CS224U Final Project: COVID-19 Twitter Sentiment Analysis

Natasha Ong
Stanford University
natashao@stanford.edu

Bradford Lin
Stanford University
blin1201@stanford.edu

Nattawat Luxsuwong
Stanford University
nashlux@stanford.edu

Abstract

In this study, we propose a new COVID Twitter sentiment dataset, perform exploratory data analysis (EDA) on our dataset, and conduct sentiment analysis on Tweets during the COVID pandemic to analyze how key events impacted user sentiment. Additionally, we train a number of different NLU models, including LSTM and CNN, on GloVe and Word2Vec to evaluate how word embeddings impact model performance, using TextBlob as ground truth. Lastly, we also compare our results against BERT, a state-of-the-art transformer-based model. Overall, we find that most Tweets during the pandemic are neutral to slightly positive and skew sharply after milestone events, depending on how Twitter users react to said event. Moreover, although all of the models we train from scratch underperform BERT, we find that our LSTM and CNN models perform significantly better, in terms of accuracy and F1 score, with word embeddings, and that GloVe outperforms Word2Vec.

1 Introduction

Throughout 2020 and the first half of 2021, a large portion of conversation on the internet–and Twitter in particular–has revolved around COVID and its occurrences. As a result, we have the unique opportunity today to use Tweet data from Twitter to conduct sentiment analysis throughout the pandemic.

We perform two experiments, one more social-sciencey and another more technical. For our first experiment, we investigate how certain milestones during the COVID pandemic caused shifts in Twitter sentiment (AJMC, 2021). Our central hypothesis for this experiment is that positive events, such as vaccine development, will lead to a short-term upward shift in sentiment. Likewise, negative events, such as record case/death counts, would lead to a short-term downward shift in sentiment. Moreover, we expect the average Tweet to have a sentiment score that is neutral, because neutral is the baseline state (and polarized Tweets will likely average to neutral). Quantitatively, the amount of discourse and reaction to certain events will likely impact the spikes and amplitudes of the spikes, though the appearance of spikes depends on the differences in sentiment from the surrounding days. To aid our social science investigation of how COVID affected Twitter sentiment, we also conduct exploratory data analysis (EDA) on our dataset to look for specific clusters and topics of interest, as well as words that are most frequently used in Tweets.

For our second experiment, we investigate how much of a boost the use of word embeddings, specifically Word2Vec and Glove, have on our model performance of classifying Tweet sentiment. We treat the sentiment scores obtained from TextBlob as ground truth and train LSTM and CNN models with these word embeddings. We hypothesize that models trained with word embeddings will perform better than models with no pre-trained embeddings since word embeddings help capture the semantic and syntactic meaning of a word as they are trained on large datasets. Furthermore, we hypothesis that Word2Vec models will perform better than GloVe models, given the way Word2Vec computes word embeddings by taking text as training data for a neural network, whereas GloVe simply computes the co-occurences of words over large corpora. This experiment will ultimately help us investigate how much more benefit word embeddings provide for overall performance as well as which embeddings may be useful in which cases.

We also explore BERT, a state-of-the-art transformer-based Language Model (LM), to better compare the performance on our dataset. Intuitively, we expect BERT to do the best as it has

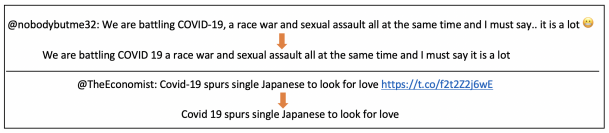

Figure 1: Data Pre-processing: Sample Tweets before and after pre-processing

greater contextual understanding: It jointly conditions on both left and right context for all layers, while LSTM and CNN models only combine non-contextual word representations. Training and testing our dataset on BERT may help us evaluate the current upper-bound performances on our dataset.

