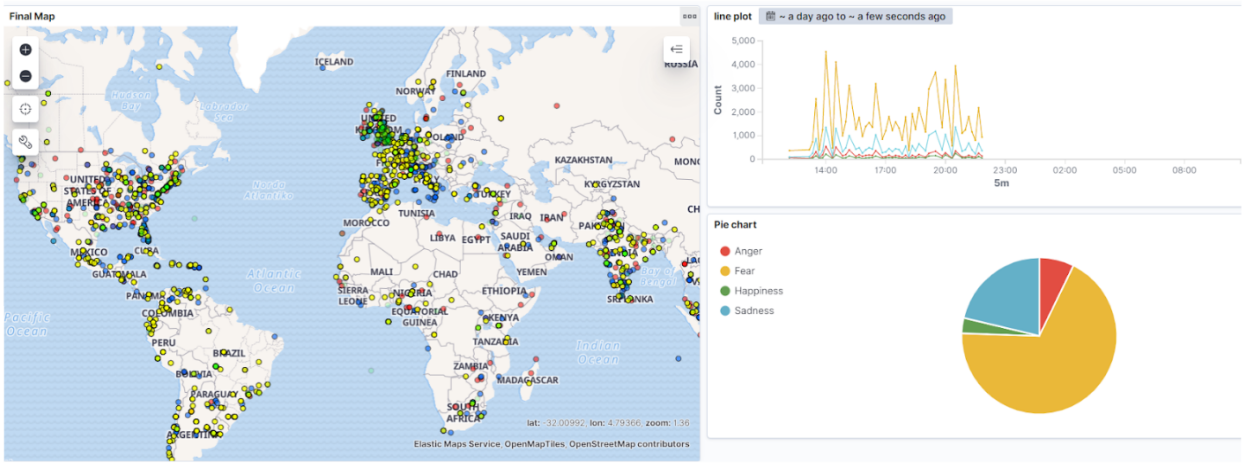

Fig. 8: Kibana visualizations of global streaming data ('#coronavirus')

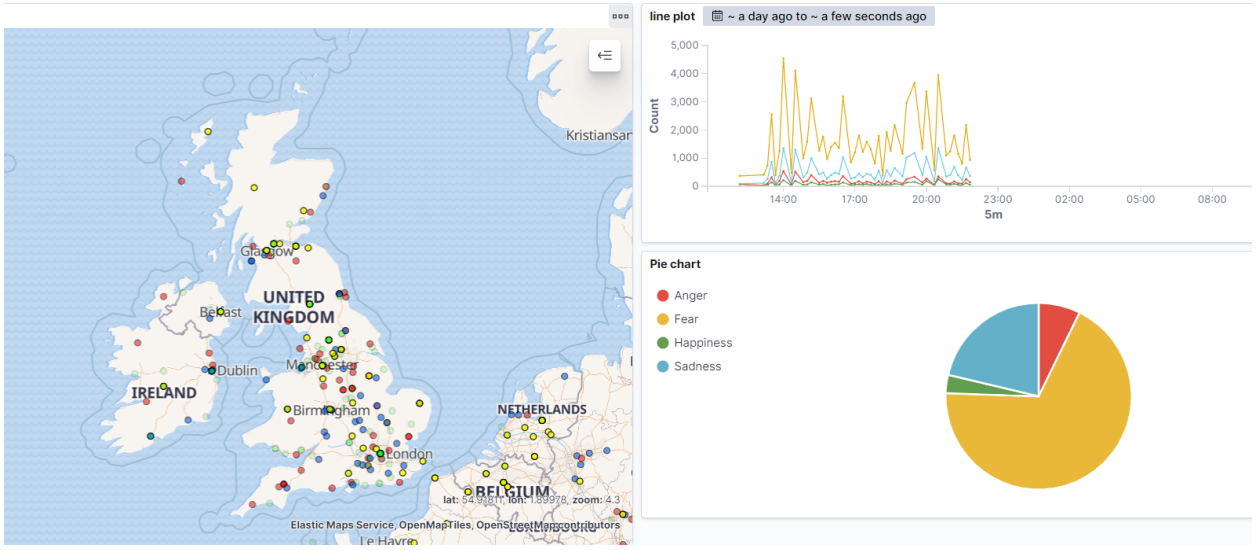

Fig. 9: Kibana visualizations of UK and Ireland streaming data ('#coronavirus')

People's feelings (based on emotional state) vary within the UK, more so than the rest of Europe. The percentage of angry tweets is much higher than the rest of Europe. This could be a result of the public perception that measures taken to prevent the spread of COVID-19 were ineffective or late, or in general because of the public behaviour concerning the spread of the virus.

