Tweet Classification

Angel Umana

Abstract

Twitter data has proved useful in our previous endeavors to assess damages in natural disasters. But there is room for improvement. This calls for a closer look at our Twitter data, in particular on classifying tweets. It is worthwhile to investigate tweets pertaining to everyday topics that are susceptible to disruption in the wake of a disaster event. In this segment of our research, four such topics are studied. Keywords for each topic are developed using previous works. Keywords are assessed for their utility before keyword filtering and collecting samples of tweets. Tweets are labeled on their relevance to the topic, the disaster, and damages. Tweets labeled as topic or disaster-related are further given a sentiment label. Random forest and BERT models are developed to classify tweets as topic-related or not. We find modest success in the random forest model which is greatly surpassed by the BERT model. The models use a method of neighborhoods whereby tweets are sliced around keywords. This method does not yield better performance for the random forest models, but gives superior results for the BERT models while also providing insight into relevant key phrases for classification.

1 Methods

1.1 Data

Tweets from Texas (Harvey) and Florida (Irma) were used. These are the same tweets from previous analyses, with dates ranging from Aug 20 2017–Sept 20 2017 (Harvey) and Sept 1 2017 – Sept 30 2017 (Irma). The preprocessing of these tweets, namely the text cleaning, has already been done as described in §1.2 of the Harvey Twitter analysis summary.

1.1.1 Topics & Keywords

Four topics were developed for classification purposes. They are defined as follows:

- *Housing* – Tweets relevant to the idea of people having a place to shelter in, as well as information pertaining to damage or lack thereof to homes and other residential property.
- *Power* – Tweets pertaining to the conditions of electric utility companies, their infrastructure, availability and lack of service, and outage recovery information.
- *Public Health* – Tweets related to the safety and livelihood of the general, local public (as opposed to, for example, reports of deaths of national celebrities.

- *Transportation* – Tweets pertaining to conditions of transportation infrastructure.

We cannot use topics to filter tweets. Instead, use define each topic by a set of keywords. Those keywords are implemented in a simple keyword filter routine on the cleaned tweets. For housing, public health, and transportation, tweets from Harvey are used. For power, tweets from Irma are used. This is because Florida's power infrastructure was more heavily during Irma than that of Texas during Harvey. The table below shows the topics and their keywords.

Housing	home, house, roof, shelter, tree
Power	blackout, duke, dukeenergy, electric, electricity, fpl, outage, power
Public Health	casualty, dead, die, evacuate, evacuation, fatality, hospital, hospitalize, injure, injury, rescue
Transportation	10, 45, 69, 610, accident, airport, belt8, beltway8, block, close, freeway, fwy, highway, hwy, i10, i45, lane, ln, loop, lp, parkway, rd, road, tollway

Table 1: Topics and their defining keywords

Keywords for transportation were borrowed from Yudi's transportation paper.

1.1.2 Keyword Scoring

After keyword filtering, there is an important step to make before tweet labeling. Almost all keywords used in filtering will inevitably yield tweets that are not topic related. Such is the reason to manually go through tweets and tag topic-related tweets. But are all keywords equally useful? A quick look at tweets filtered from some ubiquitous keywords such such as *close* or *home* yield many tweets not related to housing or transportation. On the other hand, tweets containing *fwy* are **all** related to transportation. So different keywords definitely differ in their utility. The natural follow-up question is how can we measure the utility of keywords? This is where a **score** concept was developed. The procedure goes as follows:

1. Given keyword `kwd`, apply keyword filter on the relevant Twitter dataset (Harvey or Irma) to obtain `kwd_df`, containing n tweets.

2. Randomly sample 20 tweets from `kwd_df`. If the keyword filter does not yield more than 20 tweets, then simply use the whole returned set.

3. Manually label these 20 tweets as topic-related or not. `kwd` is assigned a score of $\frac{r}{20} \times 100\%$, where r is the number of topic-related tweets from the sample.

4. Filter `kwd_df` by randomly sampling $n \times r$ of its tweets. In the case of transportation tweets, the formula was slightly different: sample $100 \times \texttt{int}(n \times r^2)$ tweets.

Using this method, keywords with higher scores (i.e. higher relevance) had a greater proportion of their tweets labeled, and keywords with lower scores had less of their tweets labeled. There was one condition to meet with this process however: the total number of samples per topic. Initially, about 1000 tweets per topic were labeled (later, those numbers were raised to 2000 for transportation and power). For the purposes of meeting these numbers, step 4 was typically more of a guideline; sometimes more tweets would be used for a certain keyword if not enough tweets from other keywords could cover the goal number. This was most prominent with power-related keywords, as the number of tweets returned from filtering with keyword *power* was greater than all other power-related keywords combined.

