Twitter Election Prediction

Twitter Developer Api and Google Natural Language Processing Applications
Molly Housego

Background

Boston Mayoral Race

- This november Boston will hold a historic election where a new mayor will be selected after almost 7 years of incumbency
- My thoughts were to utilize the developer apis to see if sentiment from twitter could be an indicator of the sentiment of the general populous and therefore reflect the vote outcome
- In this study I ran sentiment analysis on all tweets containing the subjects Michelle Wu and Annissa Essaibi George, compared their average sentiments and how passionate the authors of the tweets were when discussing each candidate
- This Social media analyzer could be used by campaigns to monitor twitter performance and potentially predict election results.

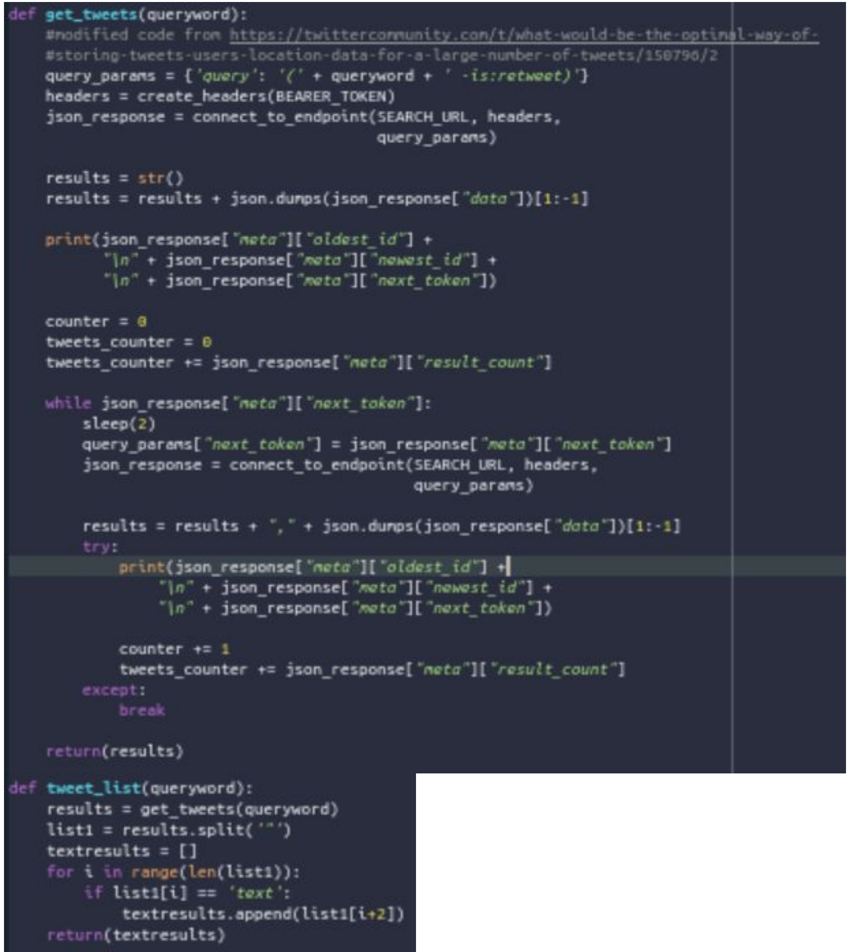
Twitter and Google APIs

- Due to limited developer permissions, I could only access tweets from the last seven days. Ideally the analysis would be performed over a longer period, but for this study i assumed seven days was a representative sample of any other length of time.
- The scoring system for sentiment is provided by the google nlp where score is the overall emotion of the text ranging from -1 to 1. A negative score is indicative of negative emotion and vise versa. A zero score is considered neutral. Sentiment magnitude is considered intensity of the text, for example: “I feel bad” and “I feel bad!” will have the same sore but the latter will have a larger magnitude

