RecoTweet

Victor Arango-Quiroga
varangoq@ford.com

Elizabeth Yam
echyam@gmail.com

Abstract

A set of experiments on sentiment analysis datasets were performed to compare two embedding models, BERT and BERTweet. We are using the BERTweet-large model [3] which was pretrained with 873M English cased Tweets. Our hypothesis was that BERTweet would outperform the BERT model on tweets due to the different data that each model was trained on (e.g. well-structured text vs informal text from tweets). Our findings indicate that the BERTweet model has the same or slightly better results over BERT when we compare them using datasets containing Twitter data. As part of this project, we are also using a pre-built named entity recognition (NER) model to create a general system to perform sentiment analysis on products from a dataset of tweets.

1 Introduction

Tweets are a great source of real-time information in different domains. In this project, we are focusing on a subset of tweets related to a particular company or organization from which we could extract different products mentioned in the tweets, as well as the sentiment in such tweets. Given the nature of informal structure of the tweets, pre-trained models on tweets may perform better than pre-trained models on structured data (e.g. Wikipedia, News). We explored and compared different pre-trained models such as BERT and BERTweet to perform text classification.

The following set of experiments used the datasets described under the data section and explored different data splits. Each experiment was run using two models; BERT and BERTweet.

Experiment 1: Use only dataset #1 for train/dev/test data splits.

Experiment 2: Use only dataset #2 for train/dev/test data splits.

Experiment 3: Use only dataset #3 for train/dev/test data splits.

Experiment 4: Combine train/dev/test sets from datasets #1, #2, and #3.

Experiment 5: Same as experiment 4, but labeling ”not_relevant” and ”Irrelevant” categories as ”Neutral”.

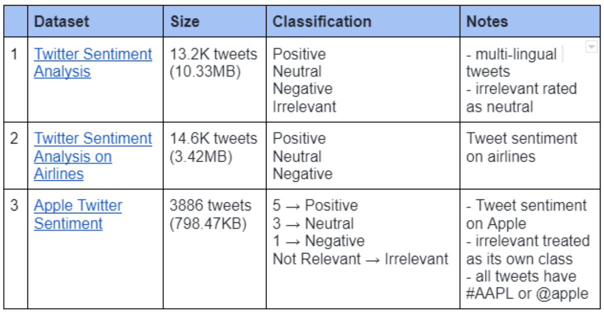
We are using a pre-built NER model from the SpaCy Python package [5] which detects entities in a tweet. We are mainly interested in the products that this NER model can detect, thus we flag the tweets where the model detects an entity of a PRODUCT category.

From the experiments performed, we selected BERTweet and used it together with the pre-built NER model to extract product names and its sentiment. Each product cluster of tweets is mapped to its corresponding sentiment predictions.

2 Related Work

There is a lot of consensus that current NLP tools and pre-trained models are not a good fit to tweets for common NLP tasks such as POS tagging, text classification, and NER. In papers such as *‘Classifying Tweet Sentiment Using the Hidden State and Attention Matrix of a Fine-tuned BERTweet Model*, *Named Entity Recognition in Tweets: An Experimental Study*, and *ner and pos when nothing is capitalized*, the importance of tweets is mentioned given its up-to-date information, but it is also mentioned how current NLP models struggle on this data because their main training corpora consisted of well-structure text as it can be encountered in Wikipedia or news articles. For instance, the last two papers previously mentioned discuss how heavily current models rely on capitalization for segmentation and NER tasks. New proposed strategies and models are trying to decrease this dependency given that tweets and small reviews do not necessarily follow grammar rules. Models such as the BERTweet model are discussed in some of these papers and demonstrate more effective performance than the state-of-the-art models that are not trained on short and informal text. These papers motivated our experimentation to compare the BERT and BERTweet models.

3 Data



	Dataset	Size	Classification	Notes
1	Twitter Sentiment Analysis	13.2K tweets (10.33MB)	Positive Neutral Negative Irrelevant	- multi-lingual tweets - irrelevant rated as neutral
2	Twitter Sentiment Analysis on Airlines	14.6K tweets (3.42MB)	Positive Neutral Negative	Tweet sentiment on airlines
3	Apple Twitter Sentiment	3886 tweets (798.47KB)	5 → Positive 3 → Neutral 1 → Negative Not Relevant → Irrelevant	- Tweet sentiment on Apple - irrelevant treated as its own class - all tweets have #AAPL or @apple

Table 1: Tweet Datasets used: Twitter Sentiment Analysis[6], Twitter Sentiment Analysis on Airlines[7], Apple Twitter Sentiment[1])

