Twitter Sentimental Analysis

Mini-Project report for the subject Big Data Analytics (Computational Lab)

by

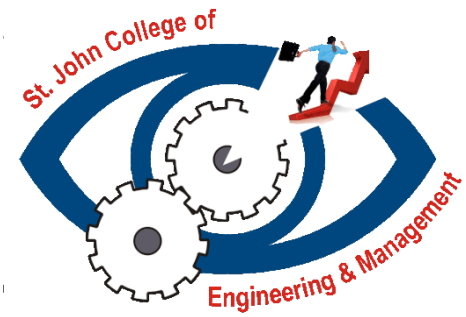
Singh Ankita Dharmraj EU1162043

Singh Shubham Santosh EU1162097

Tiwari Suraj Jitendra EU1162062

Under the guidance of

Mrs. Aditi Raut



Department of Computer Engineering

St. John College of Engineering and Management

University of Mumbai

2019-2020

1. Introduction

In the past few years, there has been a huge growth in the use of microblogging platforms such as Twitter. Spurred by that growth, companies and media organizations are increasingly seeking ways to mine Twitter for information about what people think and feel about their products and services. Companies such as Twitratr (twitrratr.com), tweetfeel (www.tweetfeel.com), and Social Mention (www.socialmention.com) are just a few who advertise Twitter sentiment analysis as one of their services.

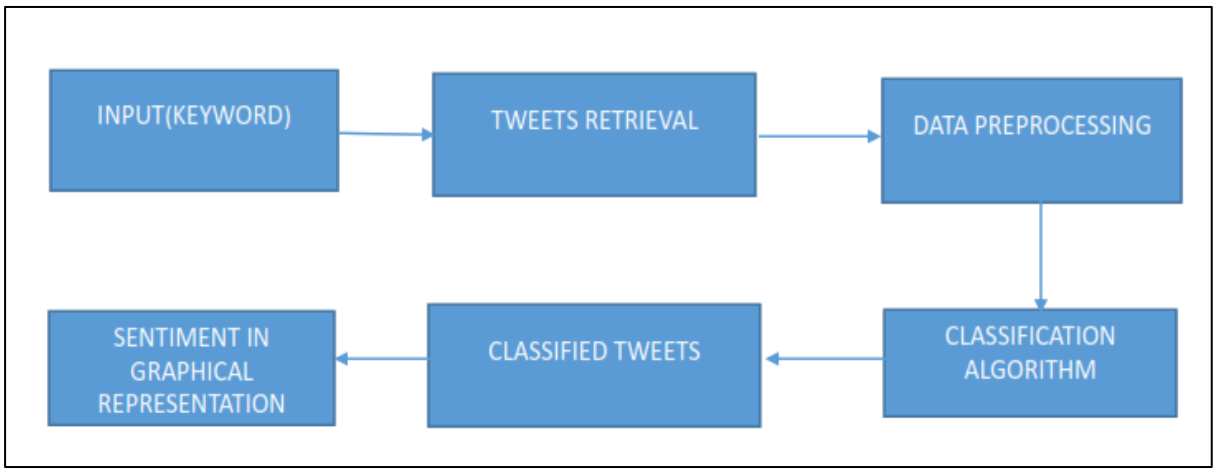
While there has been a fair amount of research on how sentiments are expressed in genres such as online reviews and news articles, how sentiments are expressed given the informal language and message-length constraints of microblogging has been much less studied. Features such as automatic part-of-speech tags and resources such as sentiment lexicons have proved useful for sentiment analysis in other domains.

It is not an exaggeration to say that people tweet about anything and everything. Therefore, to be able to build systems to mine Twitter sentiment about any given topic, we need a method for quickly identifying data that can be used for training.

Many traditional approaches in sentiment analysis uses the bag of words method. The bag of words technique does not consider language morphology, and it could incorrectly classify two phrases of having the same meaning because it could have the same bag of words . The relationship between the collection of words is considered instead of the relationship between individual words. When determining the overall sentiment, the sentiment of each word is determined and combined using a function .

Sentiment analysis refers to the broad area of natural language processing which deals with the computational study of opinions, sentiments and emotions expressed in text. Sentiment Analysis (SA) or Opinion Mining (OM) aims at learning people’s opinions, attitudes and emotions towards an entity.

2. Design



Input(KeyWord):

Data in the form of raw tweets is acquired by using the Python library “tweepy” which provides a package for simple twitter streaming API . This API allows two modes of accessing tweets: SampleStream and FilterStream. SampleStream simply delivers a small, random sample of all the tweets streaming at a real time. FilterStream delivers tweet which match a certain criteria.

