Natural Language Processing and Representation Learning (DS-GA 1011)
Project 27
Predicting Tweet Reception Using Deep Learning

Team members:
Isi Filipovic (if494), Alex Herron (ah5865), Kristin Mullaney (kmm9492), Joe Schuman (js12580)

1 Introduction

Twitter is one of the most popular exchanges for information on the internet, ranging from local to global scales. Despite the decrease in usage of historically dominant social media platforms (such as Facebook and Instagram), Twitter usage continues to climb. As a treasure trove of text data, Twitter provides an excellent source of public text data from across the globe.

Predicting the sentiment of Tweets has become a very popular task in deep learning in recent years. Twitter provides a platform for users to express their opinions, which can be associated with a multi-class sentiment (positive, neutral, or negative). This sentiment analysis is extremely valuable. From a company's perspective, sentiment analysis can help firms understand their audience, recognize new trends as they unfold, and generally quantify the discourse about their brand. Furthermore, sentiment analysis can be used to offer insight into public opinions on political discourse, recognizing social evils, and identifying hate speech.

Substantially less research has focused on predicting the sentiment of a given tweet's replies, than the sentiment of tweet itself. The aggregate sentiment of a tweet's replies can be used to quantify how the tweet was received by its audience. **We have hypothesized that deep learning models can be trained to predict the aggregate sentiment of a primary tweet's replies, based on the text of that primary tweet.**

For this project, we chose to replicate the paper "How Will Your Tweet Be Received? Predicting the Sentiment Polarity of Tweet Replies." This paper introduced the task of predicting the predominant sentiment among first-order tweet replies. In this paper, Arasteh et al. introduce RETWEET, a data set of tweets and replies. The replies have been annotated both automatically (using a standard Twitter sentiment classifier) and manually (to create a smaller set of gold label classifications). The authors used the automatically labeled data for the supervised training of a neural network to predict reply sentiment from the original tweets. Then, they evaluated the resulting classifier on the manually annotated gold label data. While we used the same modeling methods as the original authors, we conducted our own method of data pre-processing.

2 Methods / Experimental Setup

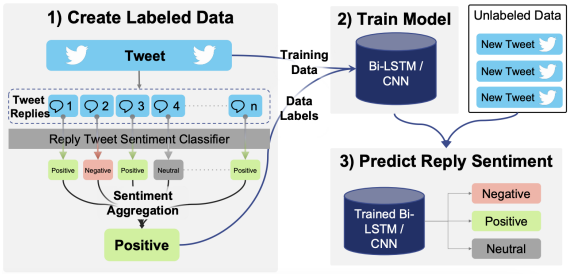

Figure 1: Diagram of data pipeline.

2.1 Data

One of the primary hurdles with this project was managing the data. The paper "How Will Your Tweet Be Received" supplied original tweet IDs and their corresponding sentiment labels. However, these tweet IDs needed to be converted to their corresponding tweets and metadata. In order to accomplish this, we used the Twitter API and 'twarc2' (a command line tool used for collecting and archiving Twitter data in JSON format). Using this tool, we converted the Twitter IDs into a jsonl file (including text and metadata for each tweet), which we then converted to a csv for our own use.

The data provided by the authors comes in two forms. The first is a set of one-to-one tweet IDs and labels that correspond to the original tweet and the aggregated sentiment of all related reply tweets. This dataset, referred to in this report as the **original training data**, includes 34,953 rows of tweet IDs and labels, which the authors claim to be unique. However, we found only 29,501 unique tweet IDs. The remaining tweet IDs are duplicates, some of which have 2 or more contradictory labels. The reason for this duplicate data lies in the summarizer function used to aggregate the reply tweet labels. To remedy this, we leveraged an adjusted algorithm and the second form of data provided by the authors to create our own training set, referred to in this project as the **re-processed**

training data.

The second form of data provided is the pre-aggregation data, including original tweet IDs and the corresponding labels of all the tweet replies associated with that original tweet ID. As such, this second dataset is of the form many labels-to-one tweet ID and includes 34,521 unique (original) tweet IDs. Using an adjusted summarizer function (discussed further below), we were able to create the re-processed training data. This dataset has no duplicates and includes all 34,521 unique tweet IDs from the pre-aggregation dataset.

We then merged the text data obtained using the twitter API to both the original and re-processed training sets, dropping all those tweet IDs without text data (presumably a result of deleted tweets). The original dataset was additionally processed to remove duplicates, selecting the label of the first row in which a tweet ID appears. The resulting training sets included 23,077 tweets and 26,655 tweets in the original and re-processed datasets, respectively.

