Sentiment Analysis

By: Bienvinido Onia, Hongkai Chang, Donghwe Ko

Ch.1 Introduction:

During the past decade, social media has become one of the major platforms in terms of communication. Social media users can exchange messages between each other without any cost and can cast their thoughts to the public at anytime and anywhere. Corporations have also grown a closer relationship between civilians due to the uprising of social media.

Twitter, one of the major social media platform, allows the user to “tweet” anything that comes to their minds. Due to its simplicity, it has attracted millions of users, including huge corporations. Corporations uses Twitter to communicate with customers to receive feedbacks and further improve customer service. Unfortunately, the corporation cannot possibly read every single tweet about them. Therefore, an efficient method of filtering out non-constructive tweets has to be deployed.

The solution to filtering out non-constructive tweets that is proposed is sentiment analysis. Essentially, the sentiment analysis algorithm takes an input (a string of text), and outputs either positive, negative, or neutral sentiment. If the text comes back as positive sentiment, there is a high chance of the text being non-constructive. (i.e. In-n-Out burger makes the best burger out there!). On the other hand, if the text comes back as neutral or negative, then there is a high chance of text being constructive and corporations should definitely take a close look at it. (i.e. United Airlines is by far the worst airline out there).

Through filtering out positive tweets and flagging the negative or possibly neutral tweets, corporations can reduce their social media operation time significantly and they can definitely read all of the tweets about them.

Ch.2 System Design and Implementation Details

*(Note: This chapter will be broken down into the following parts: data preprocessing/feature engineering, additional data preparation, classifier, and tools.)*

Pre-processing

Feature Engineering

1) Since the tweets contained unimportant features, symbols, unimportant texts, and a wide variation of similar words, we removed and changed the following features:

a) Removed all initial features except for the id, text, sentiment, and sentiment confidence

b) all other removals are done on the text

i) hashtags (#), the @ symbol, and web links

ii) stop word removal removes words like a, an, the, of, for, etc. (indefinite and definite articles)

iii) stemmed every word, in which a word’s inflection, or derivations, is reduced to its word stem, base, or root form. This is done so classification would be easier

2) Since we are aware that accuracy of the text classifier would be highly dependent on the words in its training set, we decided that it might increase its accuracy by basing part of its decision on the number of positive, negative, and neutral words per tweet. So, we then created the following features:

a) number of positive words

b) number of negative words

c) number of neutral words

Additional Data Preparation

1) After feature removal, we took each word in each tweet and placed them into a positive, negative, or neutral array, depending on how they were classified when they belonged to the original line of text.

2) For each array, we then compiled the frequency of each word using the natural language tool kit frequency distribution, to determine the likelihood that a word is positive, negative, or neutral

3) We then wanted to count the number of positive, negative, and neutral words in each line of tweet to determine the likelihood if its positive, negative, or neutral. To do this, for each word in each line of tweet, we then determined if that word belonged in the positive, negative, or neutral array. If they belonged to more than one array, we the used its frequency count in those arrays in which it belonged, to determine its likely sentiment. For each line of tweet, you would then get the number of positive, negative, and neutral words

Classifiers

After pouring through several research papers, we have found that the three algorithms that are mostly used in text sentiment analysis are: [1, 2, 3]

1) Linear Support Vector Machines (SVM)

