SENTIMENTAL ANALYSIS
NATURAL LANGUAGE PROCESSING

US President Election 2016

Sentimental Analysis

Machine Learning Approach

classifies the text using classification algorithm

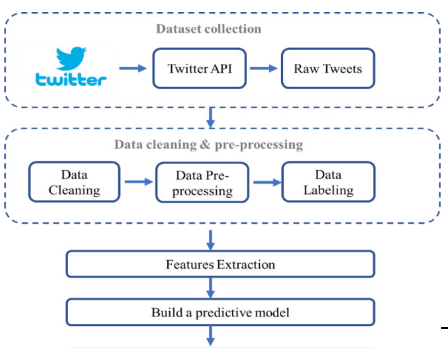
Lexicon based approach

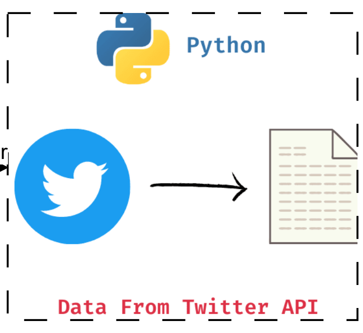
uses sentiment dictionary with opinion words and match them with data to determine polarity

Determine people's attitudes towards political figures
Analyze sentiment polarity and subjectivity of tweets

How Data has been collected?

Debate — First Presidential Address — Twitter





Twitter is used to acquire a large diverse dataset representing the current public opinions of the candidates.

The collected tweets are analysed using lexicon based approach to determine the sentiments of public

counts of positive, negative and neutral tweets for **Trump, Hilary and Bernie.**

Models Implemented

Proposed Approach

Retrieval module

Tweets collected through Twitter API. They are then filtered and classified based on the sentiments.

Preprocessing module

Removing hashtags, URLs, @tags, emoticons, trailing whitespaces and newline characters to extract meaningfull data

Libraries Used **NLTK, TextBlob, Twython**

Twython used for purpose of User information found in Twitter API which offers support for User authenticated calls and application authenticated calls.

TextBlob provides a simple API for part-of-speech tagging, noun phrase extraction, sentiment analysis, classification & translation.

NLTK[10] provides with text-processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers

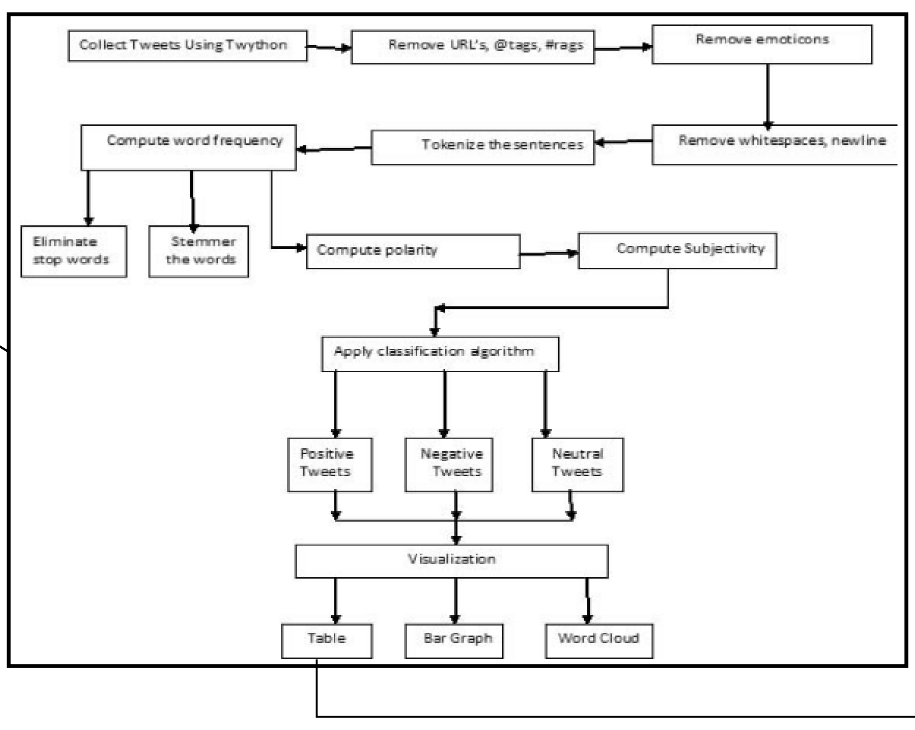
Analysis module

```
i.    Initialize positive, negative and neutral tweet count to zero.
          positive_tweet_count=0; negative_tweet_count=0;
          neutral_tweet_count=0;
ii.   Also initialize Avg_polarity=0, Avg_subjectivity=0;
iii.  Fetch the stored tweets from the file
      a.  For each line in the file do
      b.  Sent1 = TextBlob(line);
      c.  Calculate polarity and subjectivity as
              Polarity = Sent1.sentiment.polarity;
              Subjectivity=Sent1.sentiment.subjectivity;
iv.   If Polarity > 0, increment positive_tweet_count;
v.    If Polarity < 0, increment negative_tweet_count;
vi.   If Polarity = 0, increment neutral_tweet_count;
vii.  Calculate Average_polarity and Average_subjectivity as
      a.  Avg_polarity=Sum (Polarity)/len(Polarity);
      b.  Avg_subjectivity=Sum
          (subjectivity)/len(subjectivity);
viii. Calculate percentage of positive, negative and neutral tweets.
```