Similarly, perhaps less tweets were used for certain keywords to prevent a topic to be dominated by that keyword. This goes into the reasoning for having a different formula for transportation keywords: transportation as a topic covered about half of **all** keywords used in this project. In order to better divide the sample of tweets among its keyword filters, the alternative formula was used. It that ensures no more than 100 tweets are used from a single keyword, and that keywords with lower scores are more harshly disfavored; by taking the square of a score, even less tweets will be used. For example, a score of 20% would yield 4 tweets instead of 20. But a higher score is not so harshly trimmed: a score of 90% would yield 81 tweets.

1.1.3 Other Labels

Besides topic relevance, tweets are also labeled on three other categories:

- *Disaster-related* – Does the tweet mention, explicitly or implicitly, the natural disaster relevant to the state that the tweet is from? This can include non-damage related effects of the hurricanes, such as preparing for disaster, anxieties, and prayers and other wishes for good health. Topic-related tweets do not have to be disaster-related and vice versa.
- *Damage-related* – Does the tweet contain mention, explicitly or implicitly, damages pertaining to the natural disaster relevant to the state the tweet is from? Damages can be material (floods, power outages, physically damaged property) or otherwise (displacement of people, road closures). A tweet that is damage-related is also disaster-related, but not necessarily vice versa.
- *Sentiment* – Does the tweet carry a negative, neutral, or positive sentiment? Only tweets that are either topic or disaster-related were checked for their sentiment.

1.1.4 Finalized Data

We labelled 5,438 tweets, amounting to 1,000 housing tweets, 2,340 power tweets, 1,103 public health tweets, and 995 transportation tweets. Plots of Miami-Dade county data are given in the next page to illustrate the tweet distribution and varying sentiments.

There are some notable trends in our labels. For example, transportation tweets tend to be traffic reports with very formulaic structure. This makes them good for supervised learning as there are many of such tweets. However, these traffic reports do not give off any polarized sentiment; most transportation tweets have a neutral sentiment. Thus any