Our Kibana dashboard allows us to focus on specific regions or specific ranges of time. For example, if an optimistic announcement was made, we create separate visualizations for tweets before and after the announcement, and observe if the percentage of happy tweets is increasing, which could imply that the announcement was effective. In general, many different queries could be made, and many patterns could be discovered by analysts - again, this work serves as a proof of concept, and there is much still to be explored.

4. Discussion:

Comparing our model against others that have been previously researched demonstrated promising results. However, checking our confusion matrix led to some interesting conclusions that muddy the waters of this research a little further. Manually looking at the dataset and the automatically classified labels (chosen in the data querying part, not in the machine learning part) led to the notion that some of these tweets may have been initially classified incorrectly. These were mostly within the negative emotion categories. It was quite common for a tweet that appears intuitively angry to be classified as sadness. Additionally, many tweets could have belonged to more than one category (ie: a tweet can imply both anger and sadness). Our model only accounted for one of the four class labels for each tweet. Nevertheless, our model was able to correctly classify happy tweets with 87% accuracy. Some misclassifications might have been the result of sarcasm in tweets, users straying from the defined boundaries (ie: a sad tweet may have the hashtag of '#joy', there could be many reasons for this), and tweets created through bots of advertisers not being filtered.

The results imply that the trained models can successfully classify 'happy' tweets. However, the models fail to consistently classify the 'negative' classes ('anger', 'fear', 'sadness'). This might be explained by the implication that 'happy' is distinct from 'anger', 'fear', and 'sadness' due to happiness being quantified as a positive emotion and the other three are quantified as a negative emotion. Therefore, the negative classes have similarities with each other while happiness does not have similarities with any other class.

An unexpected part of the results was the idea that tweets could be classified as more than one emotion. Currently, our model only predicts a tweet as belonging to one of four emotional states, but it is possible someone might feel both sad and angry about something. See future work for further analysis on how this can be resolved.

With all of this in mind, the performance of the model is still better than expected. The model may be further improved through more data for each class (ie: reaching our initial goal of approximately 15000 tweets per class, most notably 'fear' and 'anger') as the approach on previous research[15] has proven.

As mentioned previously, analyzing the dataset uncovered a fairly high error rate in the predicted label (determined in the data querying step). One of the major problems in natural language processing is the difficulty in determining what emotion a user might be feeling. Obviously, language is quite complex and there are always exceptions to the rules. In looking at how our model classified some of the ambiguously classified tweets, we were actually quite surprised - the model was often able to predict the emotional class that intuition might suggest a tweet belongs to, despite the tweet having been classified in the data querying step as something else. Below are a few examples of ambiguous classifications.

Tweet	True label (Querying)	Predicted Label (Model)
with only <number> months left until i possess my undergraduate degree i feel like i can not handle adulthood anymore no jobs in biology	**Anger**	**Sadness**
every night i try and take a nap before work and i get started on thinking about wedding stuff which means no nap for me weary_face	**Fear**	**Sadness**
this is getting really real now flushed_face scary university <url>	**Happy**	**Fear**
<user> - please remember all of the bigots and religious ignorant that have filled the world with and hate	**Fear**	**Anger**
<user> i fear its almost too late to turn it around for our neighbours to the south	**Sadness**	**Fear**
<user> i miss you too	**Anger**	**Sadness**
fuck man another great one bites the dust . rip patrice oneal . <user>	**Sadness**	**Anger**

_____An additional challenge with training these models was the availability of data. As mentioned before, the training dataset was only 50000 tweets taken from one week's time. Due to the time frame, it's possible that the dataset may have been biased. For example, trending topics that week might have been more positive than they are normally, or people were more likely to use hashtags such as 'anger' more than previous weeks. Given the time constraints, this was impossible to investigate. Additionally, each model was trained over 70 epochs with a batch size of 256, and trained using Google Colab in order to reduce training time. Despite this, training each model still took nearly an hour or more, and would have taken orders of magnitude longer if k-fold validation was used. Intuition implies that scaling a good model to a training dataset of millions of datapoints, increasing the batch size to a much larger amount (such as 2000 or so) would lead to a more accurate model.

