HATE SPEECH DETECTION OF STREAMING TWEETS

Prachi Naik
MS2022019

ABSTRACT: Since last few years, Twitter is a social platform which is used by people from different cultural backgrounds, political backgrounds, and regions. This platform is used by the people to convey their opinions or views about different things which might upset some other community of people thus creating "cyber" conflicts between these people. My aim here is to reduce such conflict by detecting hate speech in incoming stream of tweets.

1. INTRODUCTION

Hate speech is defined as "speech expressing hatred for a person or group of people". This speech is usually intended to insult, offend or intimidate a person because of some trait. In this use case, we have to build a streaming application which will identify the polarity of incoming stream of tweets. For this, we will compare different NLP tools and we will see the performance of each tool.

2. APPROACH

Given an incoming stream of tweets, we will pass it through a pretrained model and it will generate positive, negative or neutral polarity.

2.1. Twitter developer API access

We need Twitter's developer API access for the incoming stream of tweets. We requested for "Elevated" access for developer API since "Essential" access allows 500K Tweets/month and elevated access allows 2M Tweets/month.
Old authentication method via endpoint V1.1 is depreciated this year. So, we need to authenticate connection via endpoint V2 only. So, we developed code using Python and PySpark, where we authenticate via V2 endpoint and collect the tweets into RDD.
We can use either Bearer token or API key tokens and access token.

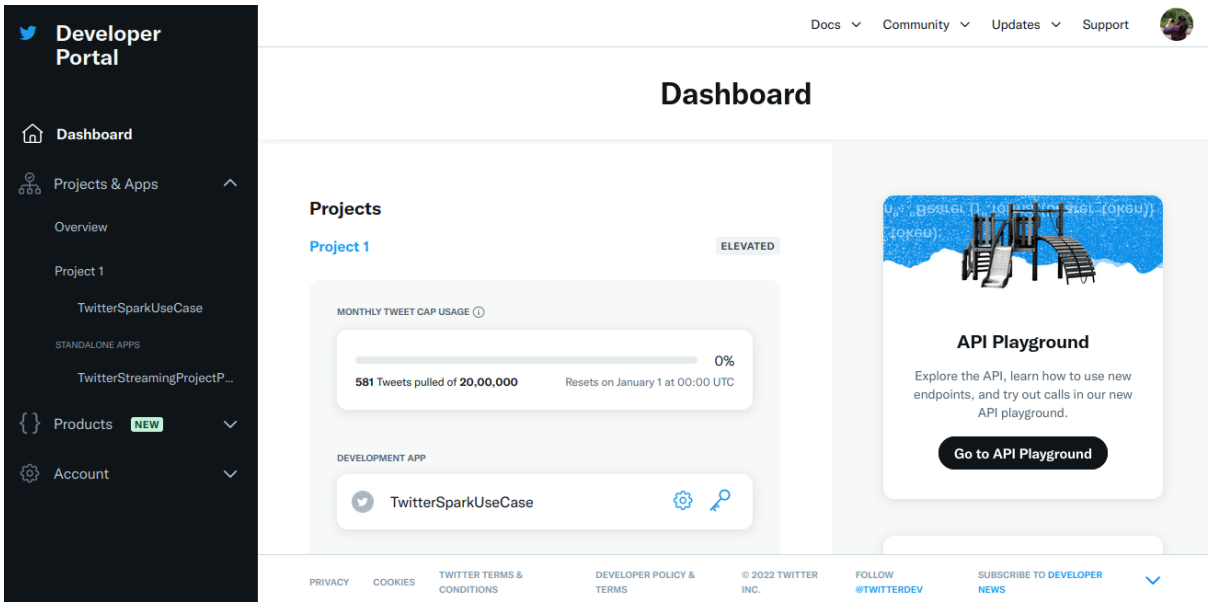

Fig.1. Twitter Developer API dashboard

2.2. Pre-processing of Data

We pre-process the tweets so we can have only the clean text of the tweet. In each batch, we receive many tweets from the Twitter API and split the tweets at the string t_end. Then, we remove the empty rows and apply regular expressions to clean up the tweet text. In more detail; we remove the links (https://), the usernames (@), the hashtags (#), the string that shows if the current tweet is a retweet (RT), and the character ':'.

2.3. Apache Spark, PySpark and RDD

Apache Spark is a data processing framework that can quickly perform processing tasks on very large data sets, and can also distribute data processing tasks across multiple computers, either on its own or in tandem with other distributed computing tools.

At the heart of Apache Spark is the concept of the Resilient Distributed Dataset (RDD), a programming abstraction that represents an immutable collection of objects that can be split across a computing cluster. Operations on the RDDs can also be split across the cluster and executed in a parallel batch process, leading to fast and scalable parallel processing.

