CMPT 713: Project Report
Using Sentiment Classification to measure online perception

Aidan Vickars
Simon Fraser University
avickars@sfu.ca

Karthik Srinatha
Simon Fraser University
ksa166@sfu.ca

Anant Sunilam Awasthy
Simon Fraser University
anant_awasthy@sfu.ca

Abstract

Due to the power of social media, a companies success often lives and dies according to their perception online. A miss-step by a company that would previously go unnoticed is now witnessed by millions and often results in millions of dollars lost. As a result, it is imperative for every company to be able to measure their perception online to be able to respond before a small miss-step becomes a catastrophic blunder. Thus, in this project we trained two sentiment classification models that are capable of measuring the sentiment towards companies online. To demonstrate the capabilities of each model on real world data, we applied them on news article abstracts and tweets scraped from the New York Times and Twitter respectively for several target companies. The results were vizualized in both discrete distributions as well as time-series representations that demonstrate that the models accurately measure the sentiment of the target companies online. Finally, due to the constantly increasing amount of data and the potential requirement to determine the online perception of a company in a short period of time, a discussion of the speed versus accuracy trade off between the two models is given.

1 Introduction

In this project we worked on the classical sentiment analysis problem from the perspective of a medium to large company that wants to model the sentiment online towards their company. To do this we trained two sentiment classifiers that accurately measure the sentiment towards a company online by determining the sentiment of tweets and short news article abstracts. To be succinct, the models will accept sequences as input and predict their sentiment as either positive or negative. To demonstrate our models on real world data we scraped news article abstracts and tweets for select companies from the New York Times and Twitter respectively. We then applied our models on this data and generated several visualizations that describe the sentiment towards these companies. We found that we were able accurately identify the general sentiment of the target companies and see precise changes in sentiment as a result of highly publicized controversies or shifts in company objectives.

The motivation for this project can easily be seen online where due to the ever pervasive power of social media, small missteps made by companies can become large catastrophic blunders that can result in millions of dollars lost. Thus, it is paramount for companies to be able to recognize when a miss step is made and respond before it becomes a catastrophe. In this project we focused on the former. By developing two models that can measure sentiment as well as by generating interpretable results, companies would be able to effectively take action when the sentiment towards their company becomes negative.

2 Related Work

Sentiment analysis has been studied extensively by researchers and need not be discussed. Instead we focus on works related to our primary objective: measuring the sentiment of companies online. To do this we trained models that can accurately measure the sentiment of two types of data. This is in contrast to existing sentiment classifiers that are trained and evaluated on only one data set at a time. For instance, the original Bert model (Devlin et al., 2018) is trained and evaluated on several data sets individually, but never across multiple datasets at the same time. As a result, its performance across multiple data sets at once remains somewhat unknown. However, we do note that at the time of writing the top sentiment classification accuracy on the Sentiment140 (Kaggle) dataset is 89% (with Code). This was achieved by user

pig4431 (pig4431) on *huggingface.co* in 2022 using a RoBERTa (Liu et al., 2019) style model.

We certainly note that tools have been developed by for profit companies that measure the sentiment of companies online such as Brand 24 (Brand24) and Lexalytics (lexalytics). However, we were unable to find any equivalent open sourced tools. We suspect this is because maintaining such a tool would be very time consuming due to the constantly changing data sources that would need to be updated on a regular basis.

From a research perspective, relatively little amounts of work has been done on this task but we do briefly discuss two works here. Starting with *Measuring e-Commerce service quality from online customer review using sentiment analysis* (Sari et al., 2018) by Puspita Kencana Sari, Andry Alamsyah and Sulistyo Wibowo, Sari and others use a Naive Bayes classifier to measure service quality using customer reviews of *Tokopedia*: an e-commerce company in Indonesia. They achieved a high accuracy of 90%. However, it should be noted that this accuracy is achieved over a very small dataset of just 100 samples. In a different facet, in *Measuring and Managing Consumer Sentiment in an Online Community Environment* (Homburg et al., 2015) by Christian Homburg, Laura Ehm, and Martin Arz, the authors used a Support Vector Machine to perform sentiment analysis. The results were used to measure interesting aspects such as consumer engagement and general consumer sentiment for each of the targeted companies.

3 Approach

In this project we used two models, each of a radically different type. The first is the base BERT (Devlin et al., 2018) model that was chosen mostly out of curiosity because it is one of the most frequently cited models. We now give a brief description of the architecture of the BERT (Devlin et al., 2018) model used. As shown in figure 1, BERT tokenizes the input sequence and computes the embeddings using WordPiece (Wu et al., 2016) embeddings with a $30,000$ token vocabulary. A special classification token $[CLS]$ is inserted at the beginning of the sequence to use for classifications. A separation $[SEP]$ token indicating the end of the sequence and $[PAD]$ tokens to pad the sequence to 512 tokens long are also added. The embeddings are then passed to BERT that is made up of 12 stacked encoders.

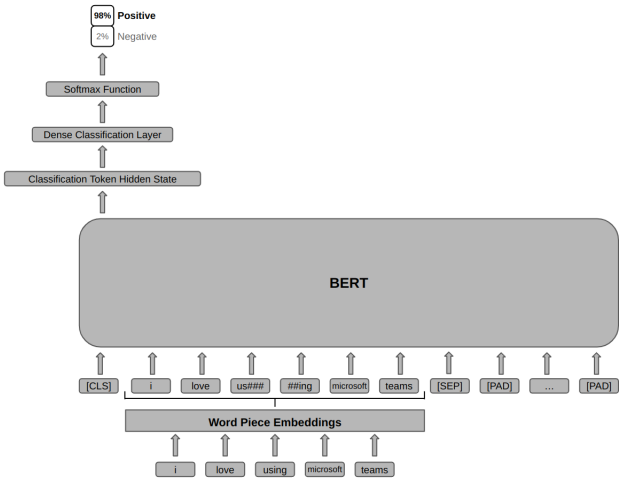

Figure 1: High level representation of the $BERT_{Base}$ model.