5. Future Work:

There were many variables that could have been different throughout the project, thus opening up the opportunity for future research.

First and foremost, the size of the dataset is smaller than we would have liked. Given more time or access to more tweets, a much larger set could have been compiled. There are approximately 5000 words used commonly in the English language, and the average English speaker knows approximately 42000 words[18]. Comparing this information to the size of our dataset makes it seem insufficient. Nevertheless, the best-performing model (two layer bi-LSTM) was about to achieve an accuracy of approximately 70%, which far exceeds our early estimates. Additionally, the model was able to achieve results on streaming data. Despite not analyzing these results in-depth, this implies that given more time and less hardware constraints, this system could be trained on more data over more time, integrated with the streaming pipeline, and produce a reasonable visualization of public opinion modelling.

Different choices could have been made in cleaning the dataset. There were many discussions surrounding how to clean, stem, and tokenize tweets. For example, we made the decision not to remove stop words as intuition implies that they do not actually affect the meaning of a phrase when it is stemmed and tokenized. However, this could be false when it comes to Twitter. Additionally, there wasn't a strong effort made to remove tweets that were likely generated by bots or tweets advertising a product or service.

There are countless neural networks that can be applied to this problem. Due to hardware limitations and time constraints, it took nearly 10 hours to train each model, meaning that only a handful of these models could be tested and parameters couldn't necessarily be fine-tuned. The choice of models here exists mostly as a proof-of-concept. It is possible that another model trained on a larger dataset with different parameters could give much more promising results. As previously discussed, it may also be prudent to reframe this as several binary classification problems as opposed to a single multiclass classification problem.

Visualization according to geolocation was an idea floated at the start of the project. Utilizing ELK to not only stream tweets, but also utilizing Kibana to visualize the geolocation of users (and their corresponding emotional states) could be useful for determining regional differences. Correctly identifying public opinion about topics based on location can inform analysts as to future decisions, for example, if one American state has a mostly happy feeling towards Trump (the current president of the United States), a political ad campaign may elect to concentrate their efforts on states that may have feelings of fear or anger towards Trump. This was explored briefly after the streaming pipeline was implemented and initial results were quite promising. Exploring this aspect of the project would be a natural extension of the research.

Tweets belonging to more than one (or less that one, ie: neutral) emotional class should be considered. A single multiclass model may be able to solve this problem by simply adjusting

the output nodes as having a threshold for each class. See below for an example of how this might classify new tweets. Bear in mind this is only one proposed architecture - there may be hundreds of different ways of solving this problem.

Tweet	**Happy**	**Sad**	**Fear**	**Anger**
with only <number> months left until i possess my undergraduate degree i feel like i can not handle adulthood anymore no jobs in biology	**No**	**Yes**	**Yes**	**No**
every night i try and take a nap before work and i get started on thinking about wedding stuff which means no nap for me weary_face	**Yes**	**Yes**	**No**	**No**
this is getting really real now flushed_face scary university <url>	**No**	**Yes**	**Yes**	**Yes**
<user> - please remember all of the bigots and religious ignorant that have filled the world with and hate	**No**	**Yes**	**No**	**Yes**
<user> i fear its almost too late to turn it around for our neighbours to the south	**No**	**No**	**Yes**	**No**
<user> i miss you too	**No**	**Yes**	**No**	**No**
<user> i suppose **(this tweet would be classified as neutral as it does not meet the criteria for any other emotion)**	**No**	**No**	**No**	**No**

Another key improvement that could have been made throughout this project is the implementation of k-fold validation when training the models. This would have ideally increased the accuracy of each model. However, when using k-fold validation, even using a small choice of k (for example, 5) dramatically increases the computation time. Even through the use of Google Colab, these models would have taken an unrealistic amount of time to train. Given more processing power or time (or both), k-fold validation would have likely led to better results for all models. Nevertheless, these models and their results exist primarily as a proof of concept, and with validation accuracy hovering just below 70% for all models, the initial results are promising.