2 Related Work

Since the onset of the pandemic, there have been a number of studies that have conducted sentiment analysis on COVID-related Tweet data. To begin, we can consider a few studies that compared Twitter sentiment before and after the onset of the pandemic. Nemes & Kiss train a recurrent neural network (RNN) and find that most people's Tweets during COVID are mostly neutral or weakly biased in one direction (positive or negative), with positivity actually having increased during COVID (Nemes and Kiss, 2021). They also compare their RNN results against more traditional sentiment analysis methods, such as human polling, only to note that old-fashioned methods quickly become outdated in the fast-paced nature of a pandemic. Conversely, Feng & Zhou (2020) run a similar analysis and find that the predominant sentiment is negative during COVID (Feng and Zhou, 2020). Manguri et al. (2020) conduct a similar as Nemes & Kiss (2021) and obtain similar results as them, although their scoring metric for sentiment was more nuanced and codified additional emotions such as fear and joy (Manguri K. H. and R., 2020).

We should note that quite a few studies used different models from each other and obtained different results. These different results are likely due to a combination of different data and models. For example, Xue et al. (2020) compute sentiment using Plutchik's wheel (Plutchik and Kellerman, 2013) and find that the predominant sentiment during the COVID pandemic was fear. Xue et al. (2020) also use LDA to identify patterns and themes of the various Tweets during the pandemic, an analysis we also conduct during our exploratory data analysis (Xue, 2020).

Other studies focused on more niche areas of sentiment analysis during COVID, such as considering the specific periods of lockdown and work from home. Naseem et al. (2021) trained a multitude of different models, ultimately finding that BERT performed the best. Their results suggest that by mid-March of 2020, when lockdown was in full swing, negative sentiment increased compared to the pre-lockdown period (Naseem, 2021).

Beyond looking at milestone events, existing literature also compares sentiments across different physical locations. Chintalapudi et al. (2021), Manguri et al. (2020), and Imran et al. (2020) all focused on different geolocations (Chintalapudi, 2021) (Manguri K. H. and R., 2020) (Imran A. S. and R., 2020). Chintalapudi et al. (2021) focused only on Tweets from India, Imran et al. focused only on Tweets from Pakistan, India, Norway, Sweden, USA, and Canada, and Manguri et al. (2020) did not target a particular continent, country, or city. This may explain the stark contrast in sentiment distributions found by Chintalapudi et al., Manguri et al, and Imran et al.. Both Imran et. al and Chintalapudi et al. agreed that LSTM models without any pre-trained embeddings performed relatively poorly compared to other models.

Chen et al. (2020) seems to be the only paper of those we reviewed that looks at topic preference related to the use of controversial and non-controversial terms associated with COVID on Twitter. Specifically, they found that there is low interchangeability between the two groups of terms (Chen, 2020).

We contribute to the existing literature by analyzing COVID's most recent timeframe, whereas most of the existing literature conducted sentiment analysis on Tweets from the early pandemic. Additionally, we consider the effect of different word embeddings on the performance of a number of sentiment analysis models, while conducting a comprehensive exploratory data analysis.

3 Data

We construct a new dataset of 858,000 Tweets about COVID between mid-March 2020 and April 2021, with 2000 Tweets each day during the mentioned period. Each Tweet is provided along with its corresponding date of creation (i.e. when the Tweet was posted) and sentiment score. We do not include other information like screen name, location, name, and URL as these features are not relevant to our particular experiments.

Apr 05 2020	okay I m bored so would you rather the hunger games existing or corona	0
April 05 2020	Absolutely COVID 19 can be overcome with teamwork and taking the right precautions Let us keep following social distance	0.4
Apr 05 2020	Whoa Borriss Joohnnsonn has corona	-0.426
Apr 05 2020	Coronavirus Twins born during India lockdown named Corona and Covid	0.8
Jun 05 2020	In one blundering step the feds interfere with constitutional rights amp with public health Free the masks BlackLivesMatter	-0.4
Jun 05 2020	Black Lives Matter spent tens of thousands of dollars to send covid masks to thousands of protesters across the country	0.1653
Jun 05 2020	We re happy that the WHO resumed trials of hydroxychloroquine I firmly believe that the WHO s decision was taken in haste It was	0.4

Figure 2: Dataset: subset taken from June 05 2020 and April 05 2020

We use Tweepy (Roesslein, 2020) to crawl English Tweets and specifically focus on important and relevant keywords/hashtags. We included the following keywords and their respective hashtags:

- corona
- coronavirus
- covid
- covid-19
- pandemic
- quarantine
- ventilators
- kung flu
- vaccine
- covid vaccine
- corona vaccine
- hand sanitizer
- social distancing
- lockdown
- stay home
- health workers
- chinese virus
- work from home

Each Tweet is then pre-processed by the removal of URLs, stop words, emojis, hashtags (just the symbol not the content of the hashtag), and usernames (ie @username). Figure 1 presents some sample Tweets before and after our pre-processing steps.