After passing through each successive encoder, the 768 dimension hidden state vector corresponding to the classification token is passed to a single dense layer classifier with two outputs where the Softmax function is applied to produce the sentiment classification.

To implement the model we used the pre-trained model available from the Hugging Face library (Face) in Python in combination with PyTorch, and applied a dimension 2 dense classification layer. We used a cross entropy loss to fine tune the entire model to our dataset. Thus, the training and evaluation code was entirely our own but the existing model implementation was used.

The second model we chose was a classical LSTM (Hochreiter and Schmidhuber, 1997) based sentiment classification model where we used the architecture of the model discussed in the article *Sentiment Analysis with LSTM* (Chaudhuri) as a starting point. Our LSTM model shown in figure 2 has a four-layer architecture starting with an embedding layer and a bidirectional LSTM layer with 128 units. The second layer is a fully connected dense layer with 24 units. The third layer is an output dense layer with a sigmoid activation function that predicts the binary sentiment classification for the input sequence. To implement the model, we used the Keras library in Python with a cross entropy loss, and trained the model on our dataset. Therefore, the training and evaluation code is our own, but we used the existing LSTM model implementation (Chaudhuri) as a starting point.

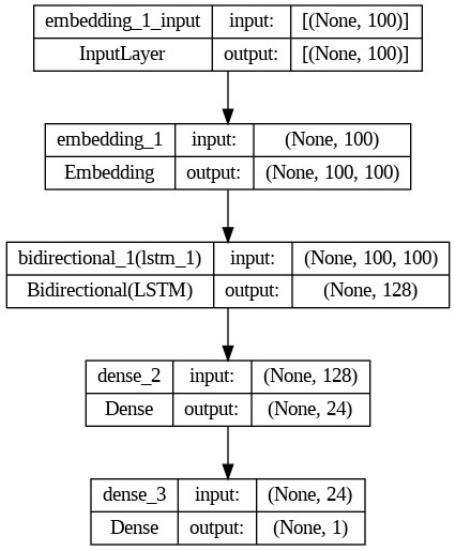

Figure 2: High level representation of the LSTM (Chaudhuri) model.

4 Experiments

4.1 Data

In our project we used four different data sets, where the first two data sets are cleaned and merged together using Python and used for model training. The first is the Sentiment140 (Kaggle) dataset that contains approximately 1.6 million tweets with positive and negative sentiment classification labels. To prepare the data for training, we performed several pre-processing steps that included removing tokens such as "@<twitter handle>", removing non ASCII characters like emojis as well as performing random sampling to reduce the size of the data set. To be succinct, we randomly sampled 4000 tweets from the dataset; the resulting distribution between positive and negative tweets is shown in figure 3.

Label	Count	Percentage	Example
Negative	2006	0.5	my dog is sick he has to go to the vet tomorrow im really worried about him
Positive	1994	0.5	haha adorable that made me giggle i needed it

Figure 3: Sentiment proportions of randomly sampled subset from the Sentiment140 (Kaggle) dataset.

The second, is the NewsMTSC (Hamborg and Donnay, 2021) dataset that contains approximately 11 000 sentences sampled from English news articles with positive, neutral and negative sentiment classification labels. To prepare the data for training, we discarded sequences that had a neutral sentiment label so that the labels matched those in the Sentiment140 (Kaggle) dataset. Similar to the previous data set we randomly sampled 4000 sequences; the result of the sampling is shown in figure 4. Note, very little pre-processing was required for this data set since the sequences are largely grammatically correct and fully formed since they were sampled from existing news articles.

Label	Count	Percentage	Example
Negative	2305	0.58	he is intellectually exhausted out of ideas and out of energy
Positive	1695	0.42	which is to say: she efficiently makes her case like the prosecutor she is

Figure 4: Sentiment proportions of randomly sampled subset from the NewsMTSC (Hamborg and Donnay, 2021) dataset.

The sampled sequences from the NewsMTSC (Hamborg and Donnay, 2021) and Sentiment140 (Kaggle) datasets were merged together to form a training dataset with 8000 sequences. The dataset was then split into a training, testing and validation splits with each split receiving 70%, 10% and 20% of the dataset respectively. The counts and sentiment proportions for each split is shown in figure 5.

		Count	Percentage
Train	**Negative**	3020	0.54
	Positive	2580	0.46
Validation	**Negative**	422	0.53
	Positive	378	0.47
Test	**Negative**	869	0.54
	Positive	731	0.46

Figure 5: Count and sentiment proportions for each split of the training data.

Of course it should be noted why we are merging two different data sets into one training set. This is because of the language differences between news articles and tweets. News article headlines are usually grammatically correct. This is in contrast to tweets that contain abbreviated and generally not grammatically correct sentences. We theorized that a model trained on both data sets could accurately model the sentiment of both types of data where

as a model trained on just one type would not be able to accurately model the sentiment of the other respectively.

