# CS 6350
# ASSIGNMENT _____3_____

Names of students in your group:
Aditya Kulkarni (axk230069@utdallas.edu)

Number of free late days used: _____0_____

Note: You are allowed a **total** of 4 free late days for the **entire semester**. You can use at most 2 for each assignment. After that, there will be a penalty of 10% for each late day.

Please list clearly all the sources/references that you have used in this assignment.

## Solution:

Part 1: Report is from page 2.
Solution Link:
https://github.com/adityavkulkarni/6350_assignment3/tree/master/Part1

Part 2: Report is from page 8.
Solution Link:
Google Colab link:
https://colab.research.google.com/drive/1yP-7jeaYKonvGEgznyWzRck4rDjr39nS?usp=sharing
Github link:
https://github.com/adityavkulkarni/6350_assignment3/tree/master/Part2
Dataset used:
https://snap.stanford.edu/data/wiki-Vote.html
Dataset is present in the GitHub repository:
https://github.com/adityavkulkarni/6350_assignment3/tree/master/Part2/input

Attached in Submission:

1. *CS6350_Assignment3 – combined report*
2. *ReadMe files for part 1 and part 2*
3. *Zip file of repository: https://github.com/adityavkulkarni/6350_assignment3*
4. *Output for Part 2 can be found in: https://github.com/adityavkulkarni/6350_assignment3/tree/master/Part2/output*

Assignment 3 Part 1

# Spark Streaming with Real Time Data and Kafka

## Problem Statement

In this part, you will create a Spark Streaming application that will continuously read text data from a real time source, analyze the text for named entities, and send their counts to Apache Kafka. A pipeline using Elasticsearch and Kibana will read the data from Kafka and analyze it visually.

## Execution steps

Detailed steps are present in [README.md]

**Execution instructions:**

1. Start Zookeeper: `bin/zookeeper-server-start.sh config/zookeeper.properties`
2. Start Kafka service: `bin/kafka-server-start.sh config/server.properties`
3. Create topics:
    - reddit:

    `bin/kafka-topics.sh --create --topic reddit --bootstrap-server localhost:9092`

    - ner:

    `bin/kafka-topics.sh --create --topic ner --bootstrap-server localhost:9092`
4. Start Elasticsearch: `cd $ELASTICSEARCH_DIR; bin/elasticsearch`
5. Start Kibana: `cd $KIBANA_DIR; bin/kibana`
6. Copy `logstash-ner.conf` from repository to `$LOGSTASH_DIR/config`, and replace your credentials
7. Start Logstash: `cd $LOGSTASH_DIR; bin/logstash -f config/logstash-ner.conf`
8. Create index "ner": `curl -X PUT "localhost:9200/ner -u user:password`
9. Make changes to config.ini to setup kafka information and reddit credentials
10. Run `ner_analyser.py`: `spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.5.1 ner_analyser.py`
11. Run `reddit_scraper.py`: `python3 reddit_scrapper.py`
12. 2Open Kibana: `http://localhost:5601/app/dashboards#`
13. Go to `Analytics->Discover->Select Dataview="ner"` and now you can visualise the data
14. Sample dashboard output is present in the report

Kafka, Spark and ELK need to be downloaded and setup for running the above steps. Libraries required to run are mentioned in: [requirements.txt]

## Data and Dashboard Description:

We have fetched top **1000** posts and then streamed **29,832** posts (submissions) from "r/all" which is a less filtered feed of the most popular posts on Reddit.
These posts are then published in Kafka topic "reddit". The published posts are read in Pyspark streaming. After cleaning text and tokenizing it, we have identified the named entities using NLTK – "*pos_tag*" and "*ne_chunk*". The named entities and their count are published to the topic "*ner*". Logstash is configured to read from Kafka topic "*ner*" and push the data to "*nerkibana*" index in Kibana.
The dashboard has the following visualizations:

1. Bar graph showing 10 most frequent named entities
2. Pie chart showing frequency distribution
3. Word cloud of frequent named entities
4. Time steps showing the count of records analyzed per 5 minutes
5. Semi-circle meter plot for unique word count vs total word count

## Output Analysis:

A total of **7.8 million** unique named entities were identified and total named entity count is **24 million**.

