BOOK REVIEW BASED ON TWITTER DATA

Rejul James **Dayana mallasserry Sunny**
A20345321 **A20346386**

INTRODUCTION:

Twitter is an online social networking and micro blogging site that has millions of users today. Twitter users 'tweet' at least twice a day on an average, thus generating a massive amount of data. The objective of this project is to give an accurate rating for book based on tweets made by the users and geographically plot the spread of users. People will see contradictory/different ratings for same books in different websites sometimes. We try to predict the rating of a book based on the data we collected from the twitter.We have selected seven books of different ratings and collected data for these books using search api of tweepy.API in python. The expected conclusion is that the rating will be equally accurate as that of the ratings from Amazon or Goodreads, since it is designed to reflect the opinion of the people.

DATA

Twitter API:

We collected data from twitter to classify our tweets to positive, negative or neutral using Twitter API. Twitter API can be accessed only by authenticated requests, which must be signed with valid Twitter user credentials. We created a special Twitter account for this purpose, with valid OAuth keys to communicate with the Twitter Search API. In order to overcome rate limits of twitter, we used tweepy search API. We enhanced and overcome the rate limits by using the since_d and max_id fields of the tweepy search method. We identified the last tweeted user and extended our search back up to two weeks and collected the latest 5000 tweets in a single request. The relevant fields collected were Tweet, User ID, User Name, Location, Retweeted etc. Out of these fields,we used Tweet, location and User ID for our project. Due to the limitations of data collection, we chose recently published 7 popular books and tried to classify them to either as positive, negative or neutral (unimportant) . The maximum number of characters that you can search for are 500 including operators. The search query we used were the book name + author's last name so that we could avoid irrelevant tweets with same words as that of the book name. Below given is a sample image showing the fields of data collected. We experimented data collection using Twitter Streaming API as well, which in turn retrieved very less number of tweets for a specific book on a daily basis.

In order to filter out tweets which are in English, we used a python package called "guess language". Guess language predicts the language of a tweet accurately. We removed Retweets as well . Retweets are commonly abbreviated with "RT". After the data pre and post processing, we manually labeled first 1300 tweets to train the classifier. Out of this 1300 tweets, 250 negatives, 400 positives and 600 neutral tweets were included.

Google Static maps API

The user location collected from Twitter API has been supplied to the Google maps API and retrieved the place_id Information. Google gives a json fle including all the place information. We collected around

5.7k user locations and their place_ids. We eliminated the irrelevant/meaningless locations by validating it against 50 countries and their cities(1 lakhs of locations).Now using the validated place_id field, latitude and longitude values are obtained from Google Maps API.

METHODS:

For this project, we used two classifiers. The proposed method of classification was using Naïve Bayes. Logistic Regression is another classifier we have used. Depending on different classifiers, the data was pre-processed in different ways. Live data was collected about some books and exported into a csv file. The most challenging part in the building of this was to develop an appropriate training set for rating books based on Twitter user data. We preprocessed the texts for removing all the unnecessary punctuations, @USER names, http urls, HEX values and similar irrelevant data. Since all the tweets extracted to develop the training data had to be specifically written for a book, these had to be preprocessed to make sure all spam, book name, and other proper nouns like names of Actors were removed so that they don't confuse the classifier. But many of the tweets do not explicitly specify a good or bad reviews, we didn't remove the book names, We used a set of stop words and removed these stop words from our vectorizer to increase accuracy. We preprocessed, tokenized tweets and we avoided negation by appending not with next two consecutive words. People sometimes tweets "lloooooovvvveeeeeeeeeeeee" to show the intensity of the tweets. We used methods to avoid repetitions of character with single character itself.

Logistic Regression Classifier:

Logistic regression is a discriminative probabilistic statistical classification model that can be used to predict the probability of occurrence of a event. It predicts the probability of occurrence of an event by fitting data into logistic function. For training the classifier, we used 1300 tweets and vectorized these tweets using countvectorizer and fed this to classifier model to train the classifier. We used l2 regularization (penalty) and C =1. Smaller value of C gives stronger regularization.

Count vectorizer converts the texts into a matrix of token counts, this implementation produces a sparse matrix of the features. The feature matrix is then supplied to the classifier to train the model. Once the model has been trained, we used 5 fold cross validation method to test the training accuracy. The data will be split into 5 sets, with 4 trainings sets and 1 test set and will be repeated over 5 folds to cross validate the accuracy on the whole data.

Naïve Bayes Classifier:

Many of the movie/book/product classification uses Naïve Bayes classifier. For each known class value, the algorithm calculate probabilities for each attribute, conditional on the class value, Use the product rule to obtain a joint conditional probability for the attributes. Use Bayes rule to derive conditional probabilities for the class variable. Once this has been done for all class values, the class with highest probability is displayed as the output. After processing the tweets, the frequency of each of the relevant words is found and made one feature list. This optimized feature list is then supplied to the classifier to fit the model. A sample data of 500 tweets has been given to the classifier to fit the model and predicted on 1500 tweets. This method gave an accuracy of 0.6454.