Tweets Retrieval

Since human labelling is an expensive process we further filter out the tweets to be labelled so that we have the greatest amount of variation in tweets without the loss of generality.

Data Pre-processing:

Data Pre-processing consists of three steps:

Tokenization:

It is the process of breaking a stream of text up into words, symbols and other meaningful elements called “tokens”.

Normalization:

For the normalization process, the presence of abbreviations within a tweet is noted and then abbreviations are replaced by their actual meaning (e.g., BRB − > be right back).

Part-of-speech:

POS-Tagging is the process of assigning a tag to each word in the sentence as to which grammatical part of speech that word belongs to, i.e. noun, verb, adjective, adverb, coordinating conjunction etc.

Classification Algorithm:

Let's build a sentiment analysis of Twitter data to show how you might integrate an algorithm like this into your applications. We'll first start by choosing a topic, then we will gather tweets with that keyword and perform sentiment analysis on those tweets. We'll end up with an overall impression of whether people view the topic positively or not.

Classified Tweets:

We labelled the tweets in three classes according to sentiments expressed/observed in the tweets: positive, negative and neutral .

Positive:

If the entire tweet has a positive/happy/excited/joyful attitude or if something is mentioned with positive connotations.

Negative:

If the entire tweet has a negative/sad/displeased attitude if something is mentioned with negative connotations.

Neutral:

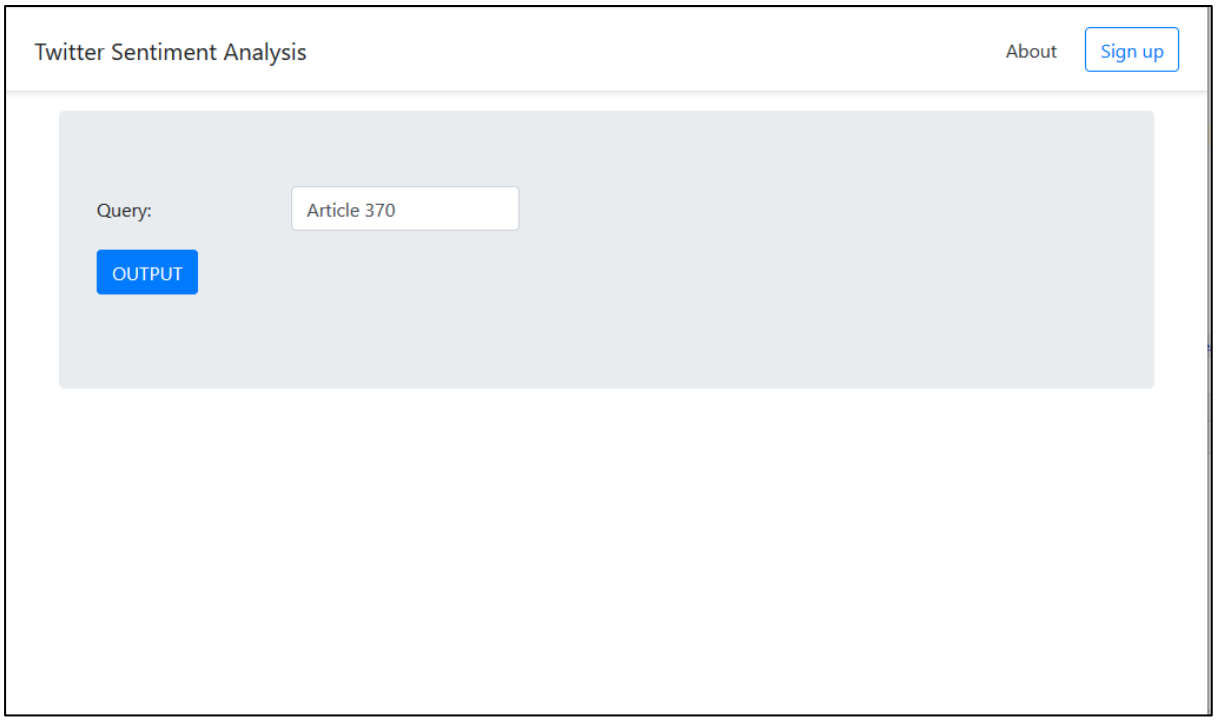
If the creator of tweet expresses no personal sentiment/opinion in the tweet and merely transmits information.

