Sentiment Analysis on Twitter and YouTube

Hemanth Reddy Karri
hkarri1@binghamton.edu
Binghamton University
Binghamton, New York, USA

Punit Paresh Jagani
pjagani1@binghamton.edu
Binghamton University
Binghamton, New York, USA

Akhil Parimi
aparimi1@binghamton.edu
Binghamton University
Binghamton, New York, USA

Vijay Kumar Kadamanchi
vkadama1@binghamton.edu
Binghamton University
Binghamton, New York, USA

ABSTRACT

Toxic comments and personal attacks have become increasingly common on social media platforms, online news commenting areas, and many other public venues on the Internet. However, deciding whether or not to "flag" a comment or post is complex and time-consuming. Not only would automating the process of detecting abuse in comments save website moderators time, but it would also promote user safety and improve online discussions. We're particularly interested in tweets and YouTube comments that contain any harsh or toxic language. We want to identify tweets and YouTube comments that contain toxic phrases and label them as negative using the data we've acquired in a certain time frame.

KEYWORDS

Machine learning models for classification, text mining, text analysis, data analysis, data visualization, MongoDB, Twitter API, YouTube API

ACM Reference Format:
Hemanth Reddy Karri, Punit Paresh Jagani, Akhil Parimi, and Vijay Kumar Kadamanchi. 2018. Sentiment Analysis on Twitter and YouTube. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The issue of trolls and spammers is getting more prominent as discussions move more and more to internet forums. Manually moderating comments and discussion forums is time-consuming, and organizations are compelled to use contractual or outside moderators to deal with the high volume of comments. We're looking for hate speech on Twitter and YouTube. We consider a tweet or a comment to be hate speech if it incorporates racist or sexist comments. The main objective is to classify negative tweets and comments from other tweets and comments.We'll be collecting real world data from popular media channels like Twitter and YouTube for this project. We'll use the data from these two sources with a defined time frame to discover and highlight any negative sensing content. We also aim to visualize the classified data of Twitter and YouTube.

2 DATA ACQUISITION

The raw tweet data is obtained for Twitter using the python module "tweetstream," which provides a framework for simple Twitter streaming API. The data for YouTube is acquired using the python library "googleapiclient".

2.1 Twitter's Streaming API

We use TweetStream python module that can be used to get tweets from Twitter's streaming API. There are two ways to obtain tweets with Tweetstream: SampleStream and FilterStream. SampleStream merely provides a short, random sample of all tweets that are being posted in real time. FilterStream sends tweets that meet a set of criteria. It has the ability to filter tweets based on three criteria:

- Specific keyword(s) to track/search for in the tweets.
- Specific Twitter user(s) according to their user-id's.
- Tweets originating from specific location(s) (only for geo-tagged tweets).

As we don't have any such constraints for our purpose, we'll use SampleStream mode.

2.2 YouTube API

The Google API Client Library for Python is designed for Python client-applications. It provides easy and flexible access to a variety of Google APIs. All API calls must use one of two types of access: simple or authorized. We utilize the build() function to generate a service object whether we're utilizing simple or authorized API access. Every collection defined by the API is represented by a function in this object. We collect all of the comments of random videos using this method and analyze them later.

3 EXPLORATORY DATA ANALYSIS

Cleaning the text data is important because it prepares the raw text for mining, making it easier to extract information from the text and apply machine learning algorithms to it. If we omit this stage, the likelihood of working with noisy and inconsistent data increases exponentially. The goal of this phase is to remove noise that isn't important in determining the sentiment of tweets and comments.

Conference'17, July 2017, Washington, DC, USA
© 2018 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

After the data is clean, We will design a classifier which accurately classifies negative tweets and comments. Basically we will use either contextual or general classifier. The classification approach generally followed in this domain is a two-step approach. The first step is to determine whether a tweet or a phrase is objective or subjective. Following that, we perform Polarity Classification, which involves using machine learning algorithms to determine whether or not the tweet is negative. We can then analyze the data and visualize it in a bar graph once all of the data has been classified.