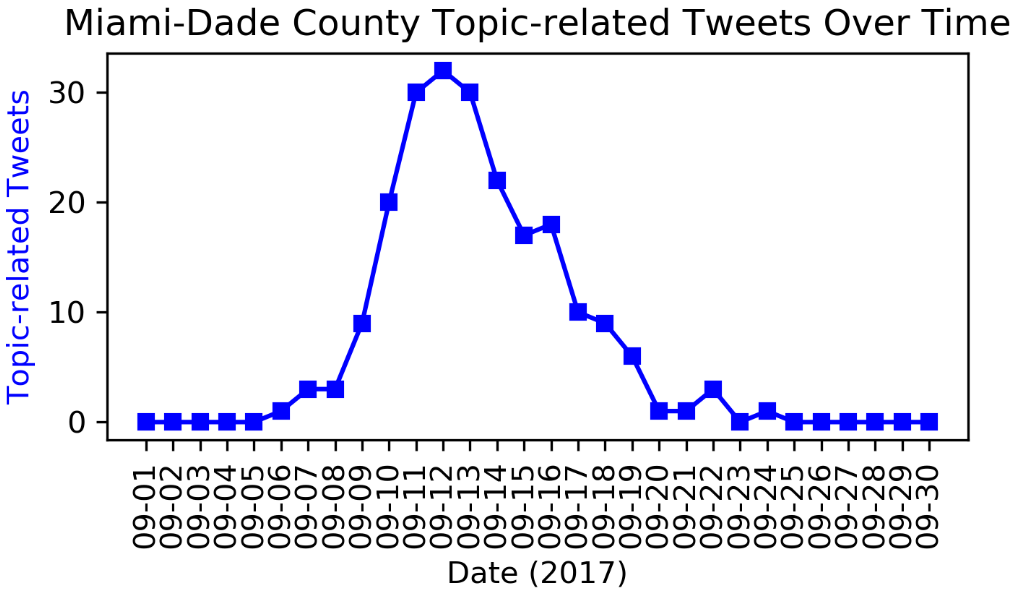


Figure 1: Time evolution of power tweets in Florida's largest county

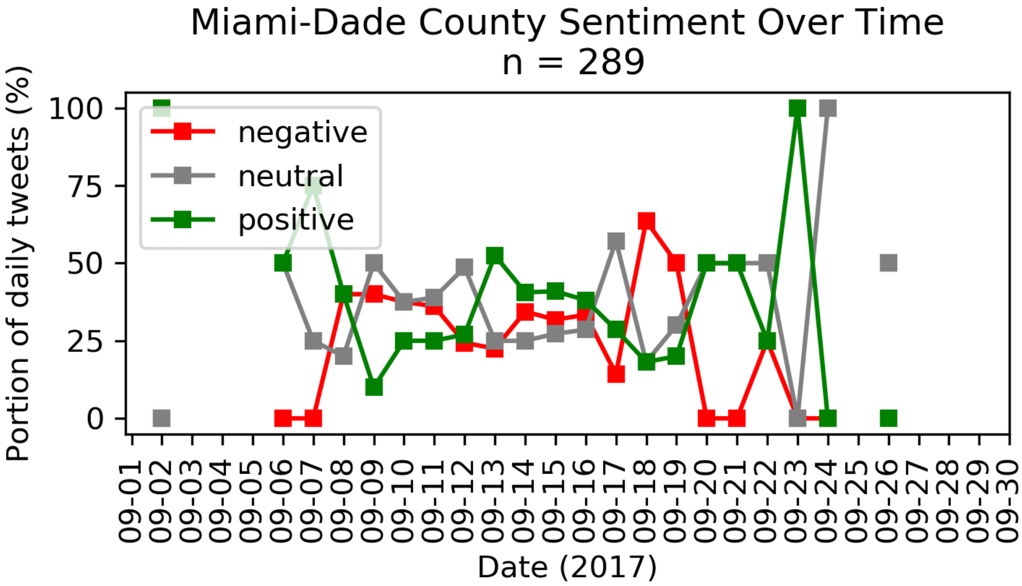


Figure 2: Sentiment evolution of power-related tweets, also in Miami-Dade

sentiment analysis work would be difficult to do with transportation tweets.

Power tweets offer the opposite story. While the tweets do not follow as rigid of a formula as the transportation tweets, they vary a lot more in their sentiment, as we can see in figure 2. At this time, a discernible pattern cannot be distinguished, but perhaps with more data, some trends can be found. If a trend in the power-related sentiment can be found, then further analyses in conjunction with power outage data can be made.

The topics range in their utility. Both housing and public health tweets yielded less than 50% topic-related and disaster-related tweets. Transportation had the highest rate of topic-related tweets, at 82.5%. However, it also had the lowest rate of disaster-related tweets, at 19.6%. Power had the best balance between the two rates, both at 74.3%. These percentages are computed by taking the number of tweets with a certain label (topic or disaster-related) and dividing by the size of the sample of that topic.

1.2 Models

Two models were developed for classification. In one implementation, we used a random forest with input processed through bag-of-words. The other model uses BERT, a deep learning NLP tool. In both models we implement a method of neighborhoods by which a substring surrounding the keyword in a tweet is taken as input rather than the whole tweet. The models only ran power-related tweets as input since this set of tweets was the largest.

1.2.1 Motivation for the Models

Our classification efforts are driven by three goals. We want to develop a model that accurately labels input without overfitting. We want this model to be deployed easily in a timely manner. Finally, we want the model to be able to tell us what key phrases drive the classification.

The random forest model covers these goals well for the most part. The main challenge is that it requires some experimentation with the parameters to provide good results. One convenience is in its automated feature importance. Since we apply bag-of-words to the input, the model is able to rank the importance of n-grams from the phrases. In conjunction with the neighborhood method, this gives us refined information relevant to our keywords, enabling us to improve the methodology from §1.1.

BERT is a recent breakthrough in NLP. It is known to provide state-of-the-art performance. Additionally, the code used to run the model is very hands-off and requires little to no tweaking. It does, however, take up far more time than the random forest model to train and run predictions. Furthermore, unlike random forest, it cannot identify important substrings within a neighborhood (random forest does this with the automated feature importance).

As previously alluded to, the motivation for the neighborhood method is to help identify relevant information surrounding keywords. This will help evolve the current basic keyword filtering method of classifying tweets.