According to bar graph, the most mentioned word on "r/all" from 13:30 to 14:45 on July 21, 2024, is "*Biden*" with about **358,000** mentions, followed by "*Joe*", "*news*" and "*Trump*" with about **150,000** counts. This is due to the news about Joe Biden dropping out of presidential race which led to widespread news and follow up discussions across Reddit. All the top 10 frequent named entities are related to politics except for "*Engine"* which has about **68,000** mentions.

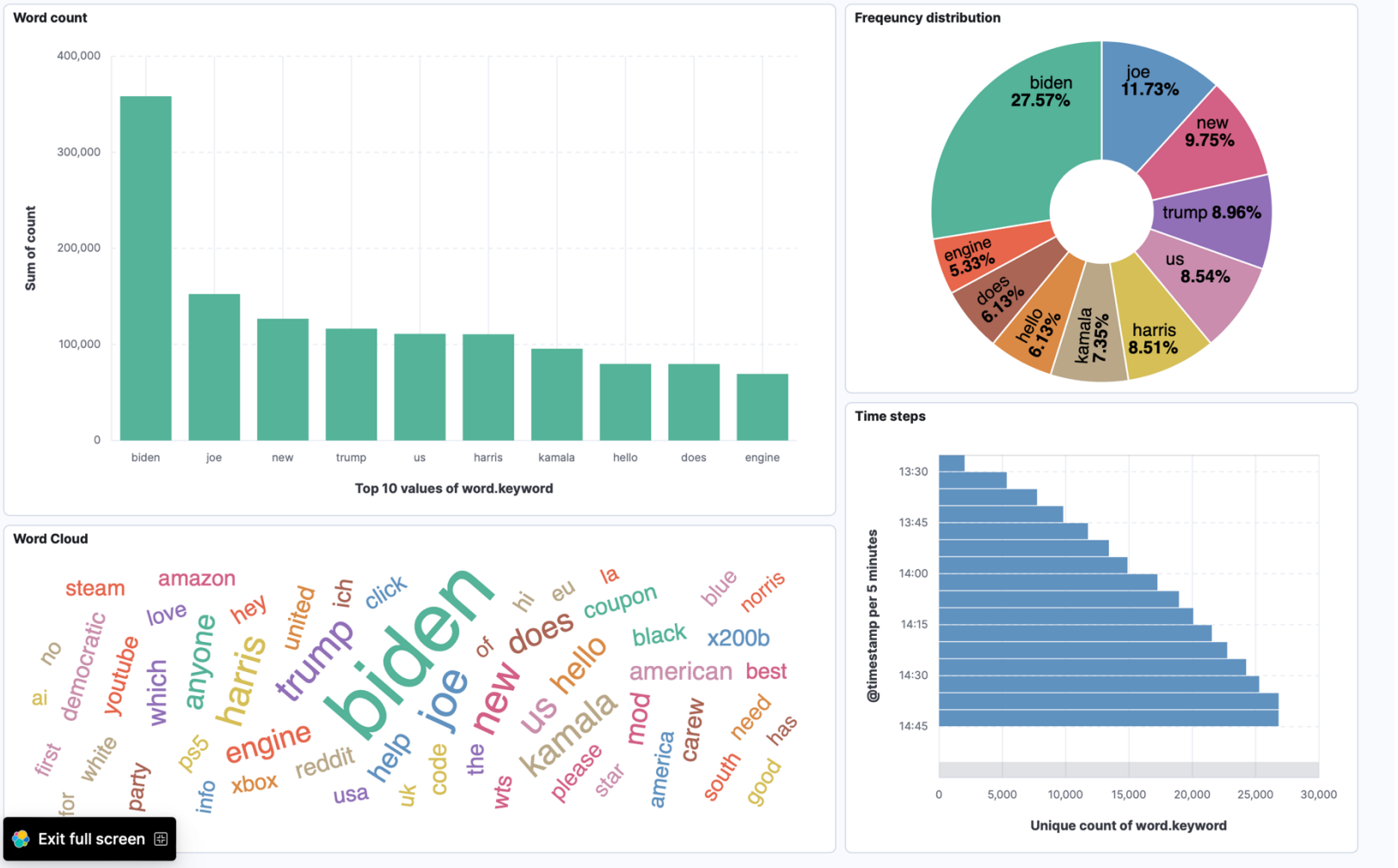
According to timestep graph, which shows number of records(named entities) pushed to Kibana in 5 minutes interval, we observed that: at *13:30* only **5,344** named entities were processed. But at *13:55*, the count increased to **14,882** which was when the news was out about the presidential candidates changing. And the count increased steadily to **24,252** at *14:30*.