The last step of preprocessing involved the cleaning of each tweet's text. This involved eliminating punctuation, URLs, videos, links, HTML reference characters, non-letter characters, and Twitter handles. Additionally, all text was converted to lowercase.

For evaluation, the authors included a "gold" test set of 5,015 tweets with their corresponding replies. The overall sentiment of these replies were labeled by three different college students. After accounting for tweets with missing text, and only including tweets agreed upon by all annotators, the final test set included 1,281 human-labeled tweets.

Sentiment Label	Re-processed	Original
Positive	6,248	5,237
Neutral	12,149	10,659
Negative	8,258	7,181
Total	**26,655**	**23,077**

Figure 2: Table of sentiment label distribution for re-processed and original data.

2.2 Methods

Similarly to the authors of "How Will Your Tweet Be Received? Predicting the Sentiment Polarity of Tweet Replies," we used two different model architectures to predict the sentiment of tweet replies.

First, we created a bi-directional long short-term memory model (Bi-LSTM). LSTMs are types of recurrent neural networks (RNNs) often used for pattern recognition in sequences. LSTMs are named for their ability to factor in "long-term memory" and "short-term memory". This means an LSTM model can retain prior short-term memory, or discard it. This results in models that are capable of recognizing longer dependencies in sequences. Bi-LSTMs differ from standard LSTMs because they can process information in both directions (past to future, or future to past). Because of this capability, Bi-LSTMs are particularly well suited to text classification tasks, such as sentiment analysis.

Next, we built a convolutional neural network (CNN). CNNs consist of three layers: a convolutional layer (as the name would indicate), a pooling layer, and a fully connected layer. The convolutional layer apply various filters to an input to create a feature map, which highlights if detected features are present in the input. The pooling layer derives a statistical summary of the nearby outputs. The fully connected layer maps the representation from the input to the output. Although primarily used for image classification, CNNs are also quite useful for sentence classification. As tweet sentiment analysis can be thought of as a form of sentence classification (simply classifying short phrases), CNNs are another good potential model.

Both the Bi-LSTM and the CNN we created for this project closely resembled the models from the original paper. Because of these similarities, we believe the bulk of the differences in our results stem from the differentiation in pre-processing methods between our group and the paper's authors.

The architecture of our Bi-LSTM follows that of the original authors, with hidden and cell states of length 300, embedding dimensions of 200 and a drop probability for each drop out layer set to 0.5. We employed cross entropy as the loss function and ADAM as our optimizer with a learning rate of $9\text{x}10^{-5}$ and weight decay of 10^{-4}.

The CNN is comprised of three one-dimensional convolutional layers with filter sizes of three, four and five respectively, each with an output feature map dimension of 200. Rectified Linear Unit, or ReLU, serves as the activation function. Each input embedding vector is fed into a one-dimensional max pooling layer with a kernel size equal to sentence length so as to remove dependency on the length. After concatenation, the result is a 600-dimensional feature map. Finally, a drop out layer,

a fully connected layer, and SoftMax is implemented to obtain the classification.

For both models, the vocabulary consists of the 750,000 most frequent words from the training set. A GloVe model trained on 27 billion tweets was used to initialize our embeddings.

Furthermore, we included 4 extensions as part of our project. These included 3 extensions that further explored the data, including extended sentiment analysis, exploratory metadata analysis, and SHapley Additive exPlanations (SHAP). Additionally, our final extension involved the use of a pre-trained model (BERT) for the same tasks as our trained Bi-LSTM and CNN.

3 Results

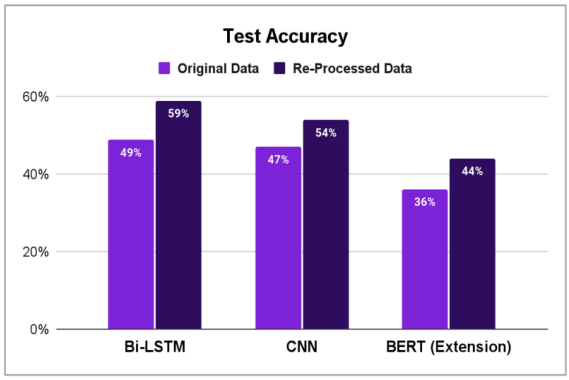

Figure 3: Comparison of test accuracy for each model, for both original and re-processed data.

