<u>Indexing & Searching for Hot Topics with ElasticSearch, Apache Spark, Kafka, Count-Min, and Heavy Hitters</u>

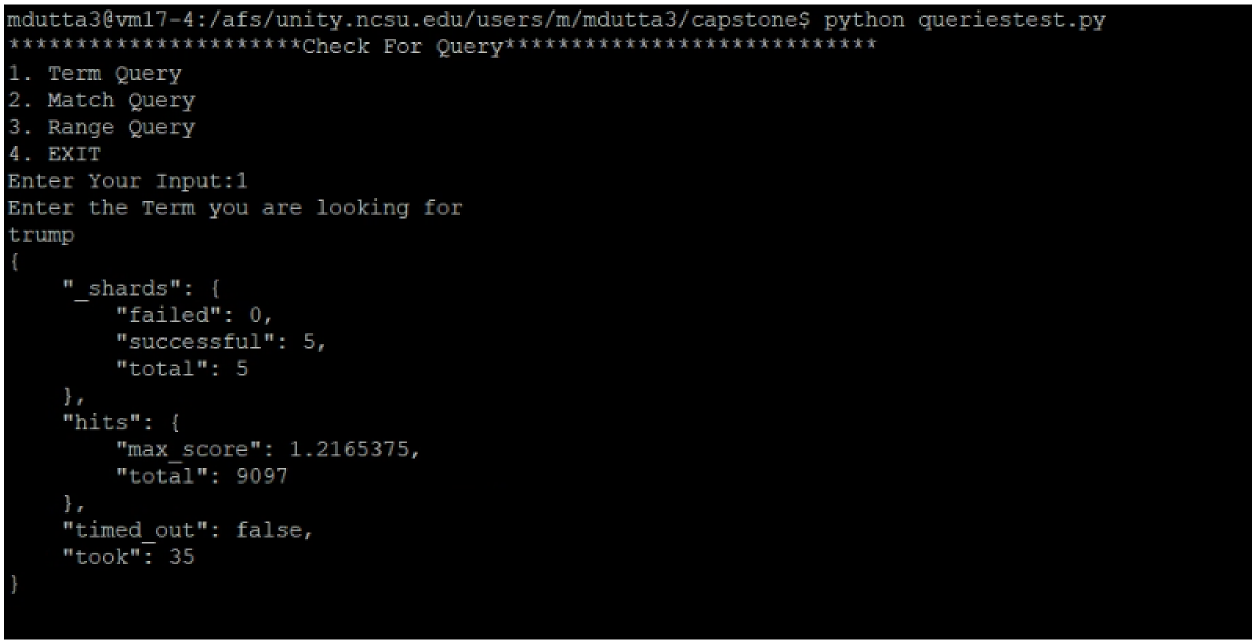
<u>Twitter Streaming</u>

We are processing streaming data in real time and this data is considered as our dataset for this project. We are using tweepy and Kafka API. In the Elastic Search part of this project, the JSON's received are pushed into Kafka and then indexed into Elastic search directly after which they are queried. The CountMinSketch part filters out only the text from the JSONs received and pushes it to Kafka.