4 PROJECT FLOW

Using Python modules TweetStream and GoogleAPIClient, we begin by retrieving real-world data from our two main data sources: Twitter and YouTube. We clean the data once it has been stored in MongoDB to remove all of the noise. Once we have noise-free data, we use machine learning algorithms to develop a classifier (either contextual or general). We now apply the techniques of tokenization, lowercase conversion, and stop-words removal to extract valuable features from the data contained in MongoDB.
Tokenization is the act of breaking down a continuous stream of text into words, symbols, and other significant parts known as "tokens."
Lowercase Conversion: Normalizing a tweet by converting it to lowercase makes it easy to compare it to an English dictionary.
Stop-words removal: Stop words are a group of extremely common terms that, when utilized in a text, provide no new information and are hence considered useless. Examples include "a", "an", "the", "he", "she", "by", "on", etc.
We can then use this data to train the agent once the data matches our expectations. We give the agent points for each successful recognition, and we deduct points if the recognition fails. We transmit our raw data to the agent after it has been trained and tested, and the agent will assess if it is sentimentally negative. After that, we'll evaluate and visualize our findings in a bar graph.

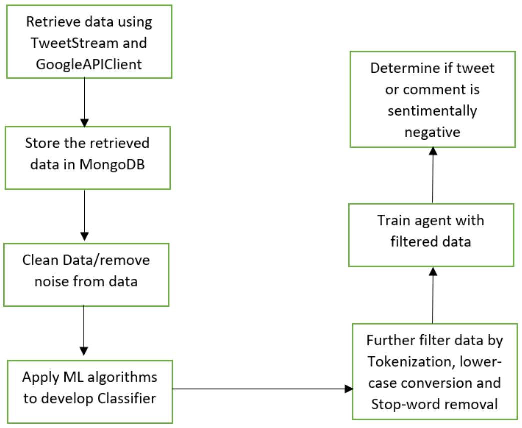
Retrieve data using TweetStream and GoogleAPIClient
Store the retrieved data in MongoDB
Clean Data/remove noise from data
Apply ML algorithms to develop Classifier
Further filter data by Tokenization, lower-case conversion and Stop-word removal
Train agent with filtered data
Determine if tweet or comment is sentimentally negative

Figure 1: An image of a galaxy

5 IMPLEMENTATION

To fetch the data from the Twitter streaming API, we used "http client" for authorization using bearer token. Next we used "URIBuilder" class instance by sending API URL as an argument. Then we connected to mongoDB using "mongoclient" class driver and stored each filtered response data in the database. Next, to fetch the comments data from YouTube API, we created a service "getservice()" which returns "httpTranport" which is similar to a http request. This service will return comment Threads which has around 10 comments stored in it. We then update this on each request by setting the "developer key". Then again to connect to mongoDB we used "mongoclient" class driver and stored each filtered response data in the database.

6 DATA COLLECTION ESTIMATES

We are collecting around 1M tweets per day which means collecting around 41,667 tweets per hour. Based on the rate of our data collection, We collected a total of 7162195 tweets (approx 7.1 million tweets) and 1839114 YouTube comments (approx 1.8 million comments). we used this data to generate meaningful data which can be used to run our analysis.

7 CHANGES COMPARED TO PROPOSAL

There are no changes compared to the proposal as of now

8 REFERENCES

Rawan Fahad Alhujaili and Wael M.S. Yafooz(2021). Sentiment Analysis for Youtube Videos with User Comments: Review.

Albert Biffet and Eibe Frank(2010). Sentiment Knowledge Discovery in Twitter Streaming Data. Discovery Science, Lecture Notes in Computer Science, Volume 6332/2010, 1-15, DOI: 10.1007/978-3-642-16184-1

Alec Go, Richa Bhayani and Lei Huang(2009). Twitter Sentiment Classification using Distant Supervision. Project Technical Report, Stanford University.

Mohd Majid Akhtar(2019). Sentiment Analysis on YouTube Comments: A brief study. Jamia Milia Islamia