Analysis of Twitter Data:

Sentiment Analysis and TF-IDF

Mining Content and Relationships in Social Media

|  |  |  |
| --- | --- | --- |
|  | **Curtis Green** CSCE 489 Texas [greencur000@tamu.edu](mailto:greencur000@tamu.edu) |  |



Figure 1: Sentiment scores per tweet according to the AFINN lexicon

ABSTRACT

Labeled twitter data was used to create three models with the ability to predict the sentiment of other twitter statements. The models are the sentiment scores of the tweet, an SVM model using the sentiments scores as inputs, and TF-IDF utilizing SVM as well.

1 INTRODUCTION

There are several ways to analyze the perceived sentiment of a given text, within this project three different methods have been done and analyzed comparing their accuracy through F1 measure and their computational costs. It is often difficult for artificial intelligence and machine learning models to be able to develop the same sense of discernment possessed by humans, so this area of study and research can be very useful in understanding the feelings of users on a macro scale if employed on social media or other online platforms. The objective of this project was to create several models that demonstrate the improved accuracy achievable through improved methods. With “naïve” sentiment analysis simply comparing the sums of the sentiment scores to determine whether they are positive or negative, SVM sentiment analysis using multiple dictionaries as the inputs to an SVM model, and last comparison, TF-IDF, computing by the weight of the frequency and usage of words within the data and inputting that into SVM. The SVM model for TF-IDF uses 10 cross validation to ensure the accuracy is not skewed by potential biased training and test sets.

2 Theoretical Analysis

2.1 Naïve Sentiment Analysis

The twitter data is broken down into words and merged with an inner join on the AFINN sentiment lexicon. AFINN contains a negative to positive scale from -5 to +5 that represents the sentiment score of a given word. Any word that exists in the dictionary will be given the appropriate score, and then the sum of the scores per tweet will be used to determine whether it is positive or negative based on whether the resulting value is greater or less than 0. This does result in any words that are not in the sentiment dictionary to be dropped, in some cases this is beneficial as there are many neutral words that do not bring any meaning to the sentence; however, there are some words that are slang or misspelled within the tweets that will not be into account. This can result in lower accuracy of the model.

2.2 Sentiment Analysis with SVM

The previous naïve sentiment analysis can be improved without much change by cross referencing another sentiment dictionary and utilizing SVM. The data used for this model was total sentiment score per tweet calculated previously using AFINN as well as the total sentiment score calculated using a combination of the Bing and NRC sentiment dictionaries. These two dictionaries are different as they identify more than just a positive-negative scale for words but also include specific emotions such as “joy”. Within this model, however, we only need to reference the positive and negatives scales, so only the labels for positive and negative were chosen and simplified to a 1 and -1 respectively. Similarly as AFINN these scores were summed up and used along with the AFINN scores as the 2 variable inputs to the actual labeled “emotion” output. This accuracy for this model should be better than the naïve sentiment as it cross-references two sentiment dictionaries and SVM identifies the most representative overlap between them.

2.3 TF-IDF

TF-IDF uses a different approach when compared to the previous two methods that use sentiment dictionaries. Rather than having a preconceived definition of the sentiment of each word and its weight, TF-IDF determines the weight based on the frequency of its use within a tweet as well as its frequency of use within all of the tweets in the dataset. If a word is used more often, it is determined to have less weight than a word that is used less often because of language structure. Within an English sentence, there are many words that do not offer any sentimental value, but are only used in order to provide structure, these common words will be thrown out by our models to ensure words like “the”, a completely neutral word, does not bias the dataset. The library used within this project creates a sparse matrix with each unique word as a column and each row as a tweet; tweets containing the word in the column will fill in that data as the TF-IDF score and words not contained in the tweet will be entered as 0. This large matrix is adjoined with the actual emotion of each tweet to determine its accuracy.

3 RESULTS AND DISCUSSION

3.1 Naïve Sentiment

The accuracy of the model, given in [Table 1](#fig1), shows the different measures used to determine which sentiment lexicon is most optimal for this dataset. Bing/NRC, as stated previously, does not use a -5 to 5 scale like AFINN so the sentiment score is determined by the sum of both Bing and NRC for each tweet. In these calculations, accuracy is determined by:

Accuracy = (tp + tn)/(tp + tn + fp + fn)

Precision = tp/(tp + fp)

Recall = tp/(tp + fn)

F1 = 2\*(precision \* recall)/(precision + recall)

Where F1 measure gives equal weight to the precision and recall measures, valuing both “whether the selected data is correct” and “whether the correct data is selected”.

**Table 1:** Comparison of AFINN and Bing/NRC lexicons

|  |  |  |
| --- | --- | --- |
| Atm | AFINN | Bing/NRC |
| Accuracy | 77.64% | 75.33% |
| Precision | 84.09% | 80.38% |
| Recall | 89.54% | 87.92% |
| F1\* | 86.73% | 83.98% |

When analyzing data it is important to know the spread of the data beforehand in order to deal with the potentially biased outcomes of models. Ideally, in a 2-variable model 50-50 divisions of positive and negative sentiment tweets would give the least biased data to determine the sentiment. Within this dataset, there is a very large offset from being an even 50-50 split between positive and negative tweets.