## Output:

**Output of reddit_scrapper.py:**

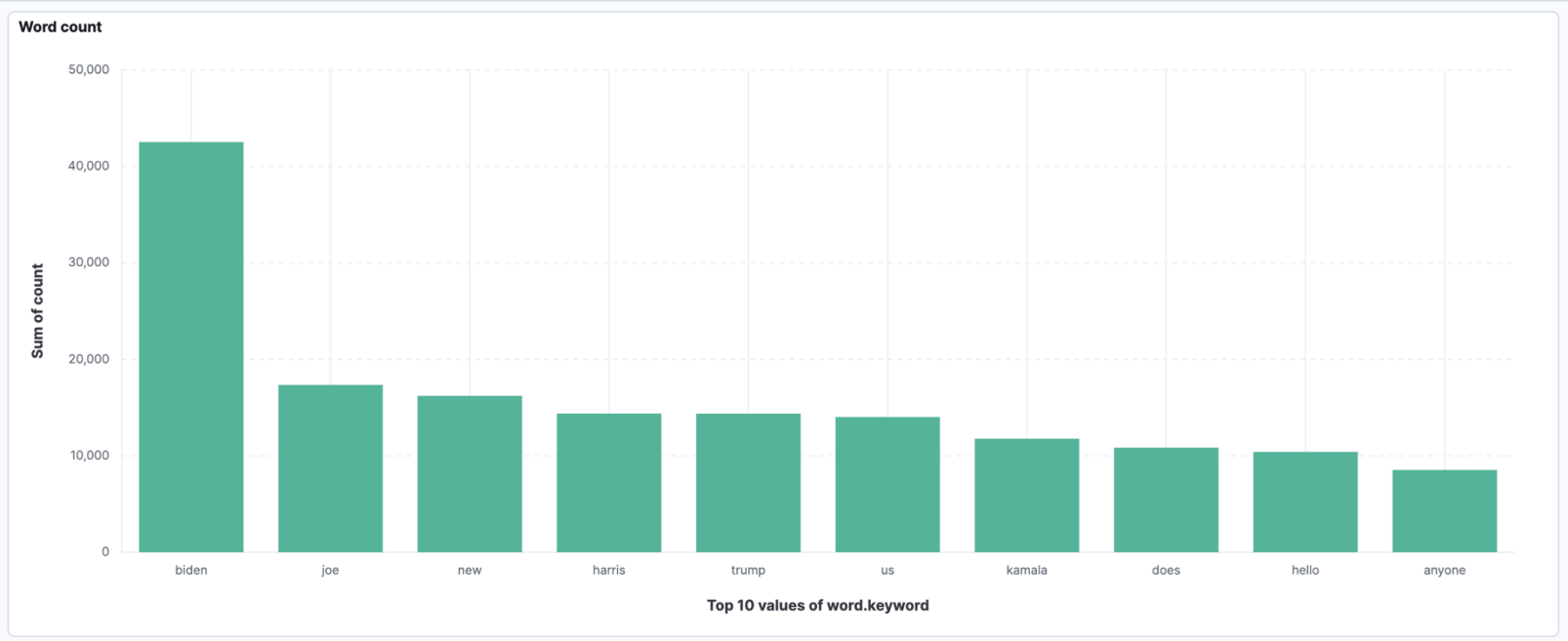
```
(.venv) adityakulkarni: ~/UTD/6350_BDA/assignment-3/part1  (master)$ python3 reddit_scraper.py
Connected to Reddit: bdastreamer6350
Connected to Kafka@localhost:9092
Connected to Subreddit: all
Fetching top posts from: all
Sent 1000 articles to Kafka queue
Streaming subreddits: all
Sent 30832 articles to Kafka queue
```