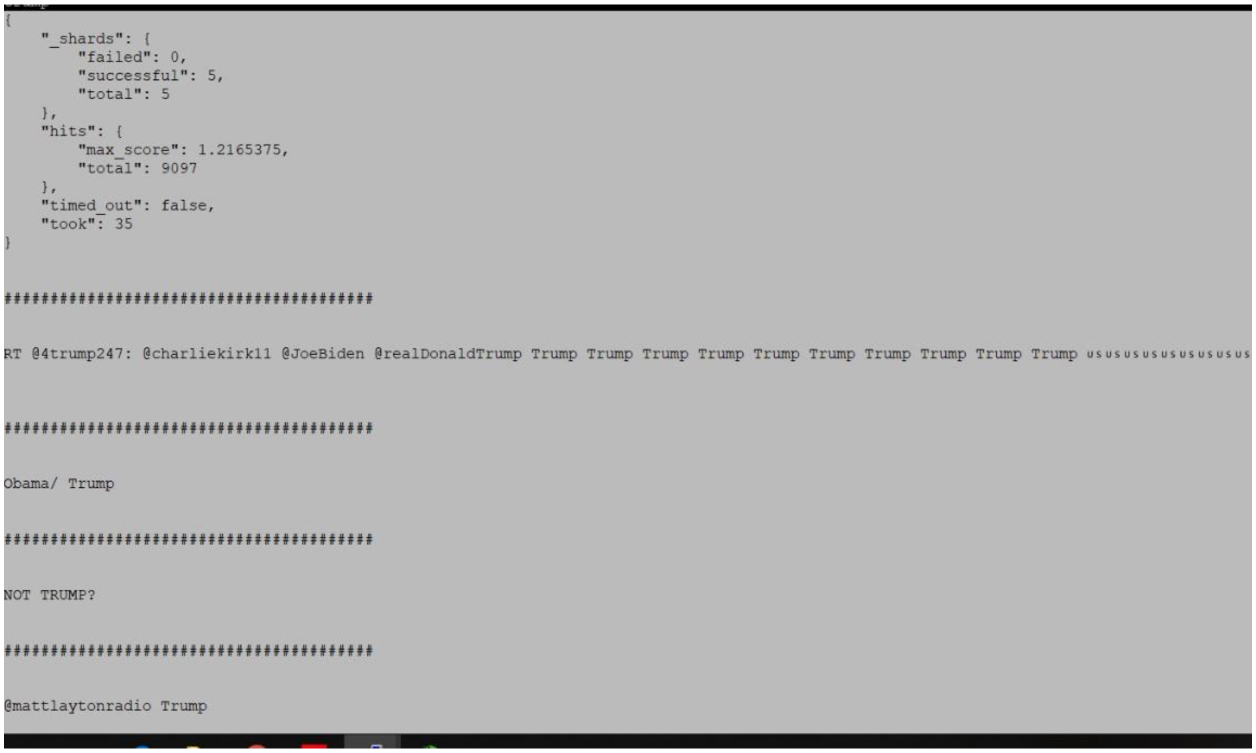
<u>SUMMARY OF RESULTS</u>

We implemented the following queries:

1. **Term Query:** Searches for the exact word "Trump".

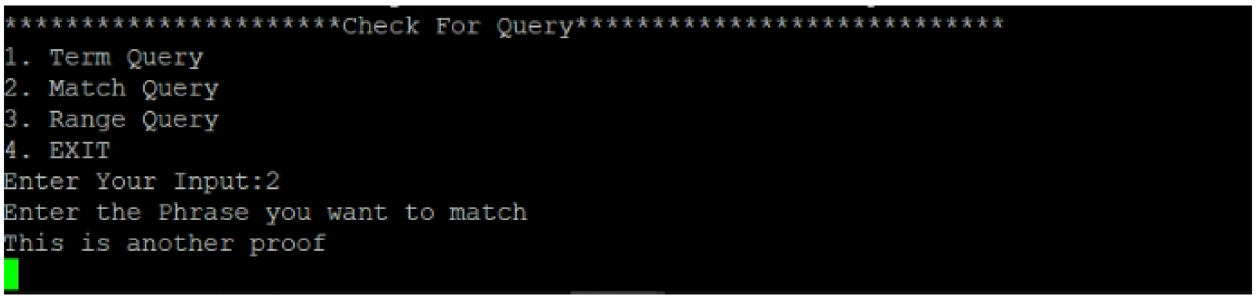


```
mdutta3@vm17-4:/afs/unity.ncsu.edu/users/m/mdutta3/capstone$ python queriestest.py
**********************Check For Query****************************
1. Term Query
2. Match Query
3. Range Query
4. EXIT
Enter Your Input:1
Enter the Term you are looking for
trump
{
    "_shards": {
        "failed": 0,
        "successful": 5,
        "total": 5
    },
    "hits": {
        "max_score": 1.2165375,
        "total": 9097
    },
    "timed_out": false,
    "took": 35
}
```