In this project, we use PySpark which is Python API for Apache Spark.

We first create a Spark Session which is an entry point to underlying PySpark functionality. Then we will fetch tweets in JSON format using Twitter developer API. We iterate through the text format of tweets and check their polarity. We save these tweets in a list.

Using SparkContext, we enter the spark functionality. Using the list of tweets, we create an RDD using parallelize() method. RDDs in PySpark are collection of partitions (basic units of parallelism). PySpark creates partitions that are equal to the number of CPU cores in the machine. Data of each partition resides in single machine and PySpark creates task for each partition.

2.4. TextBlob

We use TextBlob for detecting negativity in tweets. TextBlob is a python library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. The output of TextBlob is polarity and subjectivity. Polarity score lies between -1 to 1. -1 identifies most negative words and +1 identifies most positive words. Subjectivity score lies between (0 and 1), It shows the amount of personal opinion, If a sentence has high subjectivity i.e. close to 1. We do not need subjectivity in our project.

As TextBlob is a Lexicon-based sentiment analyzer. It has some predefined rules or we can say word and weight dictionary, where it has some scores that help to calculate a sentence's polarity. That's why the Lexicon-based sentiment analyzers are also called "Rule-based sentiment analyzers".

NLP Steps done by TextBlob for Sentiment Analysis

1. Lemmatization/Stemming - Shorten words to their root stem – e.g. removes -ing, -ion, etc
2. Lowercasing words
3. Cleaning the data - Remove special characters
4. Remove stop words, punctuation, or unwanted tokens e.g. The, was , and
5. Tokenization - create a bag of words
6. Classification Based on Polarity or Subjectivity

For the lexical approach, a dictionary is prepared to store the polarity values of lexicons. For calculating polarity of a text, polarity score of each word of the text, if present in the dictionary, is added to get an 'overall polarity score'. For example, if a lexicon matches a word marked as positive in the dictionary, then the total polarity score of the text is increased. If the overall polarity score of a text is positive, then that text is classified as positive, otherwise it is classified as negative. Though this approach seems very basic, variants of this

lexical approach have been reported to have considerably high accuracy.

2.5 Flowchart

Flow of the project is as per given in Fig.2. In case of counts of tweets, we give input of number of tweets we want to process.

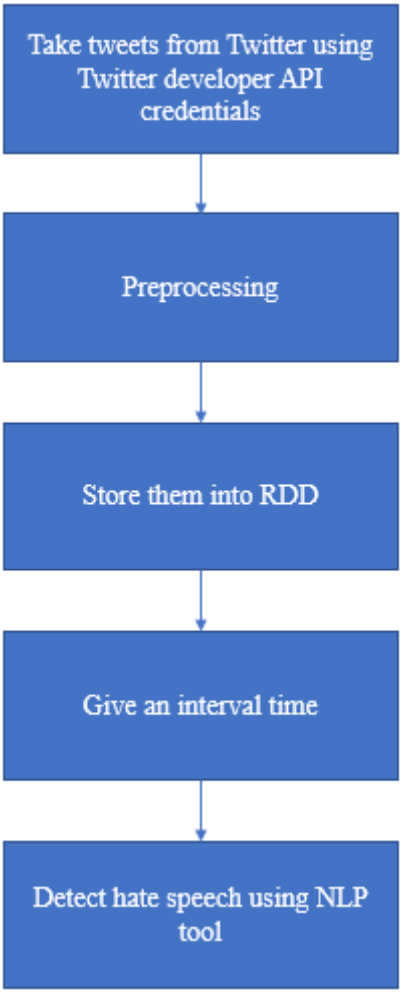

Fig.2. Flowchart

3. EVALUATION AND RESULTS

System processes 200 tweets per minute. We have compared other sentiment analysis tools as well such as Vader and Flair. Vader and TextBlob perform same in case of negative sentiment. Different factors such as capital lettered words, punctuation or emojis.

TEXTBLOB	Precision	Recall	F1 score
Negative	0.67	0.48	0.56
Positive	0.25	0.59	0.35
Neutral	0.26	0.15	0.19

4. CONCLUSION

Since the fast growth of the internet and its applications, hate speech detection is an important and interesting area for research. Such streaming application will give us real-time detection of any hatred or offensive speech so that we can take next steps to stop such tweets.

5. REFERENCES

(PDF) Hate Speech on Twitter A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection (researchgate.net)

https://spark.apache.org/docs/latest/streaming-programming-guide.html

https://textblob.readthedocs.io/en/dev/

https://spark.apache.org/docs/latest/api/python/

https://www.researchgate.net/publication/352393003_Determining_the_Efficiency_of_Drugs_Under_Special_Conditions_From_Users'_Reviews_on_Healthcare_Web_Forums