The third and fourth datasets contain short news article abstracts and tweets respectively and are used to apply our models on real world data. These datasets were generated using Python by querying the New York Times' and Twitter's APIs respectively as well as leveraging the Selenium Python package to scrape additional tweets from Twitter. Starting with the former, we were able to scrape the abstracts for $3,834$ news articles from January 2021 to March 2023 by querying the New York Times' API (Times) for 6 companies. To be succinct, the abstracts are short one sentence summaries of news articles featured in the New York Times. For instance, the following is an abstract of a news article related to Microsoft from January 2023: *the widely used microsoft teams and outlook email services were unavailable for thousands of users early wednesday*. It should be noted that while we were able to scrape a significant number of articles for most companies with the exception of FIA and the Adani Group, we discovered that the data suffered from a degree of contamination. We found that when the API was queried, many of the articles returned were only weekly linked to the queried company. For instance, consider the following abstract returned for FTX: *mayor francis suarez is selling his city as the world's cryptocurrency capital*. We can see that the only relation to FTX is the topic of cryptocurrency since FTX was a cryptocurrency exchange. Our initial approach was to remove any weakly related articles but found that the dataset became too small. As a result, we made the decision to leave these sequences in the dataset. The ramifications of this is discussed in section 4.4.

Company	Start Date	End Date	Number of Articles
Adani Group	2021-01-14	2023-03-14	46
FTX	2021-03-23	2023-03-14	291
Microsoft	2021-01-02	2023-03-15	1162
Air Canada	2021-01-01	2023-03-15	1196
Meta	2021-01-07	2023-03-15	1133
FIA	2021-05-24	2022-08-31	6

Figure 6: Count of article headlines scraped from the New York Times.

For Twitter, we were initially able to scrape $7,378$ tweets by querying Twitter's API for tweets containing the string "@<company twitter handle>" for each company respectively. However, we noticed that this often performed poorly as it was limited to tweets from the previous seven days. To augment the data we leveraged Selenium in Python and scraped an additional $8,463$ tweets. Note, in this case we scraped replies to each of the target companies tweets respectively. The amount of data scraped using each method respectively is shown in figure 8. The two sets of tweets were subsequently merged together and cleaned; the results of which are shown in figure 9. An observant reader will notice that no tweets were scraped for the Adani Group and FTX using Selenium. This is because after we initially scraped the data by querying Twitter's API, we noticed interesting patterns for these two companies respectively. For tweets related to the Adani Group we noticed that they were either almost always positive or incomprehensible as shown figure 7; we strongly suspect the use of some form of an autonomous agent or bot to inflate their twitter profile. In contrast, FTX simply does not allow replies to their tweets and we were only able to gather a suspiciously few amount of related tweets. As a result, these companies were removed this dataset.

Figure 7: Incomprehensible tweet reply for the Adani Group.

4.2 Evaluation

To evaluate our models, we consider their performance from both a supervised perspective and an unsupervised perspective. From a supervised perspective we used accuracy and F1 score. This is because we wanted the models to be as accurate as possible while not slanting towards producing more false positives or false negatives. From the perspective of measuring the scalability of each model, we used sequences per second. That is, the number of sequence classifications that can be made per second.

From an unsupervised perspective the models were applied on the third and fourth datasets described in section 4.1, and produced descriptive

	API			Selenium		
	Start Date	End Date	Number of Tweets	Start Date	End Date	Number of Tweets
Twitter Handle						
AdaniOnline	2023-03-06	2023-03-06	255			
Microsoft	2023-03-05	2023-03-05	4775	2022-10-17	2023-03-15	1252
FTX_Official	2023-03-06	2023-03-06	51			
Meta	2023-03-05	2023-03-05	831	2020-01-14	2023-02-27	1265
AirCanada	2023-03-06	2023-03-06	864	2022-09-27	2023-03-13	1392
fia	2023-03-06	2023-03-06	602	2021-08-15	2023-03-14	4554

Figure 8: Initial count of tweets scraped from Twitter.

	Start Date	End Date	Number of Tweets	Example
Twitter Handle				
Microsoft	2022-10-17	2023-03-15	4022	still runs like a charm
AirCanada	2022-09-27	2023-03-16	1242	ac 114 is delayed and now we might miss our connecting flight to dc
Meta	2020-01-14	2023-03-15	1111	a completely new capturing experience
fia	2021-08-15	2023-03-16	3790	that's great but what are going to do to win races

Figure 9: Count of scraped tweets from Twitter after cleaning.

visualizations that describe the corresponding sentiment both as a whole and over time. These visualizations are discussed section 4.4.

4.3 Methods

4.4 Results

To begin the presentation of the results, we first present the accuracy of each model. Starting with BERT we fine tuned the model on our dataset for 8 epochs using an initial learning rate of 5×10^{-5}, a cosine learning rate scheduler with a warm up of 1 epoch and an AdamW optimizer. Even though the model was trained for a total of 8 epochs, we selected the model produced after the first epoch. This is because we found that the model immediately began to overfit to the training data after the first epoch. This can be seen in figure 10 where the training accuracy always exceeds the validation accuracy by a wide margin.

Note, additional hyper parameters were experimented with such as different optimizers, initial learning rates as well as different schedulers. We found that they achieved very similar results. A summary of the training hyper-parameters used is shown in figure 11.