2) Multinomial Naïve Bayes

3) Maximum Entropy, or Logarithmic Regression

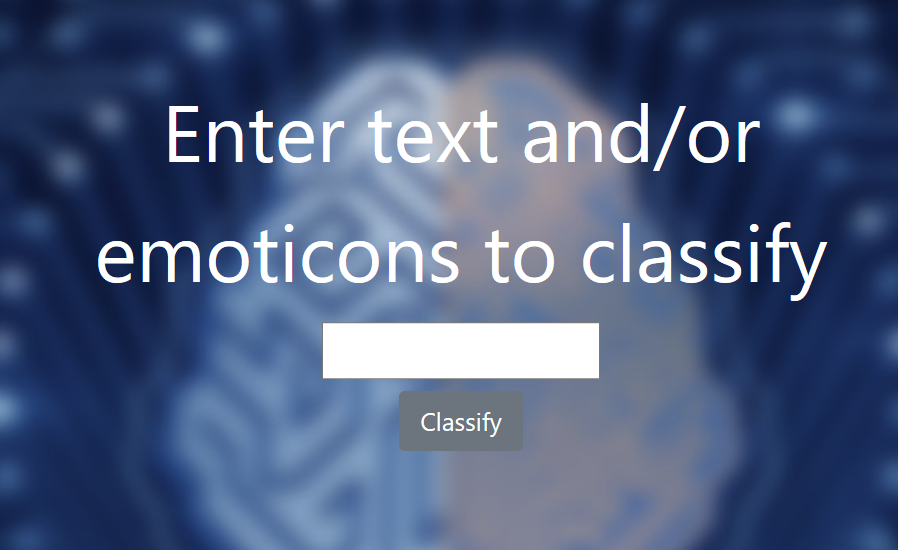
4) ensemble

a) voting classifier: to possibly increase the accuracy of the target class value

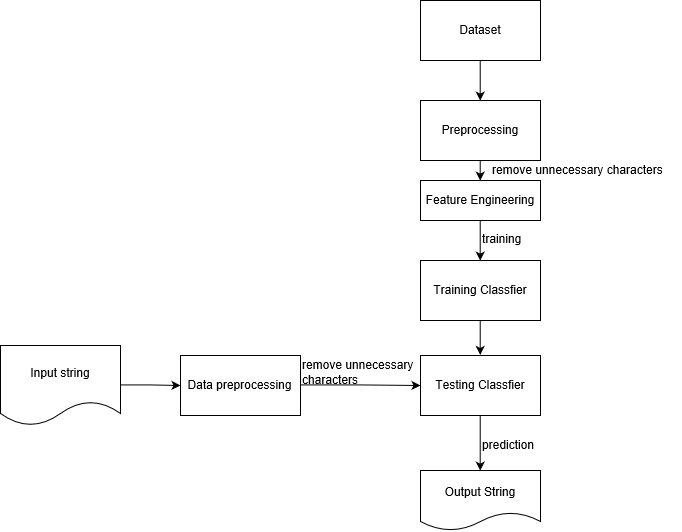
Tools

1) The Natural Language Toolkit (nltk) library [5] was used for word tokenizing, removing stop words, and stemming each word in the tweets.

Diagrams



*Figure 1:* UI



*Figure 2:* Dataflow

Ch.3 Experiments/Proof Concept Evaluation

Datasets:

First GOP Debate Twitter Sentiment: <https://www.kaggle.com/crowdflower/first-gop-debate-twitter-sentiment/data>

Twitter US Airline Sentiment: <https://www.kaggle.com/crowdflower/twitter-airline-sentiment/data>

Evaluation

Accuracy of sentiment analysis is dependent on how well it agrees with human judgement, which is usually only about 80% of the time [6]. So, a program that achieves 70% accuracy is sufficient; a model that has 100% wouldn’t make much of a difference, since humans would disagree 20% of the time, because they disagree by that much for any answer.

Evaluation was done using the holdout method, in which the ratio of training to testing data set is 0.75 to 0.25 for the base and voting classifier. Our only metric evaluation is accuracy and they are the following:

For the Airline Dataset:

a) Multinomial Naïve Bayes: 81.78%

b) Linear SVM: 82.65%

c) Maximum Entropy, or Logarithmic Regression: 82.46%

d) Ensemble: 82.60%

For the GOP Debate Dataset:

a) Multinomial Naïve Bayes: 77.02%

b) Linear SVM: 77.36%

c) Maximum Entropy, or Logarithmic Regression: 77.57%

d) Ensemble: 77.51%

For the combined dataset:

