David Wiszowaty
CS 547, Spring 2016; Final Project Report

Project Goal: The goal of my final project was to categorize Twitter data, or Tweets. Since Tweets aren't categorized, this would require me to develop a system which would take in Twitter data and apply a text categorization algorithm. The benefits of categorizing Tweets is that it would help improve filtering, and allow users to view more tweets that are related to the user's interest.
In this report, I will go into detail on what methods I have used to categorize Tweets and the analysis that I have done on the final results. I will also go into detail on the issues of categorizing tweets, and the methods that I implemented to help overcome the issues.

Approach: In order to categorize Tweets, I had to select a text categorization algorithm to use. The model that I have decided to use is Logistic Regression. The reason why I selected this model is because it requires training data to be used and a pre-defined list of categories. For each testing item, the model will compute the probabilities for each category. The probabilities would represent how close the testing item matches that category. Based on the research I have done, this model produces good accuracy, its a commonly used model, and because of these reasons I have decided to use this model for my project. The first thing that needed to be done is getting the right data and parsing the data correctly into the system. The Tweets can be extracted using Twitters free API. The Tweets are stored in a separate database, but for this project a local copy of the tweets will be stored.
The next piece of data that was required is getting the news data. I used an existing web scraper tool that I have implemented a while back to fetch the lasted web articles that have been published. For this project I have decided to only use CNN for getting news articles. This dataset provides recently published news articles and categories attached to each article. I also have the Reuters dataset, however I am not using this dataset since it is fairly dated. I wanted to use articles that contain terms that are more commonly used today.
I have created a Python script which would load in the Twitter data and the CNN data to be used by the classifier. The first thing that is done is filtering out any numeric terms since they provide very little meaning.
After the text has been filtered, I convert the training data and the testing data into a TF-IDF matrix. The matrix is made up of unigrams, bigrams, and trigrams since phrases add meaning to the text. I am also keeping stop words since they can differentiate phrases. I am also applying Inverse Document Frequency. The matrix is also limited to a maximum of 3 million features.
Once the TF-IDF matrices have been computed, I will then compute the

Logistic Regression model. I am using newton's method to assist in the computation. The model is also applying weights to the categories based on their frequencies. The CNN data contains a varied amount of news articles for each category. This would cause more popular categories to be assigned more since there is more data supporting theses categories. By applying a weight to each category, this can help balance the final results.
Once the Logistic Regression model finishes computing the predictions, the final results would be written to a csv file which would be used by another Python script for analysis and traversing the data. That script also connects the Tweets to news articles based on their assigned categories.

Technical details: The programming language that I have decided to use for this project is Python 3. I am using various packages to help me implement these algorithms. The main package that I am using is called $''sklearn''$. This package allows me to perform various algorithms and analysis on the data.
The data which I am using are Tweets from the United States that have been posted on December 30th, 2015. The only information that I am using is the $''tweet\_id''$ and the $''tweet\_text''$. I managed to get approximately 50000 tweets, using Twitters free API.
The news dataset used is from CNN.com. I have a web scarper that is automatically scraping the latest news articles that have been published on the CNN website. The file contains 12,000 news articles dating from November 2, 2015 to April 4th, 2016. The data contains the published news article, the headline, and the category. The article body and its category will be used to train the model. Other news dataset's can be used, if the article body and categories are provided.

Evaluation: Before I started implementing the classifier, I expected a lot of Tweets to contain poor categorization since Tweets are limited on the number of characters used. There also exists plenty of Tweets that aren't related to a specific category. Since the training dataset is different from the testing dataset, this would also have a potential negative impact on the categorization since the style of the text is different between the datasets. This was noticeable when I began my analysis of the categorized Tweets, and I was surprised at the percentage of Tweets that weren't able to be categorized.
As I mentioned above, I have created a separate Python script which is used for traversing the analyzing the computed results. The user is given three options, seeing the results of my manual analysis of the Tweets, seeing the results of the analysis done when the news articles are the testing dataset, and finally the user can traverse the Twitter dataset.
I have also implemented Logistic Regression using the CNN data as both the training and the testing dataset, and I am using the same exact conditions for when Tweets were used as the testing set. I am applying Cross Validation

with a KFold of 10 on the news dataset. At each iteration, I am computing both the Micro and Macro F1 score, the precision, and the accuracy of the computed results. At the end, these values would be averaged. These values would be used to compare how well the Tweets have been categorized, and how much of an impact does transfer learning have on the testing set. Below are the values which I have computed for analyzing categorization on News Articles:

Final average Accuracy: 0.87349
Final average Micro F1: 0.87349
Final average Macro F1: 0.89273

Since Tweets don't contain any categories, this caused me to manually look through the results and record which Tweet was placed in the right categories. The way this is done is that I will focus on the categories with the highest probabilities for each Tweet. I will also focus on a subset of the tweets from each unique category. The subset is determined based on the probability values of the assigned categories. The highest probabilities will be examined for each unique category.
Then I will go through each Tweet and mark the Tweet as 1 if I believe that the category assigned to the Tweet is correct. A 0 is assigned if the category assigned to the Tweet isn't correct. The results of my analysis is contained in an excel file titled, $''pred\_with\_msg\_Analysis.xlsx''$.
About 37% of the Tweets I analyzed were assigned the correct categories. When comparing to the analysis done on the news data, the difference is quite significant. Having news articles as the testing dataset, results in much better accuracy when compared to Tweets.