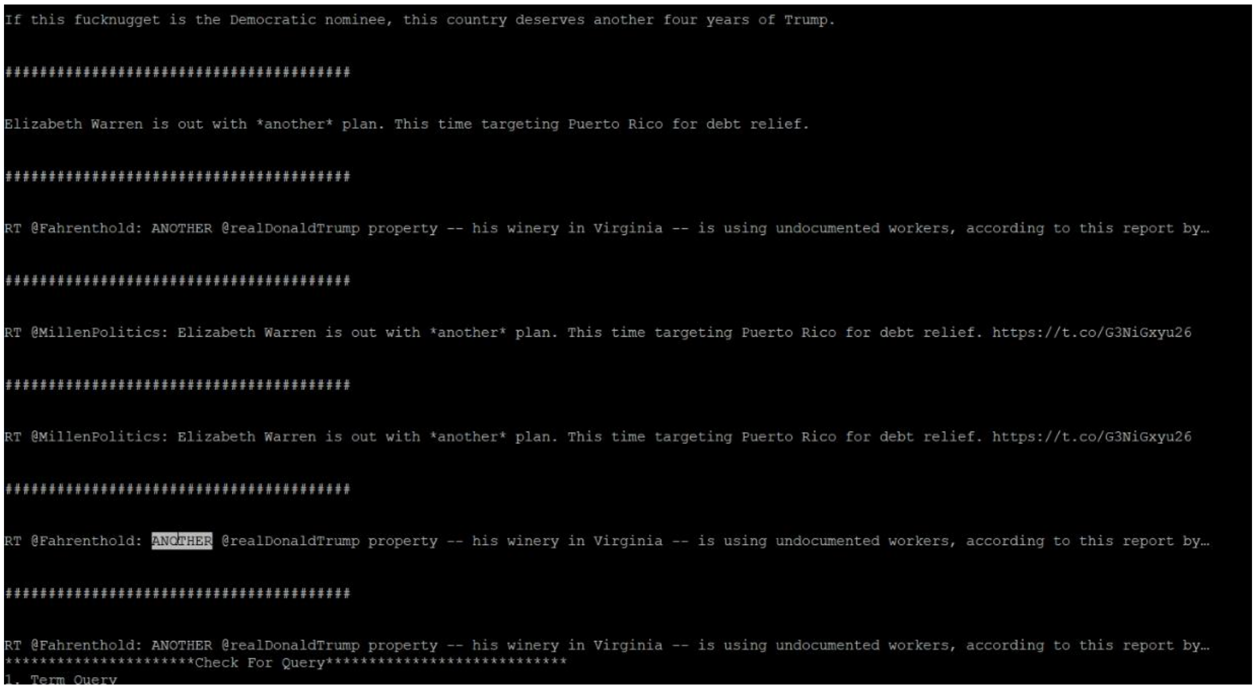


```
{
    "_shards": {
        "failed": 0,
        "successful": 5,
        "total": 5
    },
    "hits": {
        "max_score": 1.2165375,
        "total": 9097
    },
    "timed_out": false,
    "took": 35
}


########################################


RT @4trump247: @charliekirk11 @JoeBiden @realDonaldTrump Trump Trump Trump Trump Trump Trump Trump Trump Trump Trump ususususususususus


########################################

Obama/ Trump

########################################

NOT TRUMP?

########################################

@mattlaytonradio Trump
```

2. Match Query: It will match the words, not exact words but it will check all the tweets which contain these words.

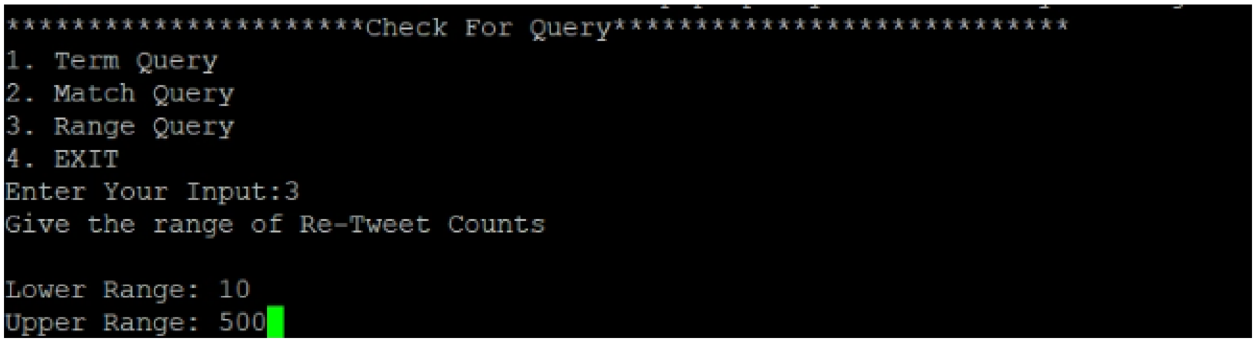


```
**********************Check For Query****************************
1. Term Query
2. Match Query
3. Range Query
4. EXIT
Enter Your Input:2
Enter the Phrase you want to match
This is another proof
```