For sentiment score, we pass these pre-processed Tweets through TextBlob (Loria, 2018), a third-party library provided in Python to estimate the sentimental polarity of words and Tweets by calculating the score as a polarity [-1 to 1].

Figure 2. includes a sample subset of our data.

4 Models

All classifiers developed to investigate word-embeddings will be based on Long Short-Term Memory (LSTM) Networks and Convolution Neural Networks (CNN). We explore BERT to better compare performance to state-of-the-art models.

4.1 Long Short Term Memory (LSTM) Network

LSTMs are artificial recurrent neural network architectures that have feedback connections. LSTM models process current input while also retaining previous state (which are the outputs from previous inputs), allowing us to process longer sequences. The capability of LSTMs to retain previous states are what enable it to understand word context, allowing it to outperform simpler architectures such as single layered Deep Neural Networks at processing long sequences.

4.2 Convolution Neural Network (CNN)

CNNs are a class of deep neural networks that are commonly utilized for image processing. However, recent studies have shown that CNNs may show promise on sequence processing as well. This paper will build on these studies in exploring how CNNs perform in classifying COVID Tweet sentiments. CNNs rely on two major operations: convolution and pooling. The convolution operation is carried out on input texts with various filters in order to produce a feature map used for performing classification. The pooling operation uses a ”sliding” filter over each channel of the feature map in order to summarize features in sub-regions of the text.

4.3 Pretrained Embeddings

In addition, this paper will explore how utilizing pretrained word embeddings affects model performance in processing text to achieve sentiment polarity extraction. We investigate **GloVe**, which contains 400,000 word vectors, and **Google's Word2Vec**, which contains 200,000 word vectors.

4.4 Bidirectional Encoder Representations from Transformers (BERT)

To better benchmark the performance of different models on our dataset, we use BERT (Devlin et al., 2018), a state-of-the-art pre-trained masked-language model that uses deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context for all layers. We implement the bert-base-uncased version of BERT using the HuggingFace documentation (Wolf et al.,

2020). We use encode_plus, which adds special tokens, pads the sentence to maximum length provided, and provides attention mask. We also use cross entropy loss and softmax, with a default hidden size of 768. Moreover, we use Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$), lr = 1e-4, 2 epochs, dropout of 0.4 (to reduce overfitting), and a batch size of 64. Our batch size is heavily influenced by the memory/space of our AWS instance.

5 Experiments

5.1 Experiment 1: How Average Sentiment Scores Change Throughout COVID

For this experiment, we do not use particular models as our investigation takes a more data-analysis and social-science approach. Specifically, we use the date of creation and assigned sentiment value from our dataset to look at how sentiment changes through time, especially around key COVID events. For this experiment, we do not particularly care about the Tweet content themselves. We visualize the average sentiment values for each day in our dataset and annotate this graph with milestones or key events. In particular, we source our milestones from 2020 and 2021 COVID timelines provided by NBC News (Muccari et al., 2021) and the American Journal of Managed Care (Staff, 2021).

5.2 Experiment 2: Performance of LSTM, CNN Models using Different Embeddings

For our second investigation, we use the actual Tweets (content of Tweet) and assigned sentiment scores as training data for a standard supervised training problem.

5.2.1 Data Processing

Our preliminary dataset contained the polarity of a Tweet, calculated from TextBlob. For the purposes of our analysis, the following ranges were used to define the sentimental polarity of a Tweet:

$$S_{Tweet} = \begin{cases} \text{Negative} & P < -0.2 \\ \text{Neutral} & -0.2 \leq P \leq 0.2 \\ \text{Positive} & P > 0.2 \end{cases}$$

where S is the corresponding sentiment for a given Tweet with polarity P.