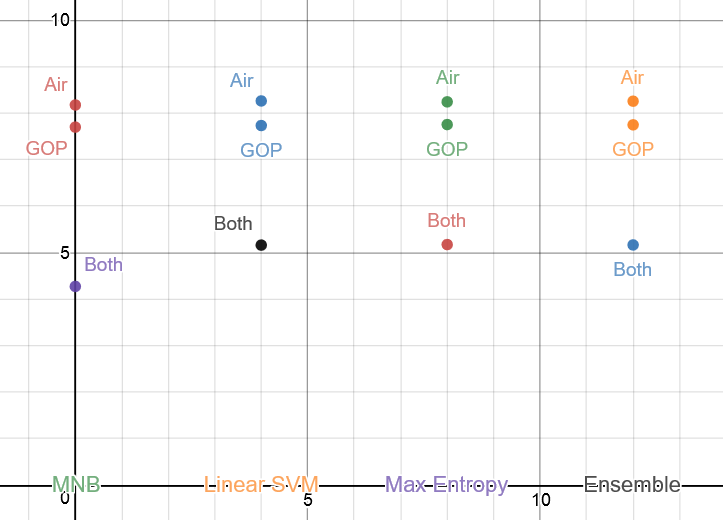
a) Multinomial Naïve Bayes: 42.75%

b) Linear SVM: 51.64%

c) Maximum Entropy, or Logarithmic Regression: 51.75%

d) Ensemble: 51.67%

We discovered that one’s text sentiment classifier accuracy is grossly dependent on the words that are in and their frequency on its training set, and that using text which includes negations, exaggerations, jokes, irony, sarcasm, slang, and abbreviations, such as lol, lmao, etc., which is typically easy for humans to detect, introduces significant inaccuracy and challenges to text sentiment analysis. We believe that this is the reason why the accuracy from the GOP Debate dataset was lower than that of the Airline dataset, since the former included a lot more sarcasm, exaggerations, and jokes. For the combined dataset, we realized that the significantly lower accuracy is due to the two types of context in which the words were being used and mixed in.



*Figure 1:* Evaluation Plot

Ch.4 Discussion and Conclusions

Given the evaluation result, it is obvious that it would not be plausible to train using both of the datasets. The decision has to be made on which dataset is the best dataset for training. However, coming up with a decision is a difficult task as there are major flaws in both datasets that will impact the prediction.

First, the GOP debate result is extremely flawed as it contains a large number of sarcastic tweets. As stated in the previous chapter, although sarcasm can be easily detected by human, it is impossible to detect sarcasm using machine learning (Poe’s Law). This is a huge issue as sarcasm will often skew the meaning of positive and negative sentiment.

Although no sarcasm in the US Airline dataset, the majority of the tweets are positives; there is a huge discrepancy between the number of positive words compared to negative. As a result, the vocabulary pools are full of positive words rather than negative words. This will impact the prediction as some negative or neutral tweets might be interpreted as positive tweets.

In the end, decisions were made and we ultimately chose to use the US Airline dataset as 1) it has the highest accuracy out of all three 2) it doesn’t skew the meaning of positive, negative, and neutral sentiments.

Obviously, because we chose to use the US Airline dataset, we knew that the prediction for positive sentiments will be near perfection while the prediction for the negative and neutral sentiments will be extremely poor. After we tested the algorithm, we have further confirmed our predictions. Texts such as “Today is a good day” and “Today is a bad day” both came back as positive. However, if there are emoticons in the text (which majority of the user uses them), then the prediction will be extremely accurate.

Overall, despite the poor balance between the number of positive words, negative words, and neutral words, the algorithm performed decently. As mentioned above, with the incorporation of emoticons, the algorithm works almost all of the time. With an optimal dataset (no sarcasm, balanced positive, negative, and neutral tweets), the algorithm should and will perform as proposed.

Ch. 5 Project Plan / Task Distribution (1/2 page)

Appendix

[1] Abirami, A. M., and V. Gayathri. “A Survey on Sentiment Analysis Methods and Approach.”

2016 Eighth International Conference on Advanced Computing (ICoAC), 2017.

[2] D’andrea, Alessia, et al. “Approaches, Tools and Applications for Sentiment Analysis

Implementation.” International Journal of Computer Applications, vol. 125, no. 3, 2015,

pp. 26-33.

[3] Goncalvers, Pollyana, et al. “Comparing and Combining Sentiment Analysis Methods.”

Proceedings of the First ACM Conference on Online Social Networks – COSN ’13, 2013.

[4] Han, Jiawei, et al. Data Mining: Concepts and Techniques. 3 rd ed., Elsevier, 2012.

[5] “Natural Language Toolkit.” Natural Language Toolkit – NLTK 3.2.5 Documentation,

www.nltk.org/.

[6] Ogneva, Maria. “How Companies Can Use Sentiment Analysis to Improve Their Business.”

Mashable, Mashable, 19 Apr. 2010, mashable.com/2010/04/19/sentiment-

analysis/#r6koJuA9tPqL.

[7] “Schikit-Learn. “Scikit-Learn: Machine Learning in Python – Scikit-Learn 0.19.1

Documentation, scikit-learn.org/stable/index.html