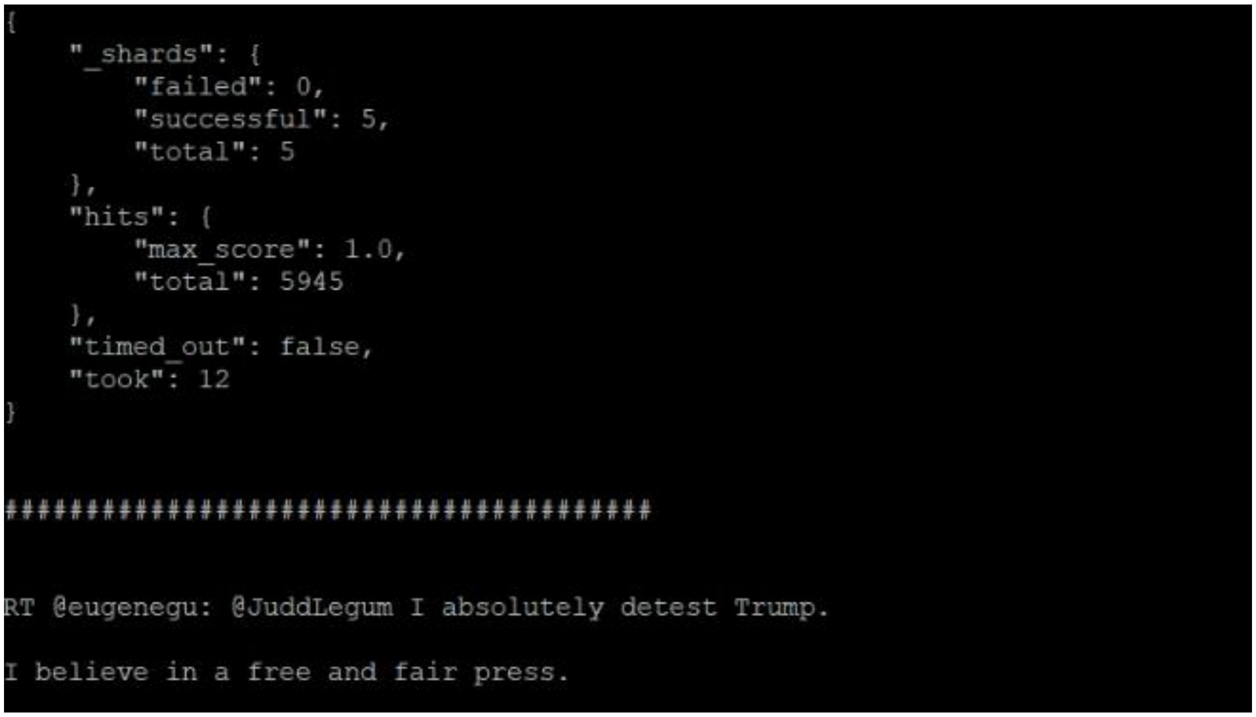


```
If this fucknugget is the Democratic nominee, this country deserves another four years of Trump.

########################################

Elizabeth Warren is out with *another* plan. This time targeting Puerto Rico for debt relief.

########################################

RT @Fahrenthold: ANOTHER @realDonaldTrump property -- his winery in Virginia -- is using undocumented workers, according to this report by…

########################################

RT @MillenPolitics: Elizabeth Warren is out with *another* plan. This time targeting Puerto Rico for debt relief. https://t.co/G3NiGxyu26

########################################

RT @MillenPolitics: Elizabeth Warren is out with *another* plan. This time targeting Puerto Rico for debt relief. https://t.co/G3NiGxyu26

########################################

RT @Fahrenthold: ANOTHER @realDonaldTrump property -- his winery in Virginia -- is using undocumented workers, according to this report by…

########################################

RT @Fahrenthold: ANOTHER @realDonaldTrump property -- his winery in Virginia -- is using undocumented workers, according to this report by…
**********************Check For Query****************************
1. Term Query
```

3. Range Query: Gives the number of retweet counts and we are entering the lower range and upper range.



```
**********************Check For Query****************************
1. Term Query
2. Match Query
3. Range Query
4. EXIT
Enter Your Input:3
Give the range of Re-Tweet Counts

Lower Range: 10
Upper Range: 500
```