1.2.2 Method of Neighborhoods

Both models have their inputs processed through a neighborhood generating scheme. It is the first step in the preprocessing. The details of this method are as follows. Given a tweet twt, desired neighborhood radius r, and keyword kwd to generate a neighborhood about:

1. Create a copy of *twt* which is split into a list by whitespace.
2. Clean and further tokenize each token (so that we have subtokens). This process is as follows:

    (a) Lemmatization

    (b) Special character removal

    (c) Tokenization using the BERT basic tokenizer, which does the following:

    - Convert text to unicode
    - Invalid character removal
    - Whitespace cleanup
    - Lower casing
    - Strip accented characters
    - Split at punctuation characters

    (d) Removal of single character tokens and stop words

    Henceforth, the previous list of tokens is now a list of list of subtokens; what was originally a token is now a list of subtokens.

3. Find the index of the first instance of *twt* in the list of list of subtokens. That is, find the first list of subtokens that contains *twt* as a subtoken, and record its index.
4. Determine the starting and end indices of the neighborhood according to r.

    - This amounts to attempting to slice the list of tokens (the ones from step 1) into a list of size r to the left and to the right of the keyword found in step 2. Start, for example, by taking all the tokens on the right, then use only the first r tokens. If r is larger than the amount of tokens on the right, then use all of them. Follow the same procedure with the tokens on the left, but reversing the list of tokens so as to start from *kwd* and move out radially.

5. Using the start and end indices, slice the list from step 1 and join that corresponding list with whitespace. This is the neighborhood.

This process is run for every tweet in the data set. The next step in the preprocessing depends on the model used.

1.2.3 Random Forest Input

After generating the neighborhoods, the random forest implementation follows a bag-of-words model by using a count vectorizer to transform the input into vectors. The vectors use n-grams as the basis. These n-grams become the features of the model and are ranked by the model automatically. Stop words are not used as n-grams. A DataFrame with rows consisting of vectorized neighborhoods and columns consisting of features is produced and used as the input.

1.2.4 Random Forest Implementation

During hyperparameter tuning, we were interested in the following parameters:

- max number of features of the count vectorizer
- size of n-grams
- min/max document frequency

After running some tests, it was determined that the min document frequency should be kept at 10 and the max document frequency be kept unlimited. Optimal max count vectorizer features and n-gram size vary for different radii.

1.2.5 BERT Input

The generated neighborhoods are tokenized according to the full tokenizer defined in **tokenization.py**. The resulting tokenization is easily read by the BERT encoding scheme and ready to be input to the model.

1.2.6 BERT Implementation

The actual code for the model is given by Xing Lu. The only modification is that we use the 12-layer model instead of the 24-layer model, as it is sufficient for our purposes. This is the convenience of the BERT model; there is little to no modification or hyperparameter testing to do in order to employ the model.

2 Results

BERT						
Metric	Radius					
	1	2	3	5	10	None
Accuracy	92%	94%	94%	95%	95%	93%
Precision	94%	94%	94%	96%	96%	96%
Recall	95%	98%	98%	98%	98%	95%
F1 Score	94%	96%	96%	97%	97%	96%