Program

Get Tweets based on Query Word

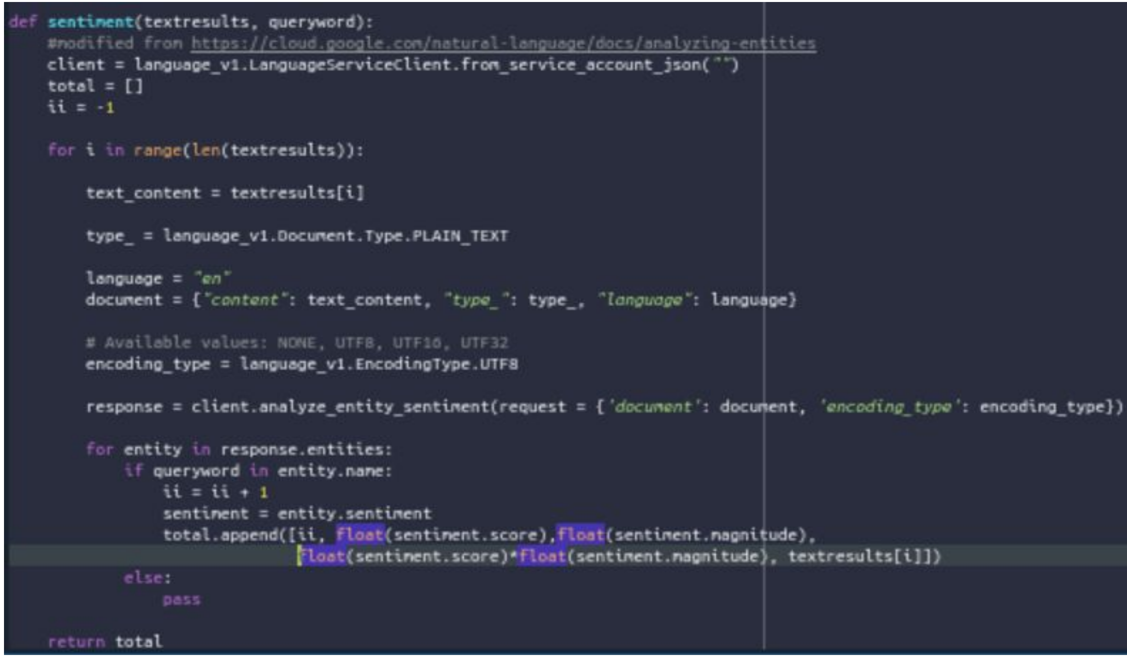
- Twitter has a paging issue where the response limit is restricted to 10 but search tokens are created for up to 300 tweet results. The get_tweets function loops through the tokens and returns all results in sets of 10.
- The tweet_list function takes the output of the developer results and parses out tweet text content only



```python
def get_tweets(queryword):
    #modified code from https://twittercommunity.com/t/what-would-be-the-optimal-way-of-
    #storing-tweets-users-location-data-for-a-large-number-of-tweets/150796/2
    query_params = {'query': '(' + queryword + ' -is:retweet)'}
    headers = create_headers(BEARER_TOKEN)
    json_response = connect_to_endpoint(SEARCH_URL, headers,
                                        query_params)

    results = str()
    results = results + json.dumps(json_response["data"])[1:-1]

    print(json_response["meta"]["oldest_id"] +
          "\n" + json_response["meta"]["newest_id"] +
          "\n" + json_response["meta"]["next_token"])

    counter = 0
    tweets_counter = 0
    tweets_counter += json_response["meta"]["result_count"]

    while json_response["meta"]["next_token"]:
        sleep(2)
        query_params["next_token"] = json_response["meta"]["next_token"]
        json_response = connect_to_endpoint(SEARCH_URL, headers,
                                            query_params)

        results = results + "," + json.dumps(json_response["data"])[1:-1]
        try:
            print(json_response["meta"]["oldest_id"] +
                  "\n" + json_response["meta"]["newest_id"] +
                  "\n" + json_response["meta"]["next_token"])

            counter += 1
            tweets_counter += json_response["meta"]["result_count"]
        except:
            break

    return(results)

def tweet_list(queryword):
    results = get_tweets(queryword)
    list1 = results.split('"')
    textresults = []
    for i in range(len(list1)):
        if list1[i] == 'text':
            textresults.append(list1[i+2])
    return(textresults)
```