Increasing the size of the dataset, number of epochs, batch size, or other model parameters may have likely led to better results as well. Given that our dataset consisted of approximately 50000 tweets and each model was trained on the same training data, and validated on the same validation set, it is possible that our models had a tendency to overfit. As mentioned earlier, the average English speaker knows approximately 42000 words, and these words can be placed together in a massive number of different ways to create different meaning or indicate different emotions. Gathering a dataset orders of magnitude larger, despite the increase in training time, has the potential to achieve a higher accuracy.

Different choices could have also been made in the choice of emotions to use as class labels. One of the key issues was most models strength in correctly classifying happy tweets, but greater difficulty in classifying the other three classes. Of these four classes, happy is the only positive emotion, while the other three are negative. Expanding the number of classes to six (three positive emotions, three negative emotions) or including some kind of integration with sentiment analysis could also give more descriptive results.

References:

[1] Choudhury, Munmun D., Counts, Scott (2012) *Not All Moods are Created Equal! Exploring Human Emotional States in Social Media.* [WWW Document] Retrieved March 5, 2020, from https://www.researchgate.net/publication/255564129_Not_All_Moods_are_Created_Equal_Exploring_Human_Emotional_States_in_Social_Media

[2] Mohammed, Saif M. (2012) *#Emotional Tweets*. [WWW Document] Retrieved March 5, 2020, from https://www.aclweb.org/anthology/S12-1033.pdf

[3] Shaver, Philip (1987) *Emotion Knowledge: Further Exploration of a Prototype Approach.* Journal of Personality and Social Psychology

[4] Baziotis, Christos, Athanasiou, Nikos, Chronopoulou, Alexandra, Kolovou, Athanasia, Paraskevopoulos, Georgios, Ellinas, Nikolaos, Narayanan, Shrikanth, Potamianos, Alexandros (2018) *NTUA-SLP at SemEval-2018 Task 1: Predicting Affective Content inTweets with Deep Attentive RNNs and Transfer Learning*. Retrieved March 5, 2020, from https://www.aclweb.org/anthology/S18-1037.pdf

[5] *Wang, W., Chen, L., Thirunarayan, K., Sheth, A.*, (2012). *Harnessing Twitter "Big Data" for Automatic Emotion Identification.* Retrieved March 6, 2020, from https://www.researchgate.net/publication/258762213_Harnessing_Twitter_'Big_Data'_for_Automatic_Emotion_Identification

[6] Mudinas, Andrius & Zhang, Dell & Levene, Mark. (2012). *Combining lexicon and learning based approaches for concept-level sentiment analysis.* 10.1145/2346676.2346681.

[7] Elvis (2018). *Deep Learning for NLP: An Overview of Recent Trends* [WWW Document]. Medium. URL https://medium.com/dair-ai/deep-learning-for-nlp-an-overview-of-recent-trends-d0d8f40a776d (accessed 3.5.20).

[8] Mittal, A., (2019). *Understanding RNN and LSTM* [WWW Document]. Medium. Retrieved March 6, 2020, from https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e

[9] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., (2019). *Language Models are Unsupervised Multitask Learners.*

[10] Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., Lehmann, S., (2017). *Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm.* EMNLP. Retrieved March 6, 2020, from https://doi.org/10.18653/v1/D17-1169

[11] Baziotis, C., Pelekis, N., Doulkeridis, C., 2017. DataStories at SemEval-2017 *Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis*, in: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Presented at the SemEval 2017, Association for Computational Linguistics, Vancouver, Canada, pp. 747–754. https://doi.org/10.18653/v1/S17-2126

[12] Jeffrey Pennington, Richard Socher, Christopher D. Manning (2014) *GloVe: Global Vectors for Word Representation.* https://nlp.stanford.edu/projects/glove/

[13] Kingma, D.P., Ba, J., 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs].

[14] Bi-LSTM - Raghav Aggarwal - Medium [WWW Document], n.d. URL https://medium.com/@raghavaggarwal0089/bi-lstm-bc3d68da8bd0 (accessed 3.12.20).

