Filtering Noise from the NHL Twitterverse

Troy Yang
Khoury
yang.tr@husky.neu.edu

ABSTRACT

This is my report for the final project. Finding interesting and meaningful content on Twitter can be difficult. Most of the time, this is a manual and time intensive process. The goal of this project was to see if machine learning could be applied to this problem, so that the process would be less time consuming and manual for people. Please note that more detailed results can be found in

1. INTRODUCTION

Twitter generates petabytes of data spanning thousands if not millions of domains. We have gone from an era of being depraved for information and data, to being overwhelmed by the vast quantities of information and data. Because of this, finding content on twitter that is interesting and meaningful and be difficult for end users, and often is a very manual process. The main idea behind this project, is trying to see if machine learning can be applied to this problem. Not nessecarily to completely solve the problem, but to make the process less manual and time consuming. Part of the reason why machine learning is probably not suitable to be applied directly to the problem is because how difficult it would be to come up with a universal and generalizable definition for what is interesting and relevant. If a hard and fast definition could not be had for human labelers, than machines would probably have a hard time as well. I had a theory based on anecdotal evidence that tweets that contain a bit of emotion are far more likely to contain interesting content than a tweet lacking emotion. For this project, the domain of the tweets will be those related to the National Hockey League. The first step of this project was to make sure that it was actually viable.

2. METHODS

In order to ensure the theory about emotion in tweets had any bearing, Professor Vitek advised collecting 100 tweets and manually tagging them with both if there was sentiment and if it was interesting. That would serve as a quick test to check the viability of this project.

After the initial analysis of the 100 tweets suggested that sentiment and interesting content were correlated, the next step was to establish a baseline for a machine learning algorithim that would indicate if a tweet contained any sentiment. I found a pre-trained algorithim called VADER. The acronym stands for Valence Aware Dictionary and sEntiment Reasoner. VADER seemed like a great pre-trained model to use to establish a baseline for many reasons. The creators claimed it had been trained specifically on social media type posts, is generalizable to any domain, and required no training. On the 100 manually labelled tweets, VADER agreed with the manual label 73 out of 100 times.

Before I could begin with training a model, I first had to go find a more robust dataset that I could train and evaluate models on. Because the source of my data is Twitter, I knew I wanted to get labeled data that also came from Twitter. I initially was going to use a dataset I found on Kaggle that contained 100k labelled tweets, however after doing some initial work with the data I realized I wanted a dataset that contained positive, negative and neutral labels, instead of just positive and negative. Without digging too much I was able to find a suitable dataset on Kaggle, provided by the machine learning company called Figure Eight. This dataset contains just under 15,000 tweets from February of 2015. These tweets were made by airline customers describing their experiences with different airlines. Although I did have 100 tweets that I labelled myself, that did not seem like nearly robust enough dataset to build a model that would not suffer from tremendous overfitting. Although the nouns the the tweets are referring to might be different (airlines vs hockey), this did not concern me at all because the objective of the model is to classify sentiment, not classify if tweets are about sports vs airline companies vs anything else. The dataset can be found at this link: `https://www.kaggle.com/crowdflower/twitter-airline-sentiment#Tweets.csv`.

Once I had my dataset, I could then begin working on figuring out how to address cleaning the data. With regards to pre-processing, I wrote a helper function that removes usernames, hashtags, URLs, excess punctuation as well as excess whitespace. When doing background research into Natural Language Processing, I also found that lemmatization is an important step when it comes to pre-processing. In case you are not familiar, long story short lemmatization is very sim-

ilar to stemming, in that if a word is not in its most "raw" or "reduced" form, applying lemmatization would "reduce" it. For example lemmatization would reduce runs to run.

After briefly evaluating some other more NLP specific libraries, it seemed fairly obvious that sklearn should be the main library I use to try to create a better model. A couple features that made sklearn the clear choice for me were the pipelines, parameter search grids, and robust evaluation metrics. The three models I knew I wanted to try were logistic regression, naive bayes, and support vector machines because these were models I had a decent understanding of because I had to implement them in my machine learning class, but also because other people have gotten decent performance for NLP tasks with said algorithms. Based on Professor Viteks advice, I decided that I would also try Adaboost. I also implemented Adaboost in my machine learning class, but in addition to that it is an ensemble model that at each iteration trains a new classifier giving more weight to the points the previous classifier got wrong. As for feature extraction, I used the familiar bag of words and TF-IDF methods. .

When I finally got the results for the other classifiers, I was honestly surprised at how bad the pre-trained model performed, especially compared to the models I trained. I trained my models and found the best parameters on a subset of the airline tweets using 3 folds cross validation. Once I had the best parameters for the given algorithm I then evaluated the tuned models on the testing data, along with the pre-trained Vader model on the testing data. I use the F1 scores to evaluate the models, as it is the most robust single metric (as far as I know).The pre-trained Vader model was by far the worst, with Naive Bayes, Logistic Regression and Adaboost all performing much better. Of the three models I tuned, Logistic Regression with Bag of Words performed the best on test data. I found that all four models had similar fluctuations in performance based on the class. Every model did worst on neutral, better on positive, and best on negative. I was consider wrapping the classifiers with some voting to create an ensemble method, but this would've only made sense if it were clear that models were better at predicting some classes than others.

Now that I a model that can detect sentiment with reasonable accuracy, the next was to apply this data to a new 100 tweets, and have two other people label those tweets as well to serve as validation.

3. RESULTS

I decided that the model should not be evaluated on tweets that the two human labelers did not agree on. That is because there would be no standard to hold the model to if there was disagreement between the labelers. In total, there were 87 tweets that the two labelers agreed on. Of those, the model detected sentiment in and agreed with them on 56. At just under 80%, I thought that was not bad at all. One of the testers said "not perfect but definitely better than nothing" with regards to the models ability to pick out interesting tweets, and I thought that summed things up perfectly. With this small sample size, I thought that the model performed pretty well, but I think if I had more reasources I would have liked to validate with more data. In addition to that I think I would have been better off training the model on a more robust dataset. That is because when I checked which features had the large weights, I saw some domain specific terms such as names or airlines, cities or airports.

All in all, I was quite satisfied with the outcome of this project. I learned a lot along the way and the results I achieved were quite promising. In addition to that, this project has allowed me to explore an out of the box algorithim, as well as tune some of my own.