The training, testing and validation scores of the

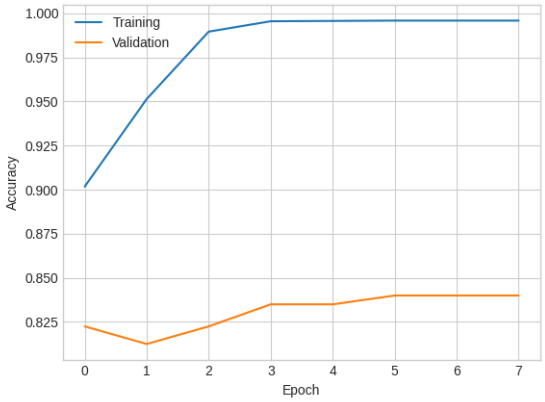

Figure 10: Training and validation of the BERT model for each epoch.

model are shown in figure 12. Over both types of data the model achieved an F1 and accuracy score of 0.8 and 0.82 on the test data respectively. However, we can see that it performed worse on the Sentiment140 (Kaggle) split of the data than on the NewsMTSC (Hamborg and Donnay, 2021) split with respect to the F1 score on each split of the data. This is unsurprising given that the former contains sequences that are often not complete sentences. We do note that the results of the BERT model

Hyperparameters	Values
Epochs	8
Learning Rate	5e-5
Learning Rate Schedule	Cosine with Warmup
Number of Warmup Steps	1 epoch
Batch Size	8
Optimizer	AdamW
GPU	NVIDIA RTX 3090
CPU	AMD 3700X

Figure 11: Training parameters of the BERT model.

are comparable albeit less than the best accuracy achieved on the Sentiment140 Kaggle dataset by user *pig4431* (pig4431) on *huggingface.co* in 2022 as previously mentioned. It is unsurprising that we were not able to exceed the accuracy achieved by *pig4431*, given that they used a more sophisticated model.

	All Data		Sentiment140 Split		NewsMTSC Split	
	Accuracy	F1	Accuracy	F1	Accuracy	F1
Data Split						
Train	0.89	0.9	0.87	0.87	0.92	0.94
Validation	0.82	0.82	0.81	0.8	0.82	0.84
Test	0.8	0.82	0.8	0.8	0.81	0.84

Figure 12: Accuracy and F1 scores of the BERT model.

The second model we trained was an LSTM based model as discussed previously in section 3 where where we trained it for a total of 3 epochs. During training we explored additional hyperparameters such as sequence padding and maximum sequence length. However, we observed that the LSTM model was sensitive to these changes, as it is commonly known that it struggles to handle long sequences and is prone to experiencing vanishing gradients.

After training the LSTM model, we evaluated the model and achieved validation and test accuracy of 0.68 and 0.7 respectively as shown in figure 13. This indicated that the model generalized reasonably well to the data. However, we do note that the model scores far better on the training data compared to the test and validation data. As a result, the argument can be made that the model is over trained on the training data but we were unable to achieve better validation and test scores than what is shown in figure 13.

In figure 13, we can see that in a direct contrast to the BERT model, the model performed significantly better on the Sentiment140 (Kaggle) split of

	All Data		Sentiment140 Split		NewsMTSC Split	
	Accuracy	F1	Accuracy	F1	Accuracy	F1
Data Split						
Train	0.89	0.87	0.89	0.88	0.88	0.86
Validation	0.68	0.62	0.67	0.64	0.69	0.59
Test	0.70	0.64	0.71	0.67	0.69	0.60

Figure 13: Accuracy and F1 scores of the LSTM model.

the test and validation data than the NewsMTSC (Hamborg and Donnay, 2021) split of the data. The cause this is unclear, however one possibility is the model is able to better capture the sentiment of the shorter sequences in the Sentiment140 (Kaggle) split of the data than in in the longer sequences contained in the NewsMTSC (Hamborg and Donnay, 2021) split.

Both models were then applied on the scraped data from Twitter and the New York Times, where the results of the LSTM and BERT model have been vizualized in the left and right columns of figures 15, 16, 17 and 18 respectively. We first consider the results of the BERT model on the scraped Twitter data in the distributions shown in the right column of figure 15, where we made three interesting observations. First, we noticed that Air Canada generally had very negative sentiment. This is not surprising from an anecdotal perspective considering that Air Canada is an airline; airlines often have difficulties maintaining a positive image due to the very negative repercussions that occur when a mistake is made (i.e. flight cancellations, lost bags etc). Second, we can see that in a direct contrast of Air Canada the sentiment towards Microsoft is very positive. This is also not surprising due to the positive attention they have received from their recent integration of the natural language model ChatGTP into their products. Finally, we can clearly see a bow in each distribution for both models. This is almost certainly due to the lack of a neutral sentiment label in the training data. As a result, the sentiment classification of each sequence is pushed towards either positive or negative during training without leaving room for neutrality in the sentiment.

With respect to the differences in the results between the BERT and LSTM models, we can see the similarities when the sentiment is very strong towards either positive or negative. For instance, if we consider the well defined negative sentiment towards Air Canada in figure 15, we can see that this sentiment is captured by both the BERT and LSTM models. However, if we consider Meta in

figure 15 we can see a clear distinction between the outputs produced by the models. For the BERT model the sentiment is pushed to either side where as the sentiment produced by the LSTM model is far more evenly distributed.

Moving onto the results of the models when vizualized over time as shown figure 16 we can see that the sentiment of both Microsoft and Air Canada are consistently positive and negative respectively as previously discussed. More interestingly is the sentiment around Meta and FIA: the governing body of automobile racing. For Meta, we can see a very noticeable increase in positive sentiment that begins between September 2021 and January 2022. This is particularly evident in the results produced by BERT in the right column. We suspect that the cause of this is likely due to their re-branding as Meta from Facebook and their announcement of the construction of the Metaverse that occurred at this time (Paul).

In the opposite direction, in the results produced by BERT in the right column we can see that FIA experienced a very sharp and brief decrease in positive sentiment around November 2022. The cause of this is almost certainly due to the controversial results of an investigation into the Red Bull Racing team exceeding the designated spending limit (Smith). The results of this investigation were released at this time. It should be noted that the LSTM model was not sensitive enough to capture this short steep drop in positive sentiment.