Result visualization module

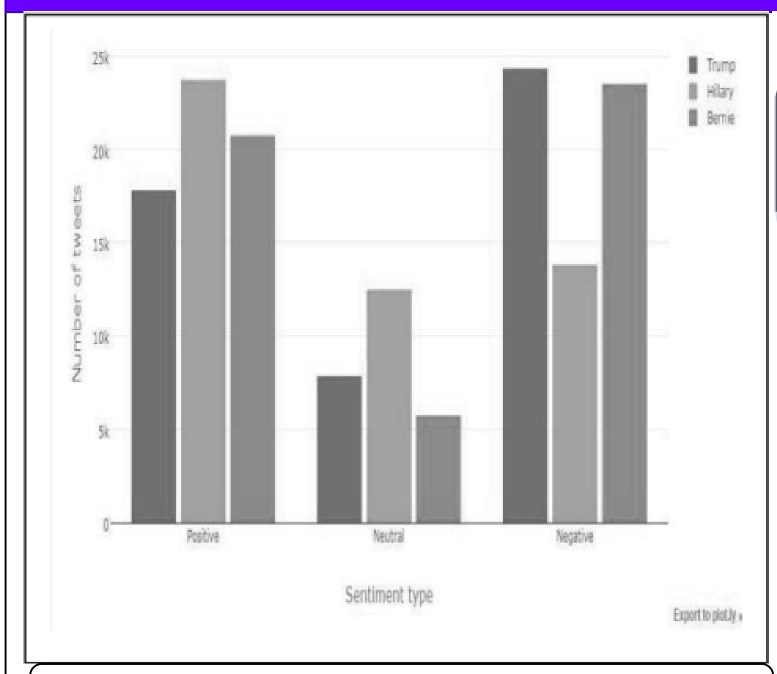
The final step of this process is to take in the filtered tweets and generate word cloud, group bar graph and table to visualize the results

Architecture



Name	Donald Trump	Hillary Clinton	Bernie Sanders
No. of positive comments	17812	23716	20748
No. of Neutral comments	7853	12475	5741
No. of Negative comments	24335	13809	23511
% Positive	35.624	47.432	41.496
% Negative	15.706	24.95	11.48
% Negative	48.67	27.618	47.022
Average Polarity	-0.0073	0.0482	0.0282
Average Subjectivity	0.278	0.2809	0.3138

Observations & Results



Comparison of sentiment type for different candidates.

The graph depicts that Hilary got the highest number of positive and neutral tweets, whereas Trump got the highest negative and lowest positive tweets. The frequent word will appear prominent in the Word Cloud, representing the word cloud as Hillary by processed tweets file. Similarly, word clouds for Trump and Bernie can also be obtained from other datasets.

Word Cloud for tweets of Hillary

Conclusion & Recommendation

In this paper, we introduce a lexicon-based sentiment analysis system designed to categorize tweets according to their sentiment, focusing on tweets from the **2016** *US presidential elections*. This sentiment classification relies on polarity and subjectivity metrics, which indicate whether users express a positive, negative, or neutral stance towards specific election candidates. This approach allows us to provide a comparative analysis of the leading candidates in the 2016 presidential elections.

In the future, we may consider delving into creating a multilingual sentiment classification system capable of encompassing tweets in diverse languages. Additionally, we could explore the integration of an automated chatbot that interfaces with the Twitter API, executing the sentiment analysis model in real-time and furnishing us with up-to-the-minute sentiment insights.