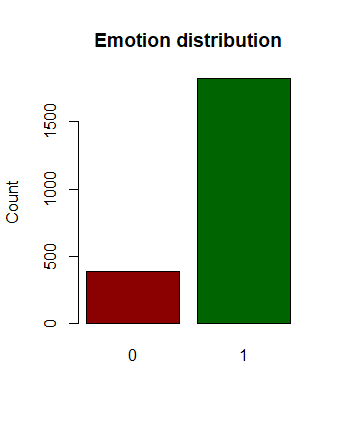


Figure 2: Distribution of Sentiment in Tweets

In Fig. 2 the distribution of positive and negatives tweets is shown, with 0 being negative and 1 being positive. Within our dataset there is a 4.7:1 ratio of positive tweets to negative tweets, this means that a model that chooses 1 100% of the time will result in an 82.4% accuracy. This can create very misleading statistics when it comes to accuracy, which is why recall, precision, and F1 are used as well. A very high score in recall in this case with a much lower score in precision can show that the model is not actually predicting, and would fail on a more balanced training set.

3.2 SVM Sentiment

[Fig. 3](#fig3) displays the classification determined by the SVM model when given sentiment scores of AFINN and Bing/NRC as the independent inputs. By using both of these models, as well as SVM, better accuracy is achieved. As shown in Fig. 4 compared to Fig. 2, F1 measures have increased by 3.3% even when comparing to the more accurate AFINN lexicon.

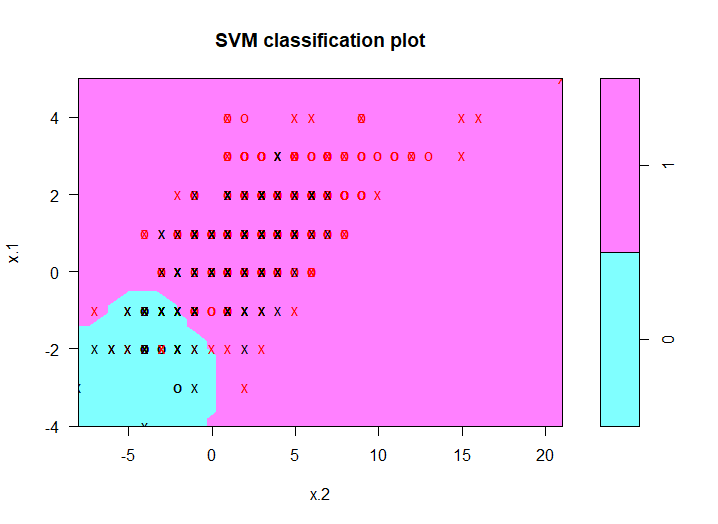


Figure 3: Plot of SVM Sentiment Scores

|  |  |  |  |
| --- | --- | --- | --- |
|  | False | True | |
| Negative | 35 | 12 | |
| Positive | 83 | 533 | |
|  |  | |  |
| Accuracy | 85.67% | |  |
| Precision  Recall  F1 | 86.53%  93.84%  90.03% | |  |

Figure 4: SVM Confusion Matrix and Accuracy Statistics

It is important to note here that the recall score is much higher than precision, which follows the concern stated previously about the large amount of positive inputs. The model is much more likely to consider positive values than negative ones; but because the accuracy and precision are above the 82.4% default correctness it is clear that the model would also predict well on other datasets.

3.3 TF-IDF

As TF-IDF has several thousand columns, one for each word, there is not a way to plot the outcome, but Fig. 5 containing the confusion matrix can be used to evaluate the performance increase.

|  |  |  |  |
| --- | --- | --- | --- |
|  | False | True | |
| Negative | 108 | 7 | |
| Positive | 462 | 2971 | |
|  |  | |  |
| Accuracy | 86.78% | |  |
| Precision  Recall  F1 | 86.54%  96.49%  91.25% | |  |

Figure 5: TF-IDF SVM Confusion Matrix and Accuracy Statistics

TF-IDF is calculated by (Term Frequency \* Inverse Document Frequency). The full calculation can be defined as:

TF = (frequency/word\_count)

IDF = log\_e(corpus/occurrences)

Where:

Frequency = Number of times the word appears in a tweet

Word\_count = total number of words in the tweet

Corpus = Number of tweets in the dataset

Occurrences = Number of tweets that include the word

These TF-IDF weights are used in their respective word/tweet pair within the sparse TF-IDF matrix whose columns are given as the independent variables for the SVM.

4 CONCLUSIONS

In summary, the sum of sentiment scores, sentiment scores with SVM, and TF-IDF with SVM have been evaluated based on their performance in determining whether a given tweet expresses positive or negative emotion. TF-IDF has been determined to be the most accurate model, boasting a 91.25% F1 measure. Using sentiment scores with SVM one is able to get results close to the accuracy of TF-IDF, but requires the discarding of many words that do not lie within the sentiment dictionaries. As language and technology change over time, TF-IDF should be able to maintain accuracy more easily because it does not need to be updated with new words; this is especially important for social media as slang is much more common to be used in this sort of informal setting.

A APPENDICES

A.1 Introduction

A.2 Theoretical Analysis

2.1 Naïve Sentiment Analysis

2.2 Sentiment Analysis with SVM

2.3 TF-IDF

A.3 Results and Discussion

3.1 Naïve Sentiment\

3.2 SVM Sentiment

3.3 TF-IDF

A.4 Conclusions

A.5 References

REFERENCES

|  |  |
| --- | --- |
| [1] | Hvitfeldt, Emil. 2018. Binary Text Classification with Tidytext and Caret *Hvitfeldt’s Blog.* www.hvitfeldt.me/2018/03/binary-text-classification-with-tidytext-and-caret/. |
| [2] | Juba. “Trying to Get Tf-Idf Weighting Working in R.” Tm - Trying to Get Tf-Idf Weighting Working in R - Stack Overflow, Feb. 2013, stackoverflow.com/questions/14820590/trying-to-get-tf-idf-weighting-working-in-r. |