The models were also applied on the New York Times data shown in figures 17 and 18. First, note that the results for FIA and the Adani Group are not shown because our queries to the New York Times' API (Times) generated very few results as shown in figure 6. In contrast, we found 291 articles for FTX and as a result we include it here. In a stark contrast to the results for Twitter, we can see that the sentiment as determined by BERT in the right column is far more neutral for Air Canada, Microsoft and Meta. A similar pattern is shown in the time series representations in figure 18. However, more interestingly is the very negative sentiment around FTX. This is unsurprising since at the time of writing FTX is embroiled serious in criminal allegations. In fact, in the right column of figure 18 for FTX we can see the point where FTX collapses in November 2022 (Mack) where there is a precipitous drop in positive sentiment. We do note that surprisingly the sentiment for FTX appears to recover shortly thereafter. We suspect the cause of this is due to the contamination of the data. As discussed previously in section 4.1, many of the article abstracts scraped from the New York Times' API are only weekly linked to the target company. Thus, it is likely that the general sentiment towards cryptocurrency may be inflating the sentiment towards FTX. As a result, we encourage the reader to interpret these results with caution.

Finally, we give a brief discussion of the speed versus accuracy trade off between both models. In figure 14, we can see that the LSTM model computes the sentiment of sequences approximately 30 times faster than BERT. However, unless millions of sequences need to be processed immediately the better accuracy of the BERT model far outweighs its reduced speed. Furthermore, the speeds given in figure 14 use a batch size of one. As a result, the speed of BERT can be significantly improved by using a larger batch size to increase parallelism. Though, it should be noted that this will increase the GPU memory requirements of BERT.

Model	Data Split	Number of Sequences	Time (seconds)	Sequences per Second
BERT	Test	1600	14.88	107.56
LSTM	Test	1600	0.52	3091.04

Figure 14: Speed of LSTM and BERT models applied on the test split of the training data.

5 Conclusion

From our experiments we determined that classical LSTM (Hochreiter and Schmidhuber, 1997) models simply cannot compete with more modern models like BERT. We found that the LSTM model performed significantly worse when evaluated on the test and validation data. The performance of the LSTM model was even more evident when applied on the scraped data from Twitter and the New York Times where it was unable to accurately capture the sentiment of the target companies. In contrast, the BERT model performed so well that we were able to identify the change in sentiment resulting from particular events. For instance, we were able to identify the change in sentiment for FIA on Twitter as a result of the release of the results of a controversial investigation (Smith).

Of course our methods were not without flaws. First, we found the lack of a neutral label in the training data to be very problematic. In every distribution for both the LSTM and BERT models

in figures 15 and 17, we can see a distinct bow in the distributions. This is very likely due to the lack of a neutral label that pushes the sentiment classifications towards either very positive or very negative. Thus, if we had additional time we would manually add data so that the training data would contain a neutral classification label. To do this we would need to hand annotate several thousand tweets as only the NewsMTSC (Hamborg and Donnay, 2021) dataset contains neutral labels while the Sentiment140 (Kaggle) does not.

Second, as discussed previously we noticed contamination in the data scraped from the New York Times where many of the articles returned were only weekly linked to the target companies. In contrast, when scraping Twitter we were able to ensure a strong relation to the target company by either scraping tweets that contained the target company's Twitter handle or by scraping replies to the target company's own tweets; in both cases a strong relation to the target company is guaranteed. Thus, our conclusion is that using news articles for targeted sentiment analysis does not work well due to the limited number of articles available for each target company. We do certainly note that this could be improved by scraping additional sources, however we suspect that the amount of data would always be considerably smaller that what is available on a social media platform.

6 Contributions

The following is a point form list of the contributions of each group member.

Aidan Vickars

- Cleaned and standardized Sentiment 140 and NewsMTSC datasets
- Trained and Evaluated BERT model
- Wrote abstract
- Wrote proposal
- Wrote milestone
- Wrote report (and made all visualizations and tables)
- Scraped Twitter data from Twitter API
- Formulated submission readiness of repository (i.e. wrote output.ipynb and project.ipynb)

Karthik Srinatha

- Scraped data from New york Times
- Lstm training and Evaluation Scripts
- Prepared presentation material
- Prepared Video
- Contribution in proposal
- Contribution in milestone report

Anant Sunilam Awasthy

- Utilized Selenium and APIs to extract data and subsequently cleaned the Sentiment 140 and NewsMTSC datasets
- Trained and Evaluated LSTM model
- Contributed in proposal
- Contributed in milestone report
- Wrote the LSTM documentation that includes README file
- Improved the LSTM model performance and scores
- Contributed in report
- Developed a Chrome extension which utilizes a Flask API to determine whether a sentiment is positive or negative using BERT

References

Brand24. https://brand24.com//. [Online; accessed 22-February-2023].

Koushiki Chaudhuri. Sentiment Analysis with LSTM. https://www.analyticsvidhya.com/blog/2022/01/sentiment-analysis-with-lstm/. [Online; accessed 22-February-2023].

Papers with Code. Text Classification on Sentiment140. https://paperswithcode.com/sota/text-classification-on-sentiment140. [Online; accessed 19-March-2023].

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

Hugging Face. https://huggingface.co/. [Online; accessed 12-February-2023].

Felix Hamborg and Karsten Donnay. 2021. Newsmtsc: (multi-)target-dependent sentiment classification in news articles. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Christian Homburg, Laura Ehm, and Martin Artz. 2015. Measuring and managing consumer sentiment in an online community environment. *Journal of Marketing Research*, 52.

Kaggle. Sentiment 140. https://www.kaggle.com/datasets/kazanova/sentiment140. [Online; accessed 12-February-2023].

lexalytics. https://www.lexalytics.com/. [Online; accessed 22-February-2023].