## Dashboard:

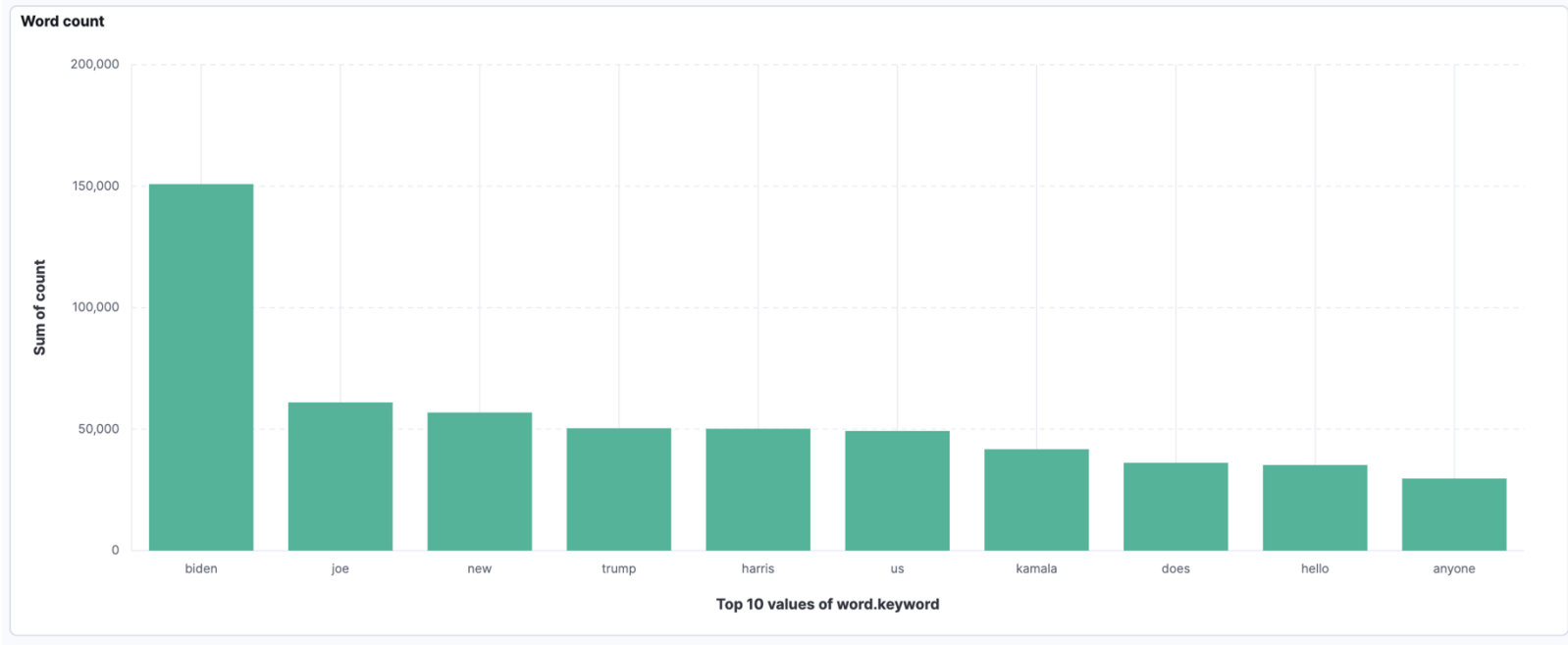

Word count
400,000
300,000
200,000
100,000
0
Sum of count
biden
joe
new
trump
us
harris
kamala
hello
does
engine
Top 10 values of word.keyword
Freqeuncy distribution
biden 27.57%
joe 11.73%
new 9.75%
trump 8.96%
us 8.54%
harris 8.51%
kamala 7.35%
hello 6.13%
does 6.13%
engine 5.33%
Time steps
@timestamp per 5 minutes
13:30
13:45
14:00
14:15
14:30
14:45
0
5,000
10,000
15,000
20,000
25,000
30,000
Unique count of word.keyword
Word Cloud
steam
amazon
love
hey
united
ich
click
hi
eu
la
coupon
blue
norris
no
democratic
youtube
which
anyone
harris
trump
biden
joe
new
of
does
us
hello
black
x200b
ai
american
best
first
white
party
ps5
engine
reddit
help
the
kamala
mod
carew
need
has
for
info
xbox
usa
uk
code
wts
please
star
america
south
good
Exit full screen