Metrics	Original Data			Re-Processed Data		
	BiLSTM	CNN	BERT	BiLSTM	CNN	BERT
Val Acc	49.0%	50.3%	38.8%	52.0%	50.7%	49.0%
Test Acc	48.7%	47.4%	36.1%	58.9%	53.9%	43.8%
F1	55.2%	54.2%	36.2%	68.3%	63.9%	44.6%
Recall	56.2%	56.5%	36.5%	75.4%	68.3%	43.4%
Precision	54.7%	52.2%	37.9%	62.9%	60.2%	48.7%

Figure 4: Table of metrics for the 3 models tested (on both the original and re-processed data, both of which have been cleaned). All metrics reflect evaluation on the test data, excluding the validation accuracy column.

4 Discussion

4.1 Hypothesis

Our hypothesis that the aggregate sentiment of a tweet's replies can be predicted using the text of the original tweet was confirmed. Our best model's accuracy was 59%, which far exceeded the baseline of randomly guessing for a three-way multi-class problem ($33.\overline{3}\%$), and demonstrates the ability of a trained deep learning model to predict the aggregate sentiment of tweet replies.

4.2 Extension 1: Sentiment Analysis

Our first extension consisted of performing sentiment analysis. First off, we noticed that the distribution of sentiments were different between the train and test data (as noted in the table below). This highlights how tweets were more likely to be labeled as positive by human labeling than by machine labeling. Next, we conducted our own sentiment analysis of all the primary tweets in the re-processed data. We then split these primary tweet sentiment labels into their corresponding replied tweet sentiment buckets, which can be seen in figure 6. The primary tweet sentiment distributions are different from the distributions of aggregate reply sentiment, indicating that one cannot simply assume that the sentiment of a tweet's aggregate replies will match the sentiment of the tweet itself.

Reply Sentiment	Train	Test
Negative	45.6%	38.7%
Neutral	31.0%	31.9%
Positive	23.4%	29.4%

Figure 5: Comparison of reply tweet sentiment between train and test re-processed data.

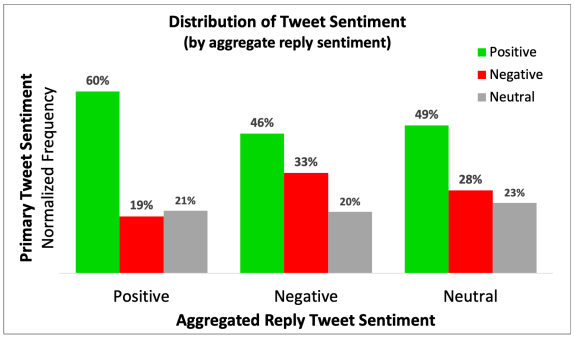

Figure 6: Distribution of primary tweet sentiment, split by corresponding aggregate reply tweet sentiment (re-processed training data).

4.3 Extension 2: Metadata Analysis

For our second extension, we performed an exploratory data analysis using the tweet metadata. The analysis yielded several intriguing observations. We noticed that negative tweets had higher engagement than neutral or positive tweets. Specifically, tweets with negative replies (456 replies/tweet) had drastically more replies on average than tweets with positive (244 replies/tweet) or neutral (206 replies/tweet) replies.

Additionally, tweets on Tuesdays and Saturdays were disproportionately negative, while tweets on Sundays were disproportionately positive. These

results are depicted in figure 7. We do not know if these findings are representative or if they are statistically significant. However, it seems plausible that general sentiment on a social media platform could predictably fluctuate by day of the week.

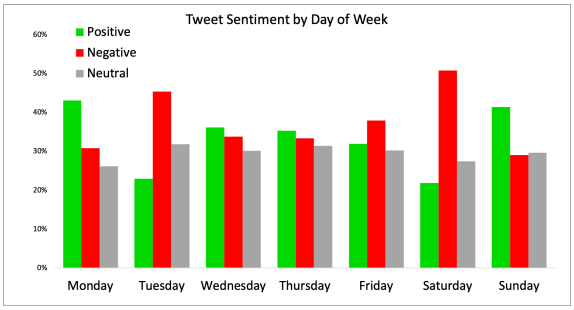

Figure 7: Normalized reply tweet sentiment by day of week for training data.

4.4 Extension 3: Pretrained Model (BERT)

For our third extension, we implemented a pretrained BERT classifier (a state of the art transformer model) to perform the retweet sentiment classification. The BERT classifier was trained for 10 epochs, and resulted in an accuracy of 44% on the re-processed data.