Since the above datasets used different sentiment classes, we did some pre-processing to map them to the same 3 bins of positive, neutral, or negative sentiment. For the "irrelevant"/"not relevant" tweets in the first and third datasets, we saw the category was massively smaller than the other three, and re-labeled those tweets as neutral. A quick overview of the distribution of data:

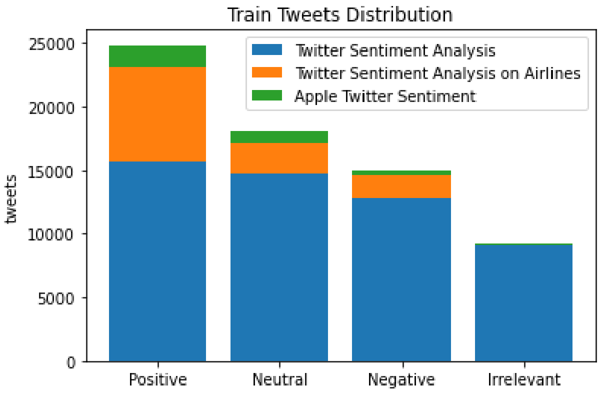


Figure 1: Class distribution of tweets used for training

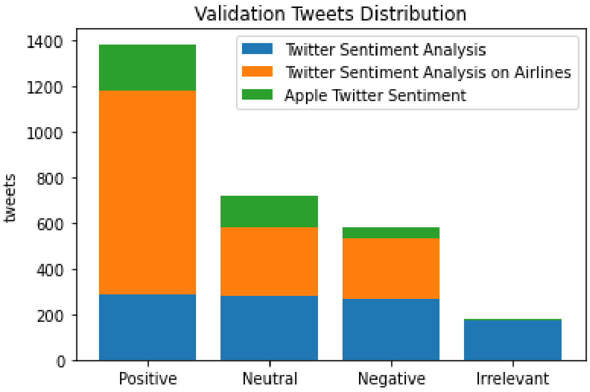


Figure 2: Class distribution of tweets used for validation

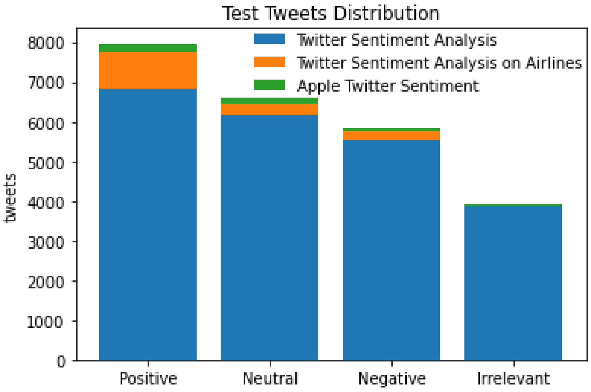
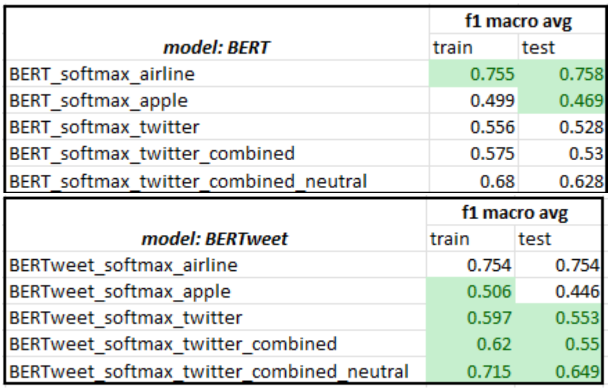


Figure 3: Class distribution of tweets used for testing

4 Models

- BERT [2]
- BERTweet-large [3]

5 Experiments



	f1 macro avg	
model: BERT	train	test
BERT_softmax_airline	0.755	0.758
BERT_softmax_apple	0.499	0.469
BERT_softmax_twitter	0.556	0.528
BERT_softmax_twitter_combined	0.575	0.53
BERT_softmax_twitter_combined_neutral	0.68	0.628

	f1 macro avg	
model: BERTweet	train	test
BERTweet_softmax_airline	0.754	0.754
BERTweet_softmax_apple	0.506	0.446
BERTweet_softmax_twitter	0.597	0.553
BERTweet_softmax_twitter_combined	0.62	0.55
BERTweet_softmax_twitter_combined_neutral	0.715	0.649

Table 2: F1 macro average score for train and test for each experiment on each model (BERT/BERTweet)

As shown in table 2, our experiments on the sentiment analysis datasets which consisted of tweet data indicates that the BERTweet-large model performs very close or better in most cases compared to BERT. BERT outperformed BERTweet on the airline dataset; however the difference in the F1 macro average score on testing was only 0.003. A more significant difference was obtained when testing on the Apple dataset where the BERT model outperformed the BERTweet model by 0.023. On the other hand, BERTweet beats BERT in the remaining other 3 experiments involving a bigger set of data and combined datasets. By inspecting the training column, we could conclude that BERTweet fits the data better than BERT, but it needs to generalize its predictions better. We suspect that if we would have fine-tuned the models, BERTweet could have performed better at generalization and possibly get consistently better results than BERT.