5.2.2 Metrics

In addition, we will evaluate the performance of our models using the metrics *Overall Accuracy* as well as the *macroaveraged F1-Score*. We calculate overall accuracy as:

Figure 3: Visualization of the 7 Topic Clusters

$$\text{Accuracy} = \frac{\text{TP} + \text{TN} + \text{TNEU}}{\text{TP} + \text{FP} + \text{TN} + \text{FN} + \text{TNEU} + \text{FNEU}}$$

Here, TP = True Positive, FP = False Positive, TN = True Negative, FN = False Negative, TNEU = True Neutral, and FNEU = False Neutral. To calculate macro-averaged F1-score, we calculate the precision and recall for each class, calculate the macro-averaged (averaged) precision and recall, and calculate the F1-score using these macro-averaged values. The general precision, recall, and F1-scores have formulas given by:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{accuracy} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision+ recall}}$$

Here, unlike above, TP, FP, TN, FN represent the true positive (TP), false positive (FP), true negative (TN), and false negative (FN) for the relevant class.

In our analysis, we utilize accuracy as it is a simple and widely utilized way to measure our model performance as it informs us the degree to which our model is able to make correct predictions. However, we recognize that accuracy by itself is not a trustful way to evaluate model. Due to the uneven class distribution in our data set, we also look at the macro-averaged F1-score which gives equal weight to each class.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 200, 100)          3595300
_________________________________________________________________
bidirectional (Bidirectional (None, 200)               160800
_________________________________________________________________
dropout (Dropout)            (None, 200)               0
_________________________________________________________________
dense (Dense)                (None, 100)               20100
_________________________________________________________________
dropout_1 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 303
=================================================================
Total params: 3,776,503
Trainable params: 181,203
Non-trainable params: 3,595,300
_________________________________________________________________
```

Figure 4: Summary of Model 2 for Sentiment Polarity Classification

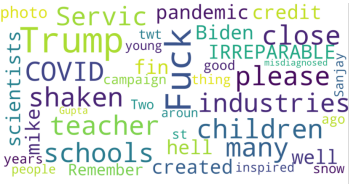

Figure 5: Sample Word Cloud (topic 5)

5.2.3 Methods and Results

To test our hypothesis, we performed 30 trials for each of our models and saved the evaluation metrics for each trial. In each trial, we:

1. Took a random sample of 20,000 tweets from the overall data set.
2. Used the Tokenizer class provided by keras to vectorize and convert the tweets into Sequences so that it can be accepted as input by the network. Sequences were padded/truncated to fit our maximum length of 100.
3. Split the sample into a training and test set with a 70-30 split.
4. Fit the model on the training set and evaluated it on the test set.
5. Saved the accuracy and F1 scores obtained on the test set.

All our models used softmax as the activation function and categorical cross entropy as the loss function. A summary of the LSTM + GloVe model is shown in Figure 4. In addition, a summary of our findings are shown in Table 2. To further test our hypothesis, we run a 1-way ANOVA test on the accuracy and F-1 scores collected in the 30 trials. The ANOVA results are shown in Figure 8 and Figure 9.