Geographical Plotting:

Using the latitude and longitude information obtained from Maps API, we plotted a sample set of users in the static maps for a set of 7 books. Google static maps has been used for plotting the users in the world map.

EXPERIMENTS & RESULTS

We tried to enhance the accuracy by changing the parameters of the Vectorizer, min_df, max_df, binary, and parameters of Logistic regression model, C and regularizations. After the experiments with different parameters, we chose the maximum accurate model and below are our results. We used l2 regularization (penalty) and C =1. Smaller value of C gives stronger regularization.

Accuracy	**Naïve bayes**	**Logistic regression**
	0.75	**0.64**
Logistic Regression	**Precision**	**Recall**
0	**0.99**	**0.94**
2	**0.94**	0.97
4	0.93	0.88

Name of the Book	**Twitter Rating**	**Category**	**Amazon rating**	**Good Reads**
Magnus Chase	5	Excellent	4.7	4.25
Six of Crows	5	Excellent	4.8	4.42
Future Crimes	4.23	Excellent	4.6	4.03
Becoming Nicole	5	Excellent	4.8	4.25
Soundless	0.83	Bad	3.7	3.5
Killing Reagen	4.8	Excellent	4.3	4.06
The Bazaar of Bad Dreams	4.915	Excellent	4.2	4.02

As per our hypothesis, twitter can be used to predict the ratings of the books and ratings obtained using our classifier proves that for majority of the books. . Figure shows that our classification model predicts almost similar ratings to the ratings from amazon and Goodeads except for one book, “Soundless”. Amazon and Goodreads shows average ratings for Soundless and with our model, rating is below average (lower than 2.0). There was no enough volume of tweets for this book and hence it badly affects the rating of the book.

The major limitation we had faced while building this model was the lack of enough negative tweets as well as representative samples. In case of books, people tend to tweet positively. It is difficult to get mixed reviews about books from twitter. Majority of negative tweets on books occur when expectations from a specific author or a famous series of books are not upto the mark. Our hypothesis was twitter can accurately predict the ratings similar to amazon or goodreads since generally more people tend to tweet about books than writing reviews over websites. The data volume for the popular books, collected for a week were over 2000 which indicates that generally many number of people tend to tweet/post in social networking sites. Though many people write reviews in other websites, they also tweet about these books in twitter as well. Locations of a sample set of users are plotted in Google static maps API. For validation of user location, We took world’s 50 famous countries and their cities. Europe and North

America contributed towards the tweets the most. Google Static Map has limitation that it will allow 50 request per user per API key per minute.

RELATED WORK

There has been a large amount of prior research in sentiment analysis, especially in the domain of product reviews, movie reviews, and blogs using naïve bayes, maximum entropy and SVM, but there is not many research on ratings of books using tweets. Rating books using Twitter feeds is different from rating movies or other products. People do not generally read random books and tweet about such a book. Similar works on movie ratings uses naive bayes and the advantage of using this algorithm for this application is that it only requires a small amount of training data to estimate the parameters necessary for classification. We have used Naïve Bayes and Logistic Regression and chose latter for predicting the ratings since latter gave better accuracy. We achieved an accuracy of 0.75 using our model of text classification. Plotting the users across the world was done to analyze the spread of users of each of these book. Due to the limitations of static maps, we couldn't extend our work to more locations

CONCLUSIONS AND FUTURE WORK:

Analyzing social networks like twitter can predict the ratings of books to an extent and people tend to tweet about books in social networking sites more. Processing or cleaning the data collected from twitter is a major part of this project and this can impact the accuracy greatly. Majority of the data collected for books contains positive and neutral tweets. Large share of one class of data can lead to over fitting of the model.

People tweet so many emoticons to show their perspective and emoticons are a better way of predicting the user's feeling towards the book. Most of the users tweet about their favorite characters or about the story than explicitly telling "Great book, good awesome etc. We are planning to extend our work by training the model using emoticons and classify more number of books. Our algorithms classify the overall sentiment of a tweet. The polarity of a tweet may depend on the perspective you are interpreting the tweet from. Using a semantic role labeler may indicate which noun is mainly associated with the verb and the classification would take place accordingly. We are also planning to extend the number of user locations.

REFERENCES:

Twitter: https://about.twitter.com/

CountVectorizer:http://scikitlearn.org/stable/modules/generated/sklearn.feature_extraction.text.CoutVectorizer.html

Logistic Regression:http://www.codeproject.com/Articles/824680/Logistic-Classifier-Overfitting-and-Regularization

Google Maps: https://developers.google.com/places/web-service/?hl=fr