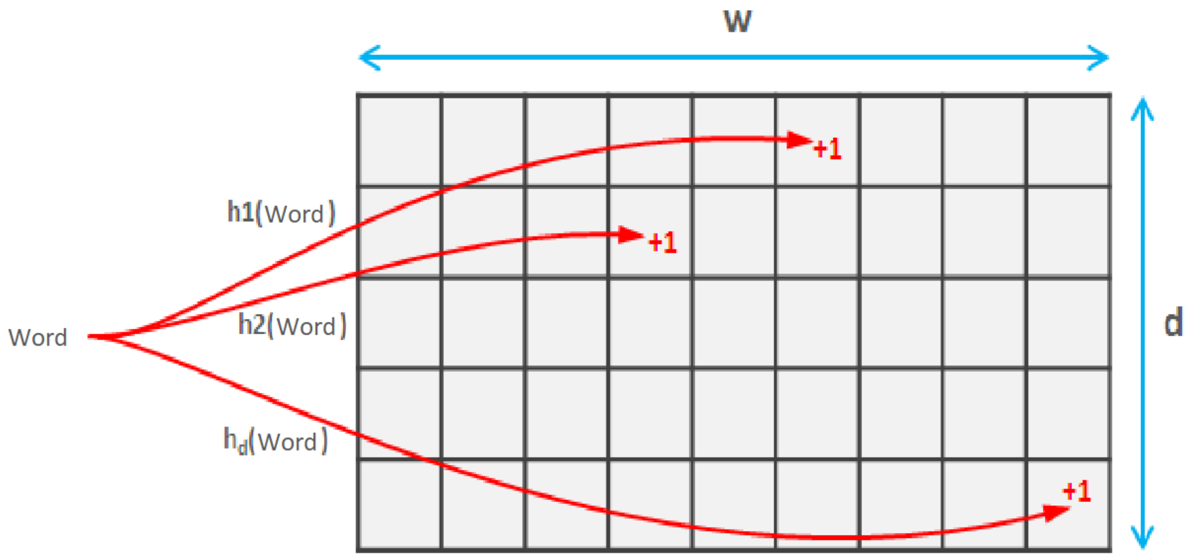


```
{
    "_shards": {
        "failed": 0,
        "successful": 5,
        "total": 5
    },
    "hits": {
        "max_score": 1.0,
        "total": 5945
    },
    "timed_out": false,
    "took": 12
}


#########################################


RT @eugenegu: @JuddLegum I absolutely detest Trump.

I believe in a free and fair press.
```

<u>TECHNIQUES/ALGORITHMS Top-K or Heavy hitters</u>

Top - K or Heavy hitters tracking in text stream has various applications in the business world like analysis of sentiments, efficiently receive feedback, find new business and product avenues, etc.

For the heavy-hitters implementation, we use a Count Min Sketch data structure. It is a probabilistic data structure that uses multiple hash functions and multiple buckets to increment counters when a word's hashed value resolves to specific buckets. In-stream data, it would be difficult to keep absolutely accurate counts of words that appear and thus, such a data structure would be a more efficient approximation, if the objective is to track the most frequently mentioned words a.k.a. the Heavy Hitters.



The depth 'd' of such a data structure is the number of hash functions used and the width 'w' can be anything the user may want to define.

Each word will be passed to the 'd' hash functions and will evaluate to a column value for each row where the counter will be incremented. Counters are initialized at 0.

Now, since the number of unique words that are possible in a real-time stream is way more than d*w (the number of counters in the data structure), we have the possibility of two completely different words incrementing the same counter. This is why the data structure is considered probabilistic in nature, which means that the words that the data structure has observed as having the most counts, is most likely an incorrect count, because of false evaluations to the same counter. To reduce the risk of overestimating these values, we pick the lowest value of all the counters that the word has evaluated to, which encapsulates the Count Min part of this data structure.

For our implementation, we have used a CountMinSketch library developed by Tyler Barrus, available at: https://github.com/barrust/count-min-sketch

License for use of this library: https://github.com/barrust/count-min-sketch/blob/master/LICENSE

Results:

When a specific topic was subscribed to from the Twitter streaming API, for example, ‘FBI’,

We were able to observe that after applying stop word filtering some of the expected words appeared in the heavy hitters query, like

1. Russia
2. Trump
3. Attorney General Barr
4. Spy
5. Information

Stop word filtering is done appropriately in accordance with the words that did not seem relevant to the topic ‘FBI’