**Bar Plots:**

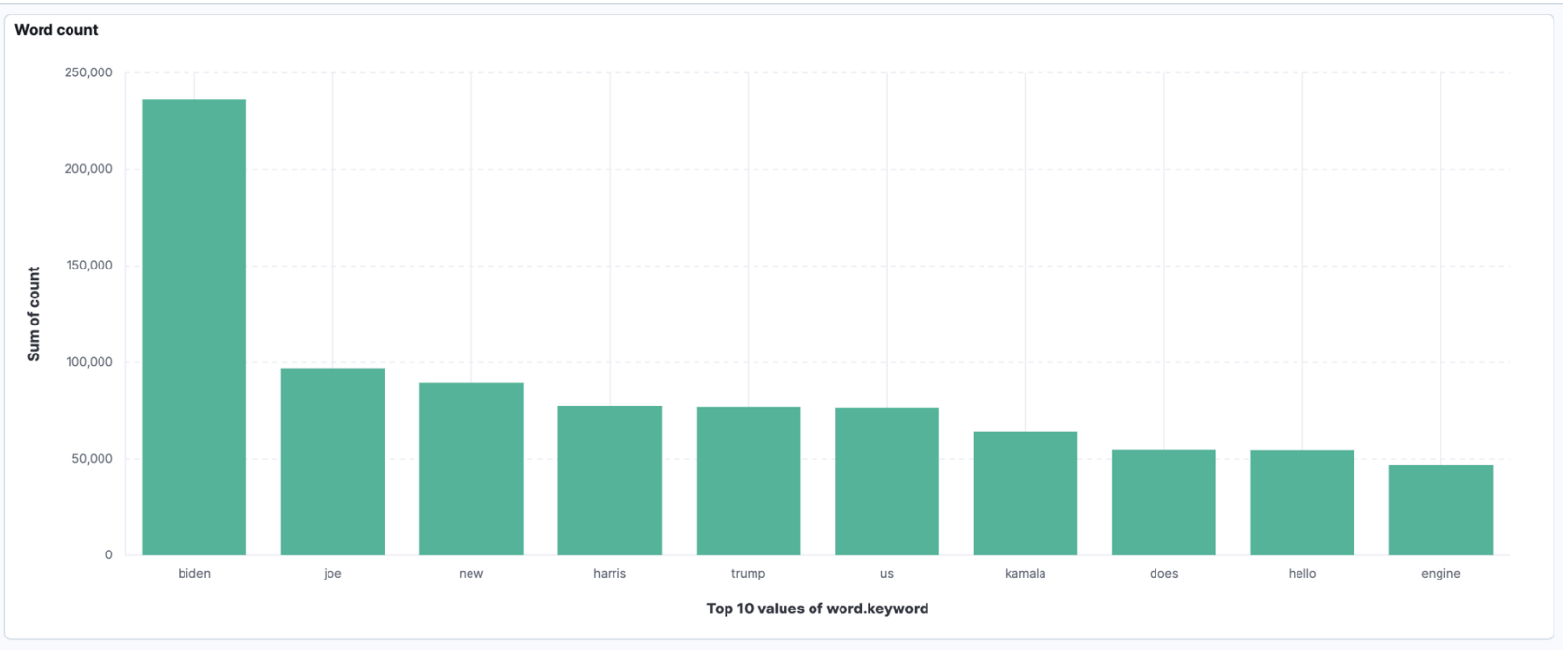
*15 minutes:*



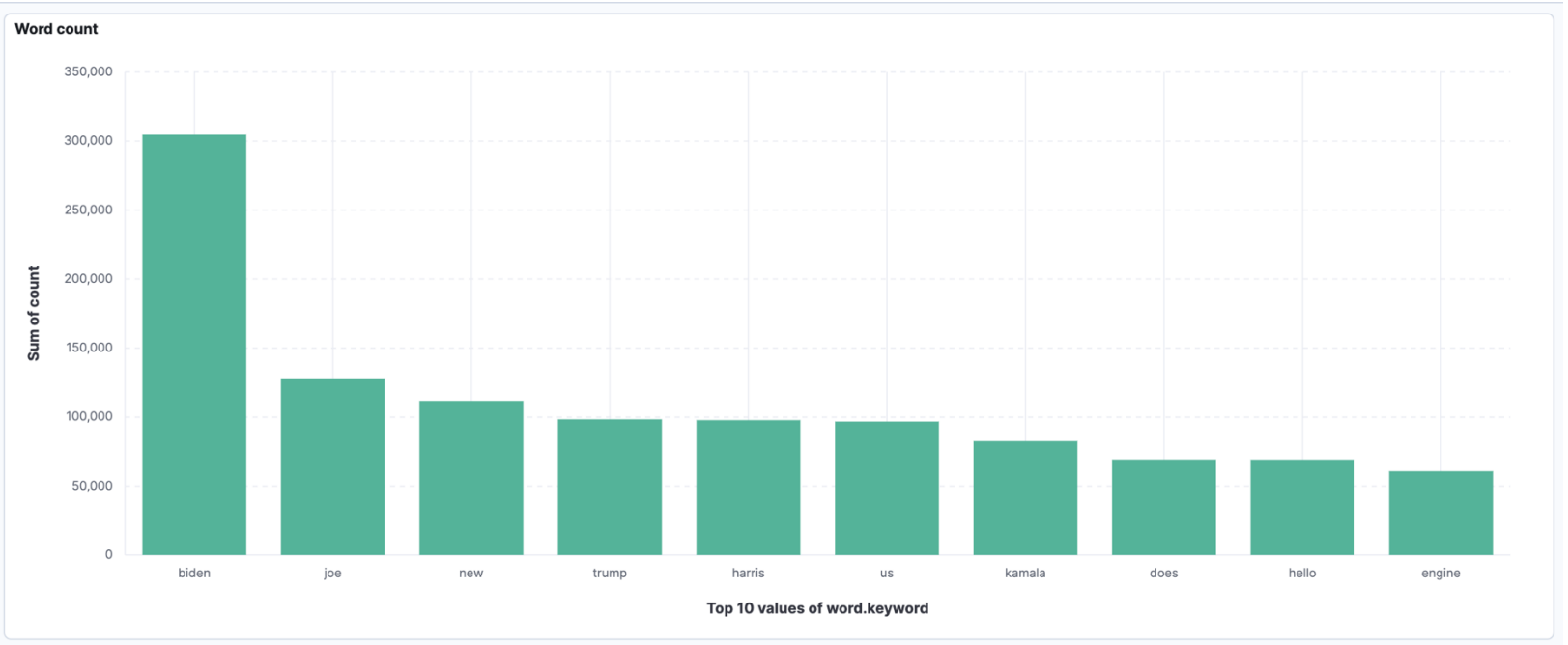

*30 minutes:*



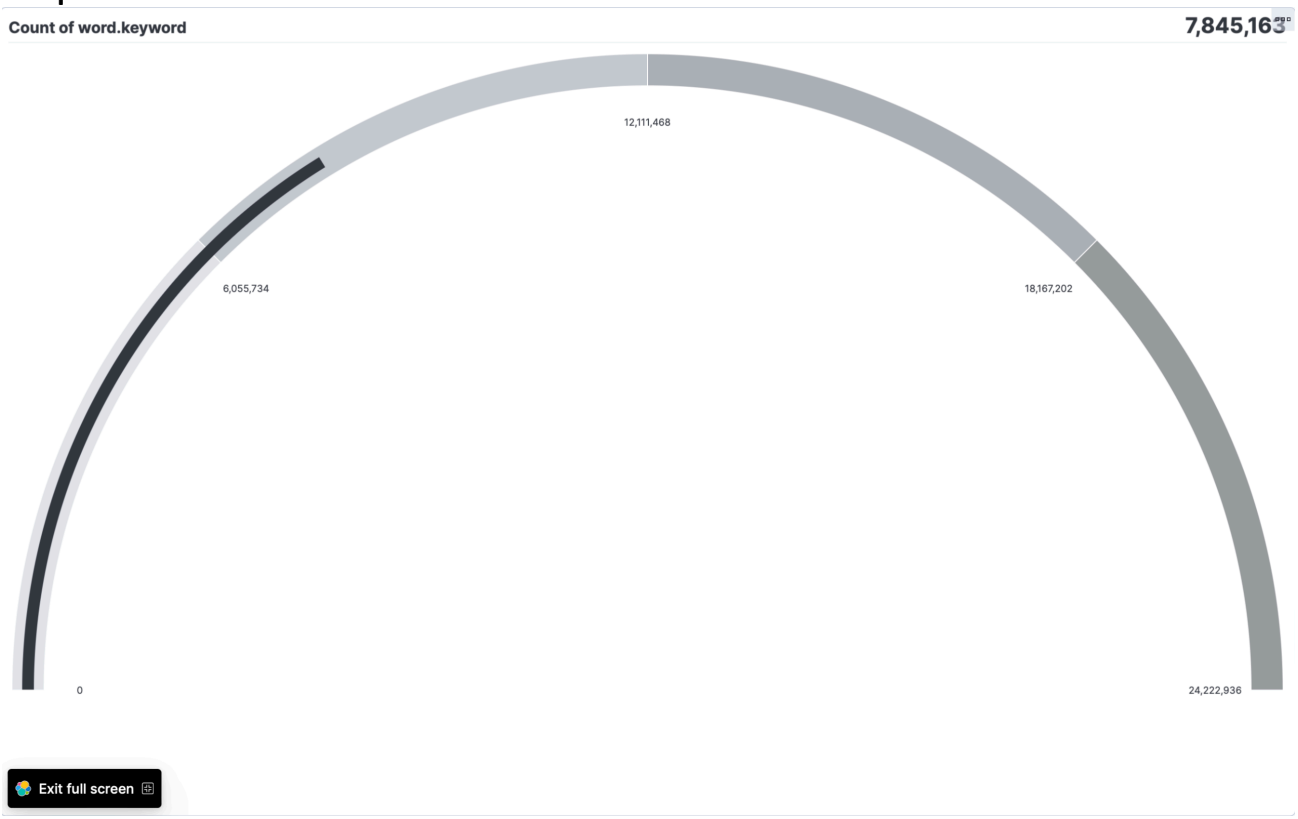
*45 minutes:*



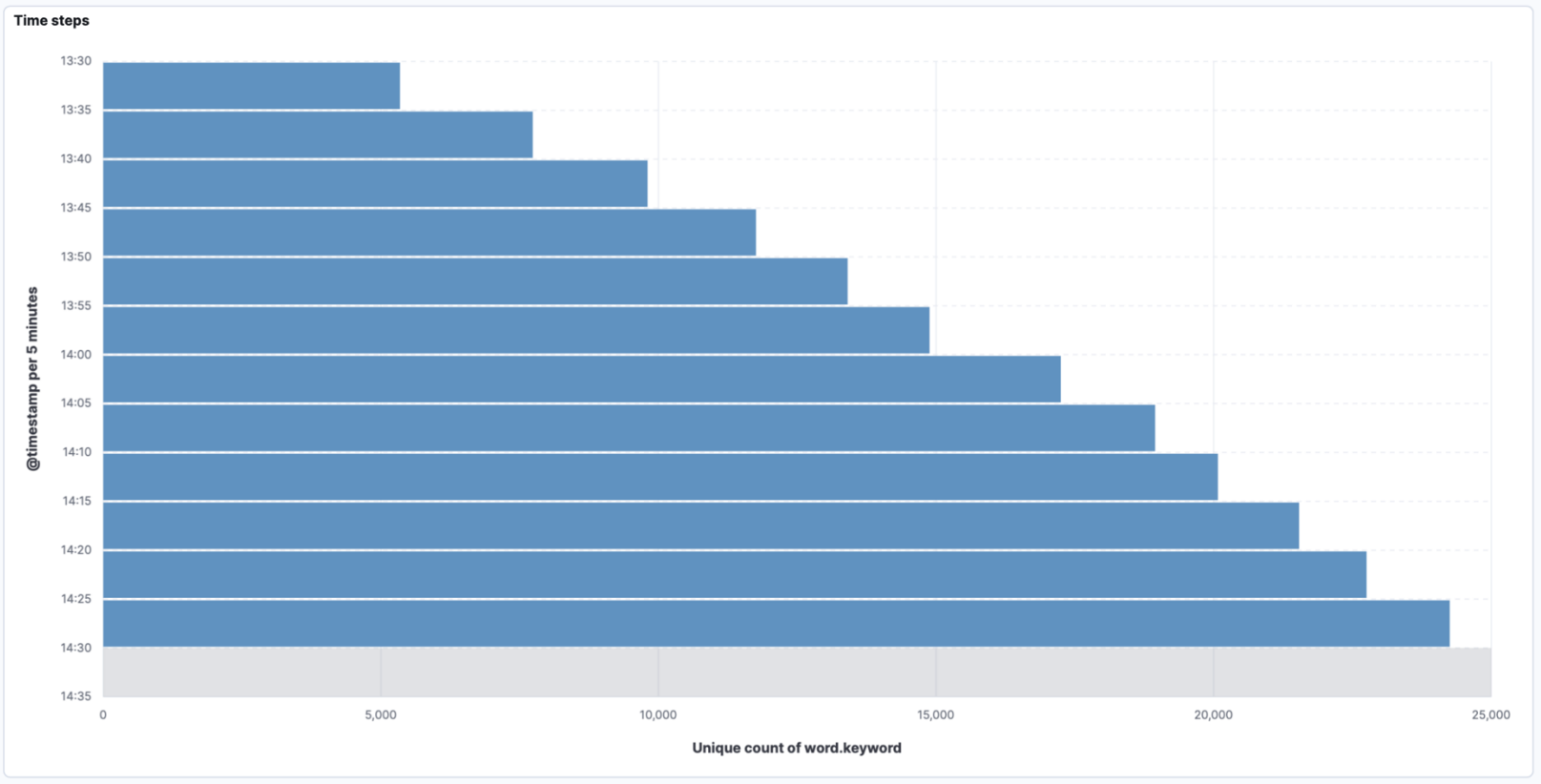

*60 minutes:*



## Unique words vs Total word count:




## Timestep plot:



Assignment 3 Part 2

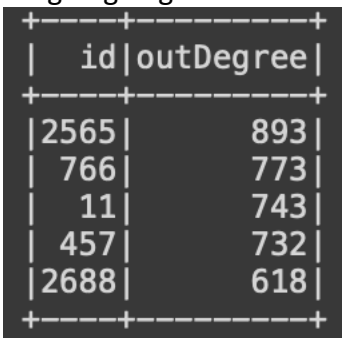
# Analyzing Social Networks using GraphX/GraphFrame

## Problem Statement

In this part, you will use Spark GraphX/GraphFrame to analyze social network data. You are free to choose any one of the social network datasets available from the SNAP repository.