Sentiment Analysis

- The sentiment function loops through the tweet list and returns the sentiment of each list item focusing on the sentiment surrounding the candidates name. This is called entity sentiment. This is because analysis on an entire negative tweet that is implying something positive about a candidate will produce a false negative result.
- The function returns the sentiment score, magnitude, weighted score (score multiplied by magnitude) and corresponding tweet text in a nested list.
- From here I use python to analyze and plot results



```
def sentiment(textresults, queryword):
    #modified from https://cloud.google.com/natural-language/docs/analyzing-entities
    client = language_v1.LanguageServiceClient.from_service_account_json("")
    total = []
    ii = -1

    for i in range(len(textresults)):

        text_content = textresults[i]

        type_ = language_v1.Document.Type.PLAIN_TEXT

        language = "en"
        document = {"content": text_content, "type_": type_, "language": language}

        # Available values: NONE, UTF8, UTF16, UTF32
        encoding_type = language_v1.EncodingType.UTF8

        response = client.analyze_entity_sentiment(request = {'document': document, 'encoding_type': encoding_type})

        for entity in response.entities:
            if queryword in entity.name:
                ii = ii + 1
                sentiment = entity.sentiment
                total.append([ii, float(sentiment.score),float(sentiment.magnitude),
                              float(sentiment.score)*float(sentiment.magnitude), textresults[i]])
            else:
                pass

    return total
```

Results

Sample Results

Neutral

Subject: Michelle Wu and Annissa Essaibi George Score: 0   Magnitude: 0   (both subjects)

“Michelle Wu and Annissa Essaibi George will face off for the mayorship of Boston, ending nearly a century of Irish American and Italian American mayors. #Boston #IrishAmerican #BostonMayor”

Strong Positive

Subject: Michelle Wu Score: 0.40   Magnitude: 0.80

“I believe Michelle’s policy platform is the best situated to make positive changes in these areas. Please join me in supporting Michelle Wu for Mayor. Reach out if you’d like to discuss more reasons why I’m supporting her, or if you want to get involved in her campaign!”

Strong Negative

Subject: Annissa Essaibi George Score: -0.89   Magnitude: 0.89

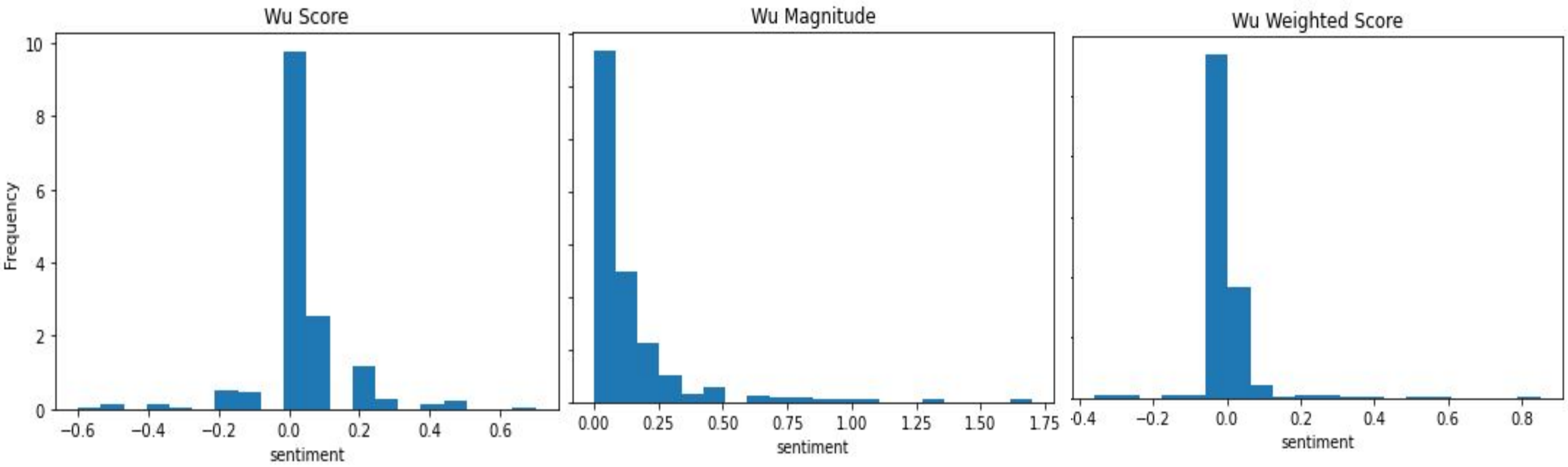
“This nativist sentiment is disgusting and has long been tinged with racism. One of the worst elements of Boston politics. Really disappointing to see Essaibi George sink there.”