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Eric Mack. The fall of ftx and sam bankman-fried: A timeline.

Kari Paul. Facebook announces name change to Meta in rebranding effort. https://www.theguardian.com/technology/2021/oct/28/facebook-name-change-rebrand-meta. [Online; accessed 19-March-2023].

pig4431. Sentiment140_roBERTa_5E. https://huggingface.co/pig4431/Sentiment140_roBERTa_5E. [Online; accessed 19-March-2023].

Puspita Kencana Sari, Andry Alamsyah, and Sulistyo Wibowo. 2018. Measuring e-commerce service quality from online customer review using sentiment analysis. *Journal of Physics: Conference Series*, 971(1):012053.

Luke Smith. Red Bull: F1 cost cap breach penalty "enormous" and "draconian". https://www.autosport.com/f1/news/red-bull-f1-cost-cap-breach-penalty-enormous-and-draconian/10391757/. [Online; accessed 19-March-2023].

New York Times. New York Times Dev Portal. https://developer.nytimes.com/apis. [Online; accessed 22-February-2022].

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

A Appendix

A.1 Figures

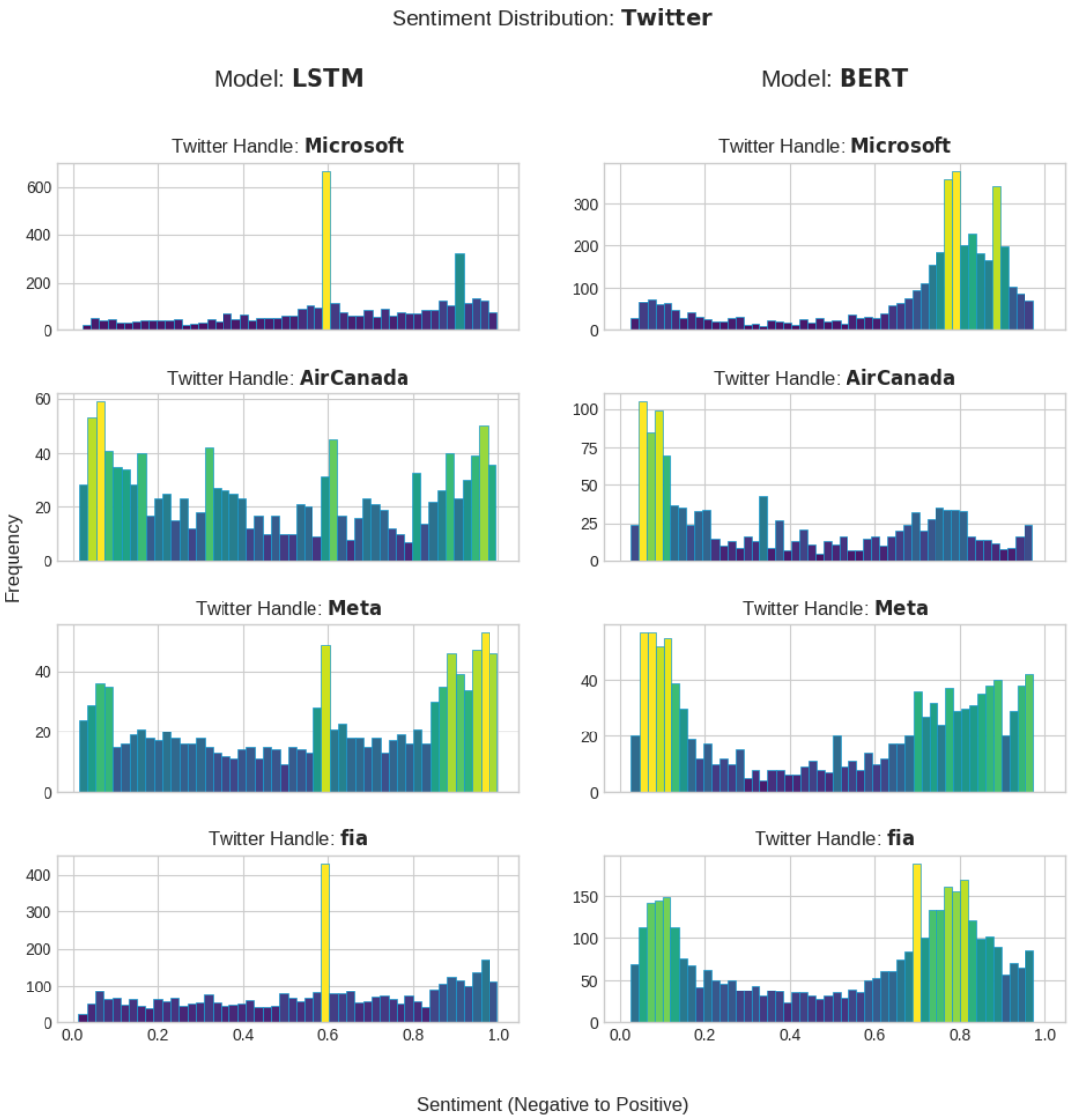

Figure 15: Sentiment distribution for each company on Twitter.