You will use this dataset to construct a GraphX/GraphFrame graph and run some queries and algo- rithms on the graph.

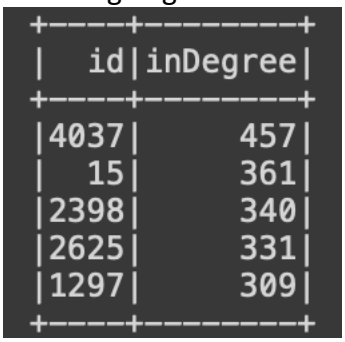
## Output of Queries

1. Find the top 5 nodes with the highest outdegree and find the count of the number of outgoing edges in each.



| id | outDegree |
|---|---|
| 2565 | 893 |
| 766 | 773 |
| 11 | 743 |
| 457 | 732 |
| 2688 | 618 |

2. Find the top 5 nodes with the highest indegree and find the count of the number of incoming edges in each.



| id | inDegree |
|---|---|
| 4037 | 457 |
| 15 | 361 |
| 2398 | 340 |
| 2625 | 331 |
| 1297 | 309 |

3. Calculate PageRank for each of the nodes and output the top 5 nodes with the highest PageRank values. You are free to define any suitable parameters.

| id | pagerank |
|---|---|
| 4037 | 32.594991861511566 |
| 6634 | 29.983324850197288 |
| 15 | 27.01099302107744 |
| 2625 | 25.133148331225353 |
| 2398 | 20.3858798963518 |

4. Run the connected components algorithm on it and find the top 5 components with the largest number of nodes.

*Connected Components:*

| component | count |
|---|---|
| 0 | 7066 |
| 532575944741 | 3 |
| 592705486870 | 3 |
| 936302870556 | 3 |
| 103079215124 | 2 |

*Strongly Connected Components:*

| component | count |
|---|---|
| 1 | 1300 |
| 26 | 1 |
| 19 | 1 |
| 0 | 1 |
| 22 | 1 |

5. Run the triangle counts algorithm on each of the vertices and output the top 5 vertices with the largest triangle count. In case of ties, you can randomly select the top 5 vertices.

| id | count |
|---|---|
| 2565 | 30940 |
| 1549 | 22003 |
| 766 | 18204 |
| 1166 | 17361 |
| 2688 | 14220 |

## Summary:

1. The indegree signifies the number of votes received.
2. The outdegree signifies the number of votes given by the person.
3. A higher PageRank indicates a higher level of importance. This is based on the idea that ids that are linked to by many other votes are likely to be more important.
4. The connected components signify the voting groups, ie. people generally reach vote within the same set of ids.
5. Triangle count suggests that 2 ids have cast votes for the same id.