4.5 Extension 4: SHAP

For our fourth extension, we applied the SHAP explainer tool to our data. SHAP (SHapley Additive exPlanations) is a game theoretic approach to explaining the output of a machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory. SHAP can provide insight into the specific words or tokens that contribute to positive or negative sentiment. The SHAP sentiment explainer tool is depicted in figure 8.

Per the figure, we can see that a word like "evil" contributes to a negative sentiment prediction. However, less intuitively, "capitalism" also contributes to negative sentiment. Words like "love" strongly contribute to positive sentiment.

This token level insight could also be used to improve model performance by identifying words that could lead to erroneous labeling and removing them from the training set. However, in our analysis we were not able to identify words that clearly contributed to misclassification.

4.6 Comparison to original study

The accuracy of our Bi-LSTM trained on our re-processed data was 59%, whereas the accuracy of

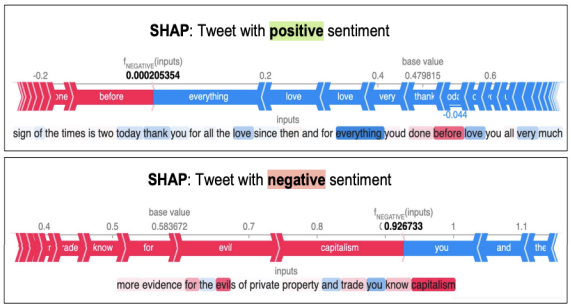

Figure 8: SHAP force plots for tweet with postive sentiment (top) and negative sentiment (bottom).

the original study's CNN was 61%. Although we did not reach the same accuracy as the original authors, we were satisfied with achieving an accuracy so close to theirs. Ultimately, the differences between the original study and our project boil down to the processing of the data.

5 Conclusion

The best model created by Arasteh et al. and the best model we created both outperformed the random guessing baseline substantially. This result substantiates the claim that deep learning models can be trained to predict the aggregate sentiment of a primary tweet's replies, using the tweet's text as input. Notably, the different preprocessing methods applied to the data proved to have a significant impact on model accuracy, with the re-processed dataset resulting in consistently higher test accuracy. This suggests preprocessing is absolutely crucial for working with Twitter data, particularly in the context of automatically labeled data.

6 Author contribution statement

- Isi Filipovic: Creation of re-processed data, modeling, metadata analysis extension.
- Alex Herron: Conversion of tweet IDs to tweet text and metadata, data cleaning, sentiment analysis extension.
- Kristin Mullaney: Creation of re-processed data, BERT model extension.
- Joseph Schuman: Creation of re-processed data, SHAP extension, poster design and data visualizations

References

Arasteh, S.T. et al. (2021) "How will your tweet be received? predicting the sentiment polarity of tweet replies," 2021 IEEE 15th International Conference on Semantic Computing (ICSC) [Preprint]. Available at: https://doi.org/10.1109/icsc50631.2021.00068.

BERT. Available at: https://huggingface.co/docs/transformers/model_doc/bert (Accessed: October 20, 2022).

Welcome to the SHAP documentation - SHAP latest documentation. Available at: https://shap.readthedocs.io/en/latest/index.html (Accessed: October 20, 2022).

PyTorch documentation - PyTorch 1.12 documentation. Available at: https://pytorch.org/docs/1.12/ (Accessed: October 20, 2022).

How to convert tweet id to the actual tweet and associated metadata (no date) twittercommunity.com. Available at: https://twittercommunity.com/t/how-to-convert-tweet-id-to-the-actual-tweet-and-associated-metadata/157648.

DocNow (no date) DocNow/TWARC: A command line tool (and Python Library) for archiving twitter JSON, GitHub. Available at: https://github.com/DocNow/twarc (Accessed: November 16, 2022).

Mining replies to tweets: A walkthrough - towards data science (no date). Available at: https://towardsdatascience.com/mining-replies-to-tweets-a-walkthrough-9a936602c4d6 (Accessed: November 17, 2022).

GETOLDTWEETS3 (no date) PyPI. Available at: https://pypi.org/project/GetOldTweets3/ (Accessed: November 17, 2022).

chayan8 (2020) Sentiment analysis using Bert: Pytorch, Kaggle. Kaggle. Available at: https://www.kaggle.com/code/chayan8/sentiment-analysis-using-bert-pytorch (Accessed: December 7, 2022).