After we completed our experimentation on sentiment analysis, we worked on integrating NER into the sentiment analysis results [4]. Here, we found interesting outcomes from the Apple dataset where we were able to find tweets related to Apple products and obtained their sentiment as shown below:

> [Negative] → None too happy with @Apple Mac OS X (Yosemite), File Vault 2, or Boot Camp right now. Thankful to own a lenovo.
>
> [Negative] → That would be brilliant! It's sad @Apple doesn't allow Mozilla's better engine OpenSourceAgent #opensource
>
> [Negative] → Apple removed songs from iPods without telling customers http://t.co/C29jxFuNcJ
>
> [Neutral] → #AAPL:Apple says plaintiffs' iPods not covered by suit...http://t.co/8V0eYzQFDQ

On the combined and twitter dataset exploration, we found our system also obtaining important information about products that we were not even expecting. The data is not targeting any cluster of tweets in particular as it was the case with the Apple and Airline datasets. Here, our system found some products mentioned such as the Call of Duty Black Ops game for which we found some tweets and their sentiment as shown below:

> [Negative] → Am I the only one who thinks that Black Ops kill notifications / health bars are incredibly distracting? Make them maybe 50
>
> [Neutral] → The Call of Duty Black Ops Cold War Beta was very Good!!! It's a solid 9 / 10 for Me. Only problem in my opinion is how overpowered Grenades are in the Game. Other that it's gonna be a friendly Game!!! Gonna grind Zombies like CZY.
>
> [Positive] → Not gonna lie that Black Ops cold war trailer was probably the best COD teaser I have ever seen

From these results, we see that our integrated system could be very helpful to identify sentiment trends on products where we can easily and quickly inspect the reasoning behind their sentiment. In this case, we observe that the model is not perfect and the tweet marked as neutral could easily be marked as positive given the overall sentiment of the tweet. Since we use a pre-built NER model, we expect to have better results by fine tuning a NER model with twitter data which could be part of future work.

6 Analysis

When we revisited the data originally labeled as "irrelevant"/"not relevant", perhaps dropping them from the experiment would have been a better decision. This category is better determined via NER, and is not actually relevant to the task of sentiment analysis. A quick examination into how tweets originally labeled as "irrelevant" were classified shows there was no dominating class. The relabelling essentially turned the "irrelevant" tweets into noisy training data, since there was no true correlation of sentiment with that class.

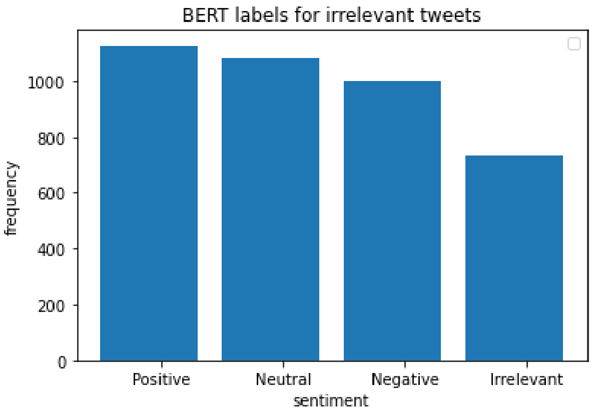
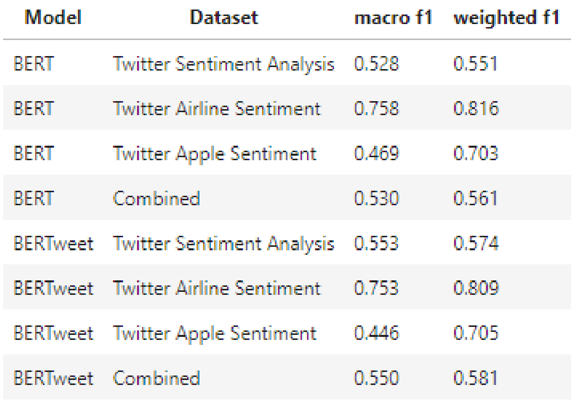


Figure 4: BERT-produced labels on irrelevant tweets

On top of that, looking at the overall distribution of labels in the datasets, all had more tweets in the positive category, which may have skewed the model to learn more about positive vs not positive words (especially in the airline dataset).