Michelle Wu

Total Available Tweets : 280

Average Score: 0.0303 Mostly neutral Positive leaning score

Average Weighted Score: 0.0125 More Neutral Positive leaning score

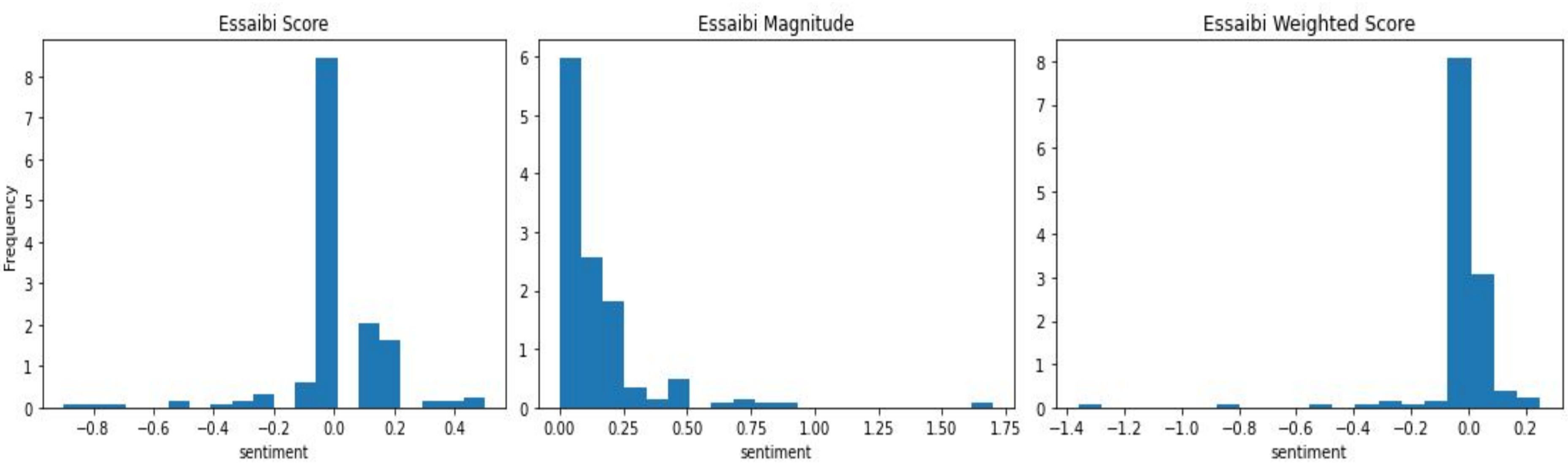


Annissa Essaibi George

Total Available Tweets: 169

Average Score: 0.019   Mostly neutral Positive leaning score

Average Weighted Score: -0.0101   More Neutral Negative leaning score



Conclusions

- The vast majority of tweets for both candidates are neutral ie. stating facts or unbiased comments
- Wu had 65% more analyzable tweets than Essaibi in the last seven days
- Tweeters with strong feelings about Wu tend to lean positive whereas tweeters with strong feelings about Essaibi tend to lean negative
- Both candidates have more positive leaning tweets than negative leaning, with Wu receiving 60% more positive tweets than Essaibi

Based on the analyzable Tweets from the last seven days, there is more positive sentiment about the Boston Mayoral candidate Michelle Wu, indicating that she may be more likely to win election