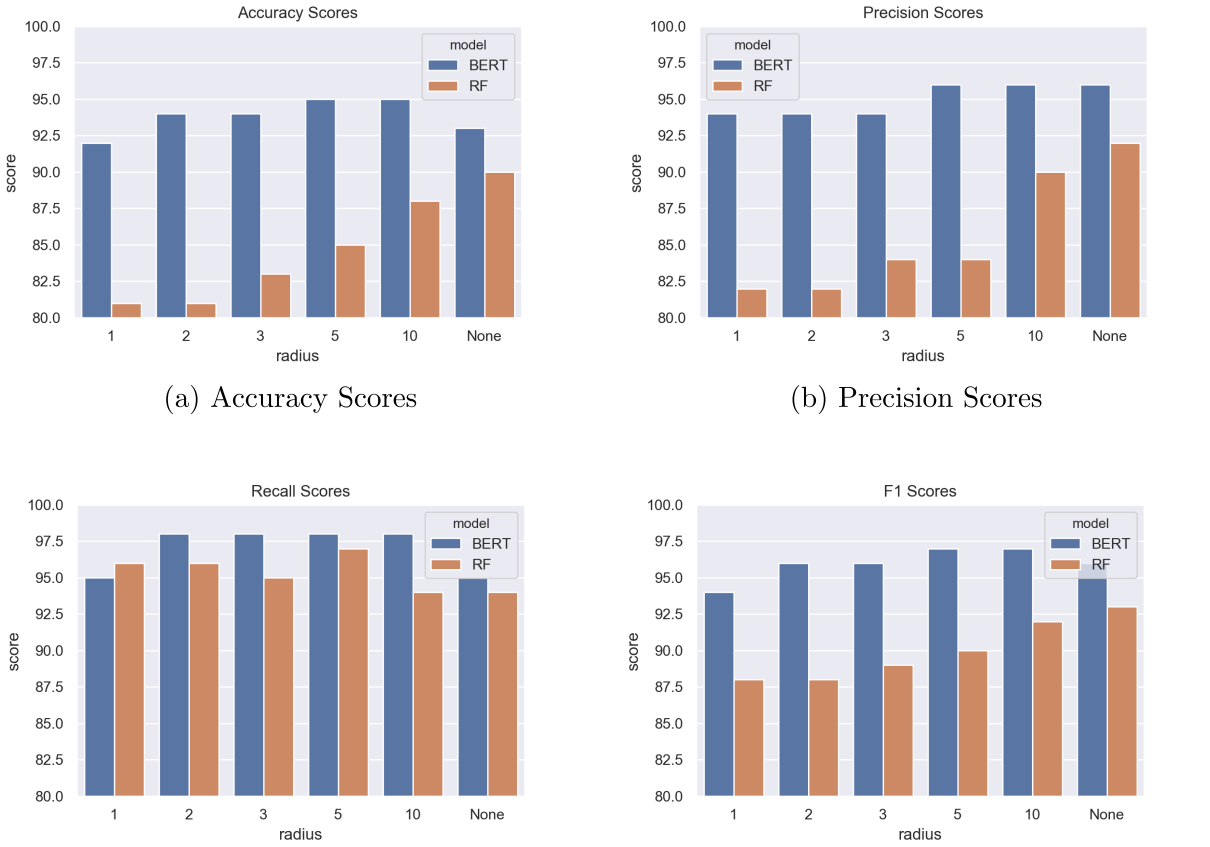
Random Forest						
Metric	Radius					
	1	2	3	5	10	None
Accuracy	81%	81%	83%	85%	88%	90%
Precision	82%	82%	84%	84%	90%	92%
Recall	96%	96%	95%	97%	94%	94%
F1 Score	88%	88%	89%	90%	92%	93%

Table 2: Testing set scores

BERT						
Metric	Radius					
	1	2	3	5	10	None
Accuracy	96%	97%	97%	97%	98%	97%
Precision	98%	98%	97%	98%	98%	98%
Recall	97%	99%	99%	98%	99%	98%
F1 Score	97%	98%	98%	98%	99%	98%

Random Forest						
Metric	Radius					
	1	2	3	5	10	None
Accuracy	81%	87%	87%	85%	87%	89%
Precision	82%	87%	86%	86%	90%	92%
Recall	97%	100%	97%	97%	93%	94%
F1 Score	88%	89%	92%	90%	92%	93%