6 Analysis

6.1 Exploratory Data Analysis (EDA)

In this section, we conduct exploratory data analysis (EDA) to obtain a more comprehensive view of our custom dataset.

6.1.1 Top Words

We develop a list of the 15 top words used across the dataset, giving us a glimpse into the core vocabulary of the data. Visualization is provided in Figure 6.

6.1.2 Topic Modelling

Topic modelling is a powerful tool in helping identify latent text patterns. In particular, we use LDA (Blei et al., 2003), one of the most popular unsupervised algorithms. LDA works by generating automatic summaries of topics using a discrete probability distribution over words for each topic and explicitly assumes that every word is generated from one of the topics (Ramage et al., 2009). For implementation, we use sklearn's LDA module (Pedregosa et al., 2011).

To reduce the overlap of extracted topics, we set the number of topics as seven. The topics and their most frequent words are shown in Table **??**. Moreover a sample word cloud for topic 5 is shown in Figure 5. This topic seems to be centered around education during COVID.

To better understand the LDA topics, we find a 2D representation of the data using t-distributed stochastic neighbor embedding (t-SNE) and visualize these clusters using the Bokeh library (Bokeh Development Team, 2018). This visualization is provided in Figure 3. The clustering of topics is quite distinct, though there is some overlap, which makes sense given the size of the dataset and number of points we decided to visualize. More specifically, Topic 1 (Dark Blue) has the most overlap with other topics. This makes intuitive sense as it is the most common topic and seems to capture COVID more generally (we can see this through most common words and frequency of the topic).

6.2 Experiment 1: Sentiment Around Milestones

We plot the average twitter sentiment score for each day, which is averaged over 2000 randomly pulled Tweets from mid-March 2020 to end of April 2021. The visualization of the graph with the milestones overlayed are shown in Figure 7. We see that

Topic	Most Frequent Words
1	covid pandemic people corona coronavirus covid19
2	covid cases covid19 amp coronavirus new
3	vaccine covid quarantine people pandemic lock
4	covid19 trump anti people mask wear
5	Trump Fuck covid school shaken children
6	covid new people trump positive vaccine
7	trump covid corona fuck president biden

Table 1: Most Frequent Words for Each Topic

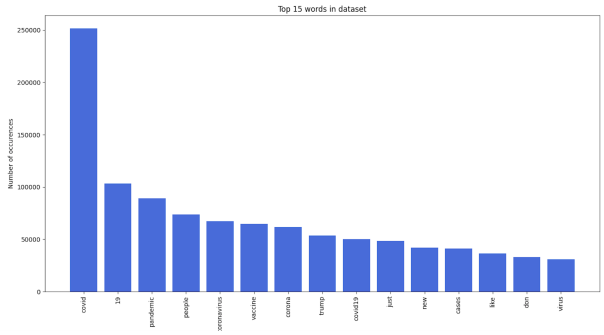

Figure 6: Visualization of the Top 15 Words in our Dataset. Top 15 words in decreasing order: covid, 19, pandemic, people, coronavirus, vaccine, corona, trump, covid19, just, new, cases, like, don, virus

many of the peaks and troughs correspond to key events. We see a positive spike after the FDA authorized a new rapid COVID test, after Moderna published initial promising results, the US Vaccine rollout started, and Biden announced the availability of vaccinnes for all US adults by May. On the other hand, we see negative spikes/drops in sentiment after announcements of Trump testing positive for COVID, the US hit new daily record of cases, Brazil ran low on ICU beds, and the J&J Vaccine got suspended. However, we also notice that many of the spikes, like the positive spike in early September, do not seem to correspond to any particularly big milestone or announcement by public figures.