Further confirming the hypothesis about noisy data was a comparison of the macro f1 scores with the weighted f1 scores by class. The "irrelevant" class, although smaller than the other three, was sizeable enough to reduce the both models' ability to correlate sentiment-indicative words with the right class. When accounting for the smaller irrelevant class, f1 scores jumped across all runs, especially on the Apple sentiment dataset, with improvements of 0.234 and 0.259 for BERT and BERTweet on the Apple dataset respectively.



Model	Dataset	macro f1	weighted f1
BERT	Twitter Sentiment Analysis	0.528	0.551
BERT	Twitter Airline Sentiment	0.758	0.816
BERT	Twitter Apple Sentiment	0.469	0.703
BERT	Combined	0.530	0.561
BERTweet	Twitter Sentiment Analysis	0.553	0.574
BERTweet	Twitter Airline Sentiment	0.753	0.809
BERTweet	Twitter Apple Sentiment	0.446	0.705
BERTweet	Combined	0.550	0.581

Figure 5: Macro f1 vs weighted f1 scores

We re-ran the experiment only with BERT to see what impact the noisy labels had on the classifier's ability by simply dropping those tweets from the whole process. We saw quite a significant jump in scores.

However, this still does not explain the jump in scores for the airline sentiment dataset, which never had any irrelevant tweets to begin with. The problem here may have been the massive difference in the number of tweets in each class mentioned earlier. Although all datasets had this issue, the difference is so large in the airline dataset as to have more positive tweets and than neutral and negative combined. Because of this, weighting the f1 scores by class size heavily skews towards good performance in the positive class. This might have been better accounted for by ensuring the training and validation subsets were more evenly distributed, forcing the classifier to learn all three classes.

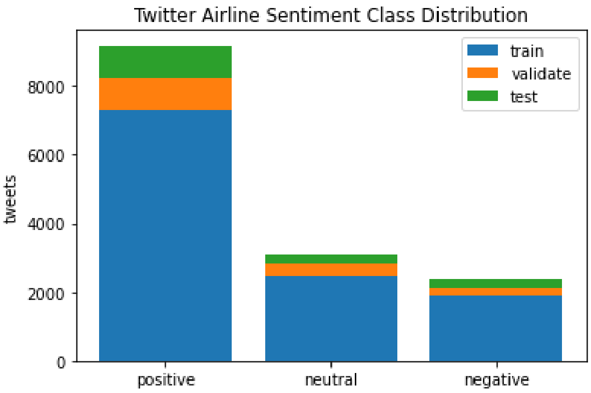


Figure 6: Class distribution of Twitter Sentiment Analysis on Airlines dataset

Surprisingly, BERTweet did not show any significantly improved performance over BERT. When we take a deeper look into the tweets in these datasets, most of them consist of well-formed English sentences, which would render extra training from BERTweet on tweets less useful.

7 Conclusion

Our main conclusion from our project is that both models BERT and BERTweet perform similarly when testing them on Twitter related datasets. BERTweet seems to consistently achieve higher F1 scores during training, but it lacks the ability to generalize that same difference when it comes to testing. This may indicate that by using more data (fine-tuning) and using regularization techniques, the BERTweet model could perhaps perform consistently better than BERT. This could be another hypothesis for future work.

Our work also tried to find a practical application of this sentiment analysis model. To do so, we integrated it with a built-in NER model to detect products on the tweets. Thus, we were able to demonstrate that by having these two models integrated we could have a system to detect sentiment distribution over products detected on a dataset based on tweets.

A Acknowledgements

We thank Prof. Christopher Potts, our course facilitator Powell Molletti and the staff of XCS224U for their guidance throughout this project.

B Authorship

All authors contributed to all project deliverables, experimental design and execution, and analysis.

> Victor Arango-Quiroga - reviewed 4 papers, found datasets, finished sentiment experiments on BERTweet, and ran the NER experiments
>
> Elizabeth Yam - reviewed 4 papers, found datasets, pre-processed tweet sentiment data, ran experiments on BERT, and did some post-experiment data introspection

C References

References

[1] Apple Twitter Sentiment, https://data.world/crowdflower/apple-twitter-sentiment

[2] Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[3] Nguyen, D. Q., Vu, T., Nguyen, A. T. (2020). BERTweet: A pre-trained language model for English Tweets. arXiv preprint arXiv:2005.10200. https://github.com/VinAIResearch/BERTweet

[4] Project Repository https://github.com/vicaranq/CS224-final-project/blob/main/NER_exploration.ipynb

[5] SpaCy https://spacy.io/models/en

[6] Twitter Sentiment Analysis, https://www.kaggle.com/jp797498e/twitter-entity-sentiment-analysis

[7] Twitter US Airline Sentiment, https://www.kaggle.com/crowdflower/twitter-airline-sentiment