Table 3: Training set scores



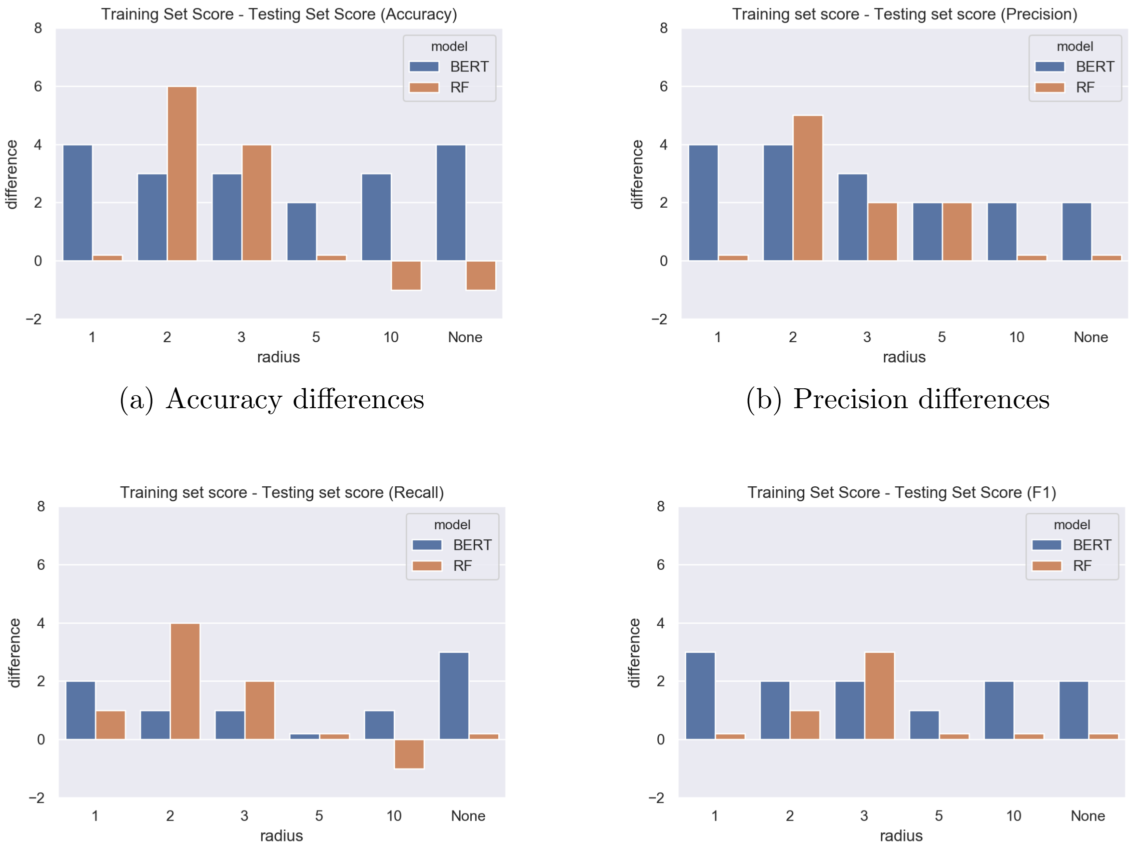
(a) Accuracy Scores

(b) Precision Scores

(c) Recall scores

(d) F1 scores

Figure 3: Scores for both models using different radius values for neighborhoods. ‘None’ means no neighborhood method was used.



(a) Accuracy differences

(b) Precision differences

(c) Recall differences

(d) F1 score differences

Figure 4: Differences between training set scores and testing set scores. A negative value means the testing set performed better than the training set. Bars that appear as only a sliver represent a value of 0; the training and testing sets performed equally.

2.1 Discussion

2.1.1 Random Forest Models

The random forest implementation was outperformed by the BERT implementation, despite hyperparameter optimization. It is also worth noting that the best perfomance comes from *not* using neighborhoods. However, in figure 3, we can still see a trend of increasing performance across separate metrics with increasing radius. This is likely to due how the count vectorizer operates; a smaller radius means an overall smaller corpus for the count vectorization. This could skew the importance of the n-grams. BERT would not run into this issue since it is already pretrained on a large corpus.

It is worth noting that while the random forests models did not offer the best performance, in figure 3 we can see that the best models had very little discrepancy between training and testing scores. Most notably, the model with radius 10 and the model with no neighborhoods both had a net *negative* discrepancy, meaning the testing sets performed better than the training sets!

2.1.2 BERT Models

The main challenge for the BERT model was implementing the preprocessing and neighborhood generation method. Initial challenges included the code not being able to find key words, tokens not being read by the BERT encoder, and resulting scores being sub par. The current version of the code can now run hands-off, and the worst models will still be better than the best random forests. In the data presented, we can see that unlike random forests, BERT can benefit from the method of neighborhoods. Radius 5 neighborhoods offered the best performance, and the scores went down with larger radius. The initial increase in performance is likely due to the fact that BERT is a contextual model. It works better with more information. Small radii will obscure the context surrounding keywords.

It is worth noting that in figures 3 and 4, we can see that the BERT models with neighborhood radii 5 and 10 outperformed the original BERT model in all metrics as well as providing smaller discrepancies between training and testing results, again across all metrics. This provides reason to believe that even though BERT performs better with more information, too much information (i.e. the original model with no neighborhoods) can also obscure the meaning in a tweet. Having sifted through all the tweets myself, I can see that difficulty. One part of a tweet may incline towards a certain label, but another part of that tweet may provide the contrary. This is particularly true for labeling sentiment.

3 Conclusion

What we have now in this project is a working model that can accurately classify tweets as power-related or not without overfitting. The method of neighborhoods was successfully implemented into the preprocessing and has enabled not only superior results, but insight into words and phrases that define a power-related tweet. The project can be extended in a few dimensions:

- Study the classification of the other labels; disaster-related, damage-related, and sentiment
- Attempt to use other topics with the BERT model despite the smaller data set sizes. This could give insight into the sensitivity with respect to the amount of data.
- Improve the keyword filtering methodology by sifting through the useful phrases used by the BERT model. Could use this in conjunction with power outage data to find insights on the relationship between power outages and twitter activity.