There are a few potential reasons for the existence of spikes not corresponding to any particular milestone. First, we realize that each day's average sentiment score only looks at 2000 Tweets. Thus, the existence of a couple of outliers or polarized Tweets will have a relatively big impact on the average, in comparison to taking the average of a larger number of Tweets. This is one shortcoming of our dataset, as we wanted to our Tweets to cover a wide range of time so as to have a better understanding on changes in sentiment over the course of a year. Moreover, we note that our dataset size was limited by Tweepy's API, which imposes a 100,000 calls/day rate limit on pulling Tweets (ie taking us over 8 days to create our dataset in the short period allotted for the project). Lastly, we realize that NBC News and the American Journal of Managed Care are subjective sources in that they seldom pick and choose milestones to publish and report on. As such, we may be missing some key announcements or events like those of popular media that were not covered by these sources. We mainly used these sources as they provide key dates over a long range of period, reducing the extra work placed on us to investigate hundreds or thousands of milestones.

Model	Avg F1	Avg Accur.
LSTM w/o Pretrained Embeddings	0.553	57.0%
LSTM + GloVe	0.573	69.7%
LSTM + Word2Vec	0.560	58.5%
CNN w/o Pretrained Embeddings	0.567	61.1%
CNN + GloVe	0.576	67.1%
CNN + Word2Vec	0.573	63.4%
BERT	0.703	82.7%

Table 2: Model Performance

6.3 Experiment 2: Word Embeddings on Model Performance

6.3.1 LSTM

We find that there is a statistically significant difference between LSTM models trained with word embeddings vs. without any pre-trained embeddings. Specifically, the LSTM models with no word embeddings consistently underperform the models with word embeddings. Our p-value ($p = 1.02 \times 10^{-52}$) is statistically significant compared to our threshold of significance $\alpha = 0.05$ (Figure 8).

Of the two word embeddings, we find that training LSTM on GloVe significantly outperforms training LSTM on Word2Vec. Specifically, training LSTM on Word2Vec actually only yields marginal improvements in accuracy over the baseline model. The same results can be seen with F1-score as well ($p = 6.964 \times 10^{-12} < \alpha = 0.05$) in Figure 8.

Overall, our results are in line with our expectation that our sentiment analysis models would

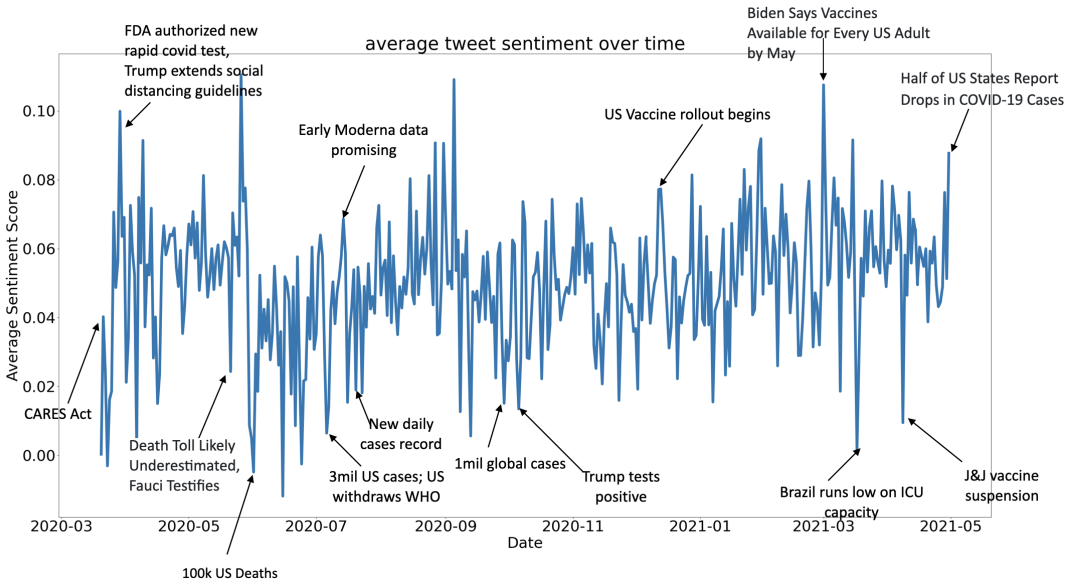

Figure 7: Twitter Sentiment with Marked Milestones

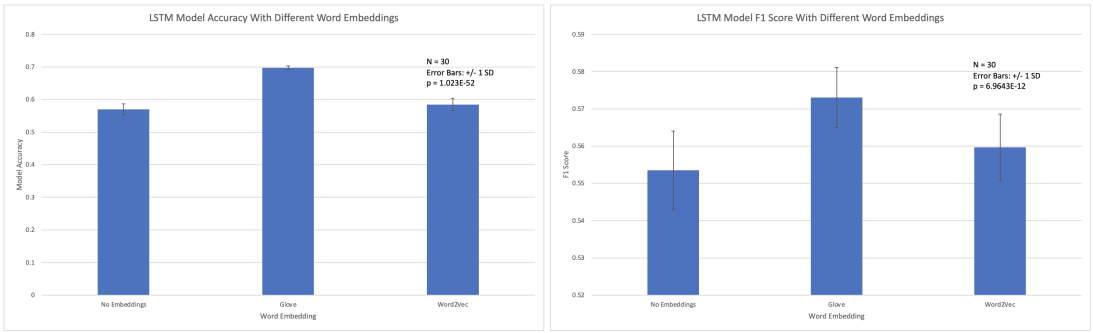

Figure 8: Left: 1-way ANOVA on LSTM Model Accuracy With Different Word Embeddings. Right: 1-way ANOVA on LSTM Model F1-Score With Different Word Embeddings. Left bar: no pre-trained embeddings, middle: Glove, right: Word2Vec

perform better with word embeddings, because our model has more information on the relation of words to each other, and thereby be able to compute a more representative sentiment score. Although it is impossible to say for certain why GloVe outperformed Word2Vec, we suspect that the vocabulary used in the Tweet data we mined may have been more limited, and it is possible that Word2Vec scales better to larger vocabularies, thereby causing it to potentially underperform in a simpler setting.