[15] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E., 2016. Hierarchical Attention Networks for Document Classification, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Presented at the Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, pp. 1480–1489. https://doi.org/10.18653/v1/N16-1174

[16] Colneric, N., Demsar, J., 2019. Emotion Recognition on Twitter: Comparative Study and Training a Unison Model. IEEE Trans. Affective Comput. 1–1. https://doi.org/10.1109/TAFFC.2018.2807817

[17] Mohammad, S., Bravo-Marquez, F., Salameh, M., Kiritchenko, S., 2018. SemEval-2018 Task 1: Affect in Tweets, in: Proceedings of The 12th International Workshop on Semantic Evaluation. Presented at the Proceedings of The 12th International Workshop on Semantic Evaluation, Association for Computational Linguistics, New Orleans, Louisiana, pp. 1–17. https://doi.org/10.18653/v1/S18-1001

[18] Marc Brysbaert, Michaël Stevens, Paweł Mandera, Emmanuel Keuleers. How Many Words Do We Know? Practical Estimates of Vocabulary Size Dependent on Word Definition, the

Degree of Language Input and the Participant's Age. Frontiers in Psychology, 2016; 7 DOI: 10.3389/fpsyg.2016.01116

[19] Charles Malafosse, 2019, FastText-sentiment-analysis-for-tweets , GitHub repository, https://github.com/charlesmalafosse/FastText-sentiment-analysis-for-tweets/blob/master/betsentiment_sentiment_analysis_fasttext.py

[20] Kyokomi, 2019, Emoji, GitHub repository , https://github.com/kyokomi/emoji

[21] Saif M. Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, Svetlana Kiritchenko. Semeval-2018 Task 1: Affect in Tweets.. *Proceedings of the International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA, June 2018.

[22] Saif M. Mohammad and Felipe Bravo-Marque,2017, WASSA-2017 Shared Task on Emotion Intensity. z. In Proceedings of the EMNLP Workshop on Computational Approaches to Subjectivity, Sentiment, and Social Media (WASSA), September 2017, Copenhagen, Denmark.

Appendix:

A. List of emotion synonyms

Emotion	Synonyms Used as Hashtag
Joy	#happy , #joy , #liking , #lovin , #loving , #happiness , #joy , #happy , #glad , #delight , #cheerful , #thrill , #exciting , #thrilled , #delightful,#happy , #joy , #liking , #happiness , #joy , #glad , #delight , #cheerful , #thrill , #exciting , #thrilled , #delightful , #cheerfulness , #jolliness , #enjoyment , #euphoria , #jubilation , #joyful , #jubilant , #enthusiasm , #excitement , #excited , #proud , #pride , #optimism , #relief , #adoration , #ease
Sadness	#sadness, #sad, #depressed, #depressing, #desperation, #unhappy, #brokenhearted, #heartbroken, #heartbreak, #pain, #suffer, #misery, #grief, #suffering, #regrets, #disappointment, #disappoint, #disappointed, #disappointing, #letdown, #loneliness, #homesick, #lonely, #lonesome, #regretting, #despair, #sorrow, #agony, #depression, #hopelessness, #unhappiness, #despairing, #cheerless, #dejected, #hurt,, #friendless, #embarrassment, #embarrass, #embarrassed, #embarrassing, #abashment, #awkwardness, #ashamed, #remorseful, #guilt, #guilty, #regretful
Anger	#mad, #hate, #irritation, #annoyance, #irritating, #irritated, #annoying, #annoyed, #frustrated, #disturbing, #anger, #rage, #outrage, #frustrating, #frustration, #angry, #disgusted, #offended, #disgust, #enraged, #outraged, #furious, #infuriated, #bothersome, #irksome, #fury, #wrath, #frenzy, #irate, #ireful, #offended, #raging, #wrathful, #disgust, #disgusting, #disgusted, #frustration, #frustrated, #frustrating, #frustrate, #envy, #jealousy, #jealous #envying
Fear	#fright, #terror, #scaring, #panic, #scared, #frightened, #fearful, #panicking, #panicked, #panicky, #anxious, #nervousness, #tenseness, #tension, #uneasiness, #worry, #anxious, #nervous, #tense, #uneasy, #worried, #worrying