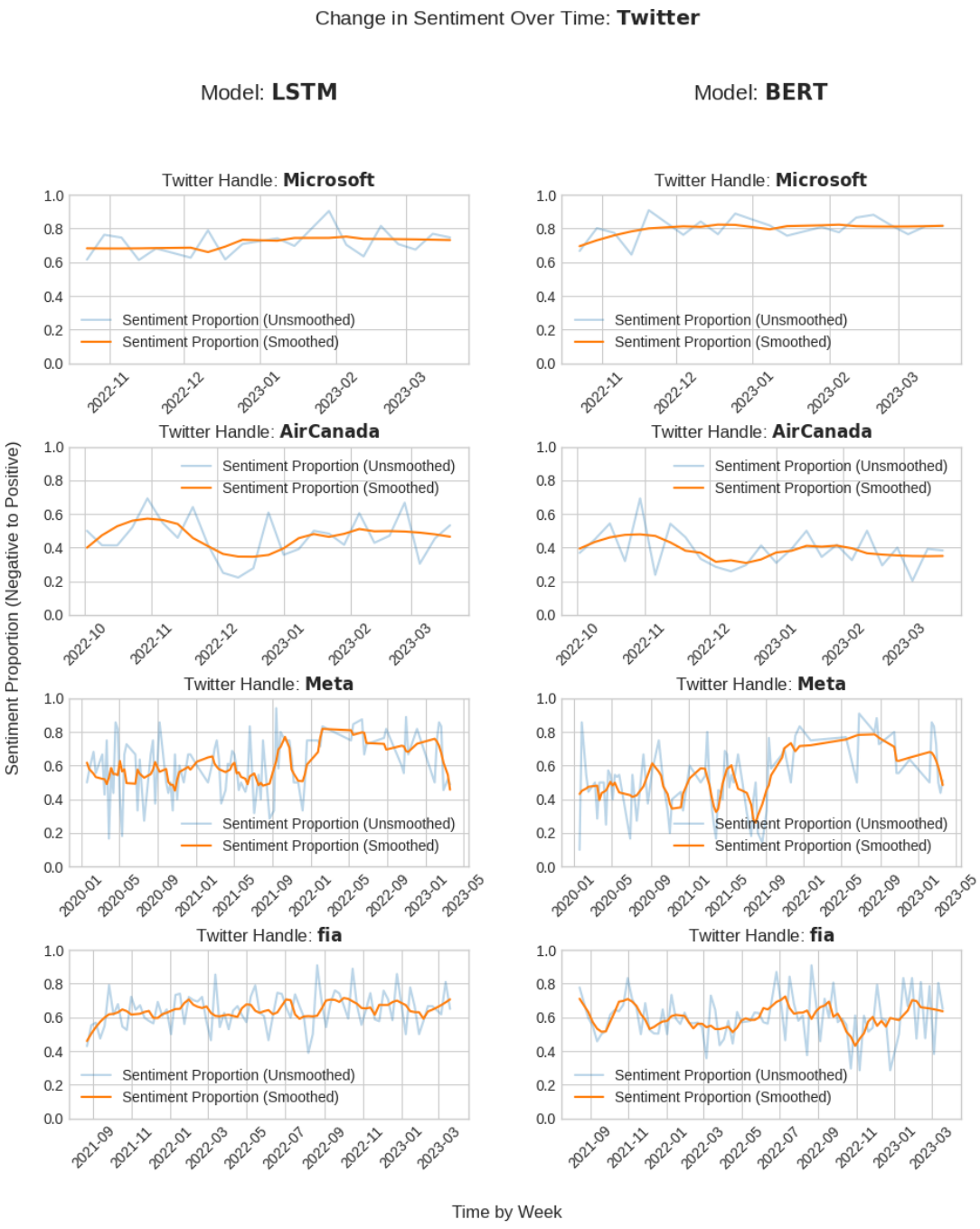

Figure 16: Proportion of positive sentiment week by week for each company on Twitter.