6.3.2 CNN

We run the same 1-way ANOVA tests on CNN accuracy and F1-score across different word embeddings, and our yield largely the same patterns, with both experiments yielding statistically significant p-values. We should note some important differences, however.

First, we note that the difference in performance between word embeddings (as a function of both accuracy and F1-score) was more pronounced for our CNN model than for LSTM. Specifically, for CNN, GloVe significantly outperformed Word2Vec, which significantly outperformed the baseline model, whereas in LSTM, Word2Vec and the baseline model had more similar accuracy measurements, shown in Figure 9. Additionally, the variance of performance metrics outputted was greater for CNN than for LSTM, visible through the size of the standard deviation error bars. Lastly, our LSTM models generally outperformed CNN, with the exception of Word2Vec, where CNN performed better.

6.3.3 Discussion

These results lead us to a number of interesting discussion points. First, we should note that our LSTM models had more layers than our CNN mod-

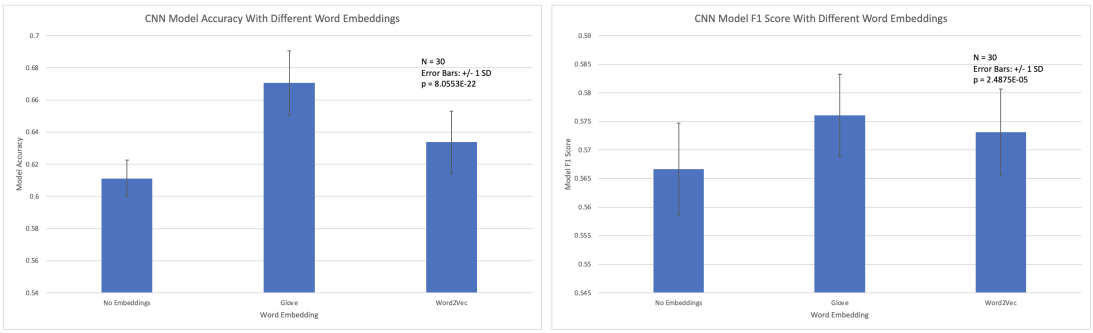

Figure 9: Left: 1-way ANOVA on CNN Model Accuracy With Different Word Embeddings. Right: 1-way ANOVA on CNN Model F1-Score With Different Word Embeddings. Left bar: no pre-trained embeddings, middle: Glove, right: Word2Vec

els. Although the added complexity meant that LSTM took much longer to train than CNN, we do see a marked increase in overall performance, along with decreased variance. Second, we find that different models perform better with certain word embeddings. In our case, we find that LSTM pairs better with GloVe, whereas CNN pairs better with Word2Vec, although in both cases, GloVe outperforms Word2Vec.

Overall, our experiments do verify our hypothesis that our sentiment analysis models perform better with pre-trained word embeddings than without them. Moreover, we find that amongst different word embeddings, GloVe outperforms Word2Vec in our situation, perhaps due to the fact that our Tweet vocabulary size is smaller, and Word2Vec's scalability to perform well on large vocabularies might actually be harming its performance in our situation.

Lastly, we see that BERT outperforms all other models, both in F1-score and accuracy. We attribute this to BERT's ability to capture contextual word representation, which all other methods fail to capture. These results are in-line with previous studies that demonstrated that BERT and its variants (ex. RoBERTa, DistilBERT) perform better than traditional methods such as TF-IDF or word embedding that combine noncontextual word representation methods.

7 Conclusion and Future Work

In this paper, we propose a new COVID Twitter sentiment dataset, perform EDA on the dataset, investigate how much different embeddings (particularly Glove and Word2Vec) impact performance, and benchmark different models' performance on our dataset. Our work's significance is to allow researchers to quickly gauge user sentiment via computation, potentially optimizing the distribution of aid or policy without the need for costly human polling for sentiment. Moreover, we find that for training sentiment analysis models, word embeddings significantly boost model performance, with GloVe models outperforming Word2Vec. As expected, BERT performs the best overall. Our research is significant for providing another basis for comparing model performance across different word embeddings.