3. Results & Discussion

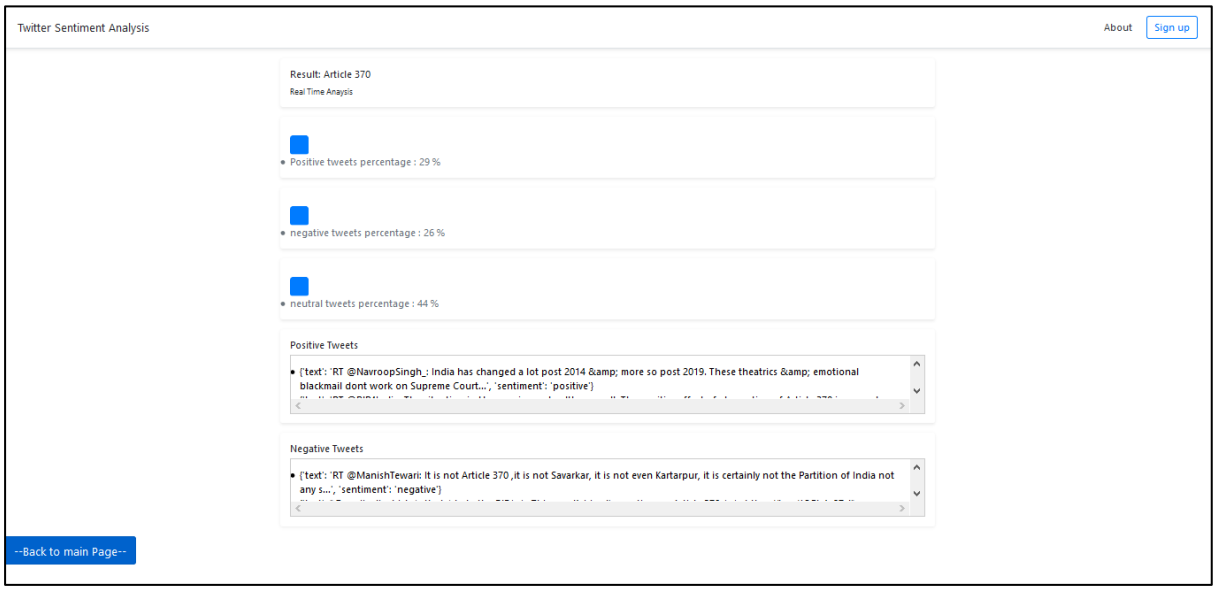
We will first present our results for the positive / negative classifications. These results act as the first step of our classification approach. We only use the short-listed features for both of these results. This means that for the positive / negative classification we have 3 features. For these results we use the Naïve Bayes classification algorithm, because that is the algorithm, we are employing in our actual classification approach at the first step.

The above is the GUI (Web-Application) for our project. It is being designed using Flask technology.

Finally, we conclude that our classification approach provides improvement in accuracy by using even the simplest features and small amount of data set. However, there are still a number of things we would like to consider as future work which we mention in the next section



Input to the System



Sentimental analysis Output

4.Conclusion

The task of sentiment analysis, especially in the domain of micro-blogging, is still in the developing stage and far from complete. So, we propose a couple of ideas which we feel are worth exploring in the future and may result in further improved performance.

Right now, we have worked with only the very simplest unigram models; we can improve those models by adding extra information like closeness of the word with a negation word. We could specify a window prior to the word (a window could for example be of 2 or 3 words) under consideration and the effect of negation may be incorporated into the model if it lies within that window. The closer the negation word is to the unigram word whose prior polarity is to be calculated, the more it should affect the polarity.

Right now, we are exploring Parts of Speech separate from the unigram models, we can try to incorporate POS information within our unigram models in future. So say instead of calculating a single probability for each word like P(word | obj) we could instead have multiple probabilities for each according to the Part of Speech the word belongs to.

One more feature we that is worth exploring is whether the information about relative position of word in a tweet has any effect on the performance of the classifier. In this research we are focussing on general sentiment analysis. There is potential of work in the field of sentiment analysis with partially known context. So, we can attempt to perform separate sentiment analysis on tweets that only belong to one of these classes (i.e. the training data would not be general but specific to one of these categories) and compare the results we get if we apply general sentiment analysis on it instead.

Also, while performing the analysis we restrict the sample size to 200 tweets due to less availability of resources and faster result. However, we can increase the sample size to a desirable number if enough resources are available.