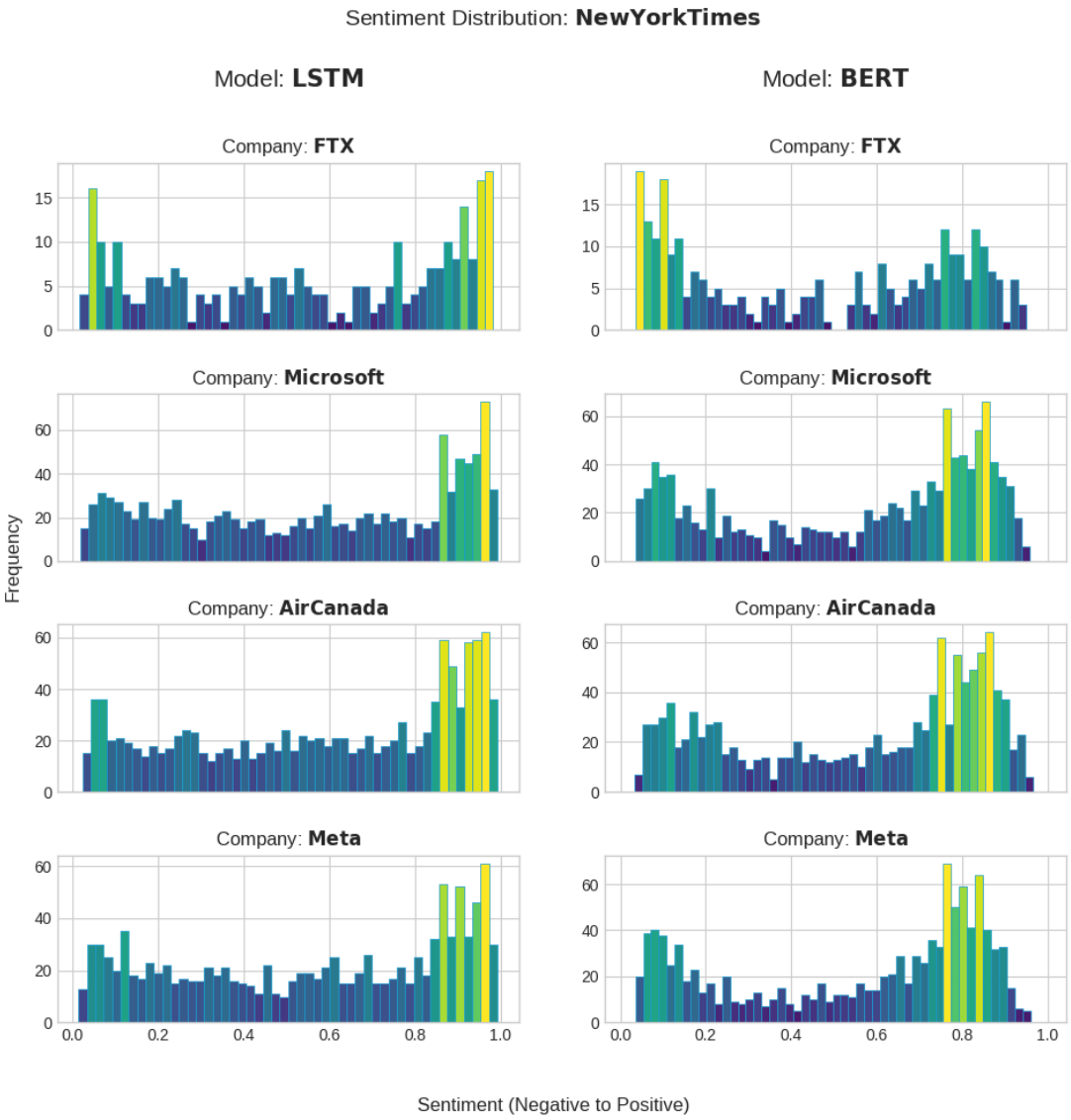

Figure 17: Sentiment distribution for each company on the New York Times.

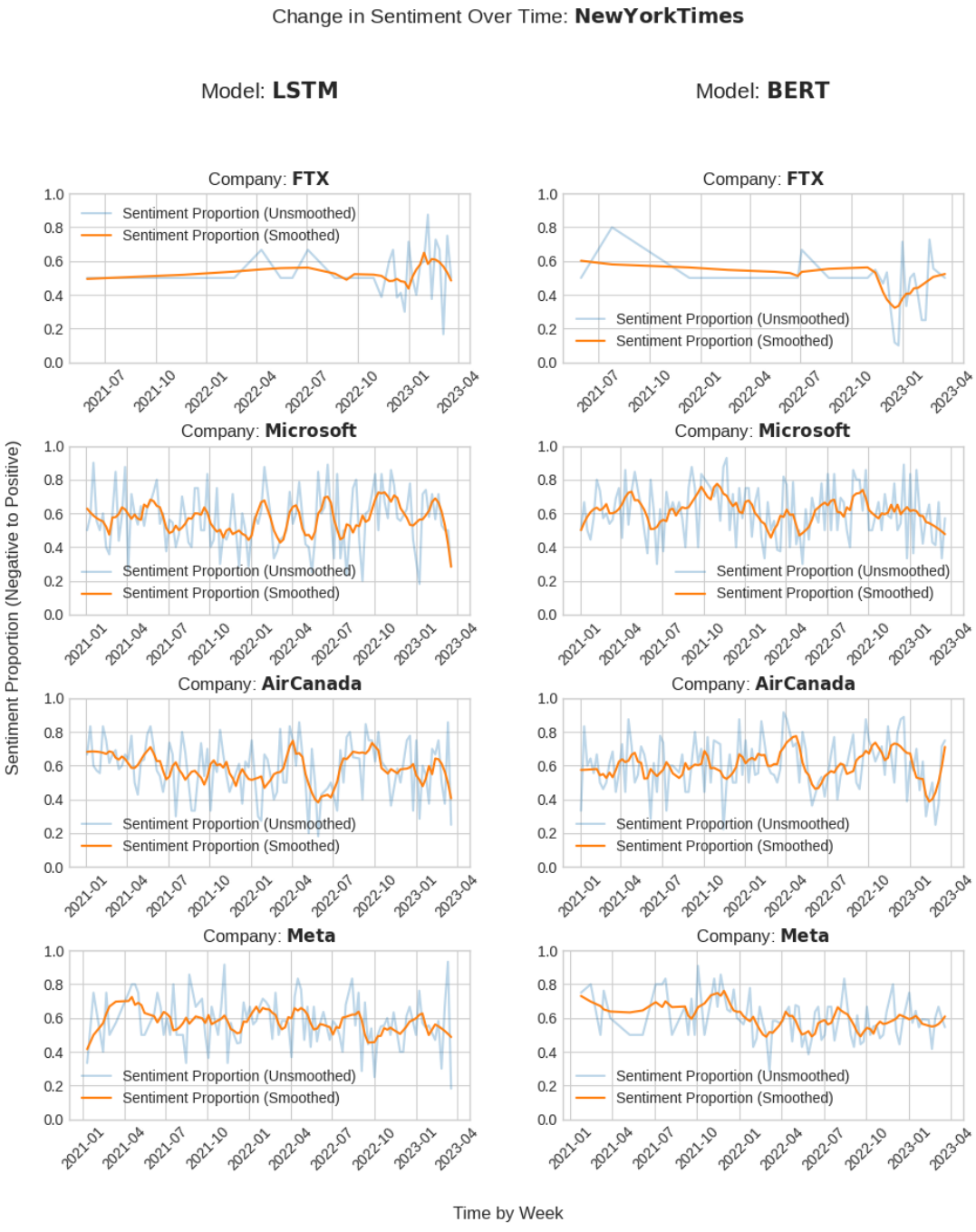

Figure 18: Proportion of positive sentiment week by week for each company on the New York Times.