For future research, we would like to run more statistical analyses on our dataset and model outputs, as well as do more cleaning and filtering of Tweets to improve model performance. We might consider targeting a narrower dataset with fewer hashtags, as there were many seemingly random and unrelated Tweets. With more time, we would benchmark and compare performance of a wider variety of models, like hybrid models and other transformer LMs, which we did not have time to explore for this study.

8 Author Contributions

All authors contributed to the project equally. NO developed the dataset, ran EDA, and trained BERT. BL and NO created the sentiment analysis timeline. BL and NL developed the LSTM and CNN models and ran the ANOVA analyses. NO, BL, and NL all contributed to the manuscript.

References

AJMC. 2021. A timeline of covid-19 developments in 2020. *American Journal of Managed Care*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Bokeh Development Team. 2018. *Bokeh: Python library for interactive visualization*.

Long et al. Chen. 2020. In the eyes of the beholder: Analyzing social media use of neutral and controversial terms for covid-19. *arXiv preprint arXiv:2004.10225*.

G.; Amenta F. Chintalapudi, N.; Battineni. 2021. Sentimental analysis of covid-19 tweets using deep learning models. *Infect. Dis. Rep., 13, 329–339*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yunhe Feng and Wenjun Zhou. 2020. Is working from home the new norm? an observational study based on a large geo-tagged covid-19 twitter dataset. *http://arxiv.org/abs/2006.08581*.

Kastrati Z. Imran A. S., Daudpota S. M. and Batra R. 2020. Cross-cultural polarity and emotion detection using sentiment analysis and deep learning on covid-19 related tweets. *IEEE Access, vol. 8, pp. 181074-181090, doi: 10.1109/ACCESS.2020.3027350*.

Steven Loria. 2018. textblob documentation. *Release 0.15*, 2.

Ramadhan R. N. Manguri K. H. and Mohammed Amin P. R. 2020. Twitter sentiment analysis on worldwide covid-19 outbreaks. *Kurdistan Journal of Applied Research, vol. 5, no. 3, pp. 54-65*.

Robin Muccari, Denise Chow, and Joe Murphy. 2021. Coronavirus timeline: Tracking the critical moments of covid-19.

et al. Naseem, Usman. 2021. Covidsenti: A large-scale benchmark twitter data set for covid-19 sentiment analysis. *IEEE Transactions on Computational Social Systems*.

László Nemes and Attila Kiss. 2021. Social media sentiment analysis based on covid-19. *Journal of Information and Telecommunication, vol. 5, no. 1, pp. 1-15*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Robert Plutchik and Henry Kellerman. 2013. *Theories of emotion*, volume 1. Academic Press.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. 2009. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pages 248–256.

Joshua Roesslein. 2020. Tweepy: Twitter for python! *URL: https://github.com/tweepy/tweepy*.

AJMC Staff. 2021. A timeline of covid-19 developments in 2020.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

et al. Xue, Jia. 2020. Public discourse and sentiment during the covid 19 pandemic: Using latent dirichlet allocation for topic modeling on twitter. *PLOS ONE, vol. 15, no. 9, Sept. 2020, p. e0239441*.