From Reddit to Insights: A Big Data Pipeline for Entity Recognition on Comments from Subreddit “news”

Amber Hasan, Mikiyas Midru
Executive Masters, The University of Texas at Dallas

***Abstract*— Live comments are fetched via the “news” subreddit with Reddit API, streamed to Kafka, and processed using PySpark to extract named entities and their frequencies. Processed data is then visualized in Kibana as bar plots, highlighting the top entities over time.**

I. Introduction

In our class, we learned about Kafka and live data streaming. This project extends past that by connecting to visualization tools to display the results using the ELK stack (Elasticsearch, Logstash, Kibana).

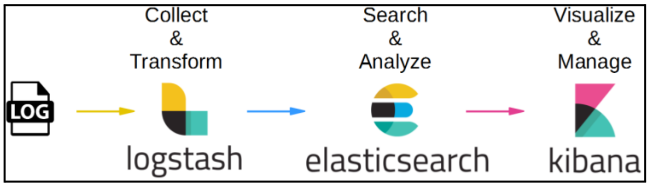


Fig. 1 We are using the ELK stack to collect, analyze, and visualize entities from Reddit API.

II. Background Work

Here is the flow we wanted to do:

1. *Reddit API Integration:*

reddit-api.py fetches live comments from Reddit API “news” subreddit and sends to Kafka topic1.

2. *Message Broker:*

Kafka acts as the central message broker, storing data from the Reddit API before further processing.

3. *Stream Processing:*

pyspark-named.py processes data from Kafka using PySpark, performing Named Entity Recognition (NER) via SpaCy. Results written to Kafka topic2.

4. *Data Transformation:*

Logstash (logstash.conf) consumes data from Kafka topic2, applies transformations

5. *Data Storage:*

Elasticsearch indexes the transformed data, making it searchable and suitable for visualization.

6. *Visualization:*

Kibana visualizes data from Elasticsearch, displaying insights in a dashboard.

III. Study of the Technique

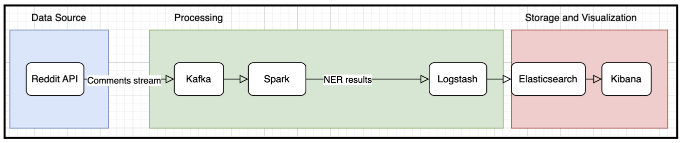
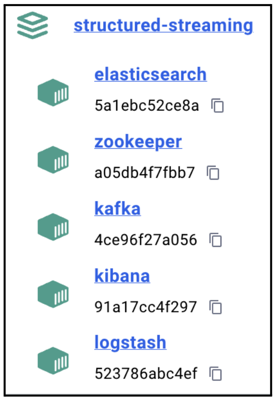


Fig. 2 This is the full tech stack of the project from start to finish.

1. *Docker*

With “docker compose up”, we were able to only run the containers, then our Reddit API script, then our Spark command. This made the integration a lot easier.



structured-streaming

elasticsearch
5a1ebc52ce8a

zookeeper
a05db4f7fbb7

kafka
4ce96f27a056

kibana
91a17cc4f297

logstash
523786abc4ef

2. *Data Collection*

We got our own client key and secret for the Reddit API (PRAW). Our python script, which we ran using “python reddit-api.py” would fetch comments from the “news” subreddit every 10 ms and post in topic1.

3. *Data Processing*

Spark Structured Streaming reads from Kafka, performs Named Entity Recognition using SpaCy, and counts occurrences of entities. The output is published to another Kafka topic (topic2) for downstream processing.

4. *Data Visualization*

Elasticsearch and Kibana retrieve data from topic2. The most frequent named entities are visualized as bar plots in Kibana.

IV. Preliminary Results

The pipeline successfully ingests live comments from a subreddit and processes named entities.

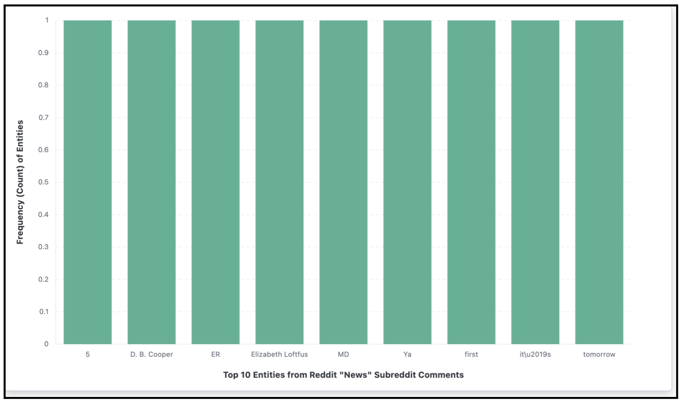


Fig. 3 In the first minute, we see random first entities with count 1.

V. Results and analysis

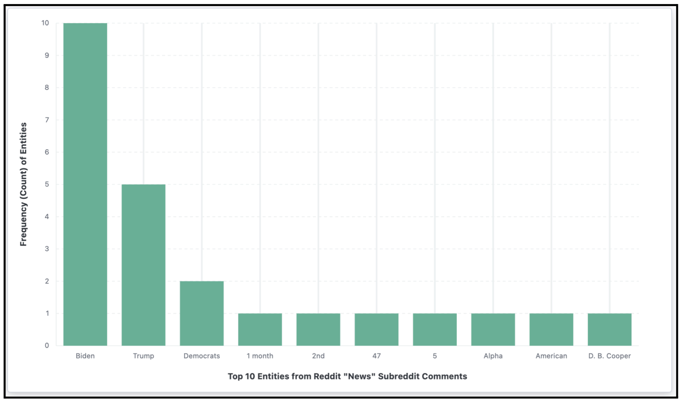


Fig. 4 In the next few minutes, count quickly increases with Biden and Trump

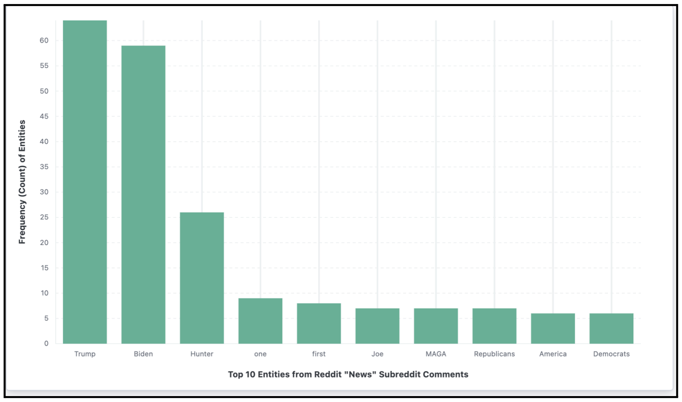


Fig. 5 In the next minutes, Trump comes at a higher count than Biden.

These primary results indicate that Trump is talked about a lot these days in Reddit, probably because he was chosen as our next president. Biden is also talked about a lot since he is currently in office until Trump takes the next spot.

In the next couple hours, we see a new entity "Hunter". We were not sure of what "Hunter" meant so we looked it up and found out that Hunter Biden was pardoned by Joe Biden, which is why he is talked about in the comments as well. Of course, Republicans and Democrats are also key entities in Reddit API:

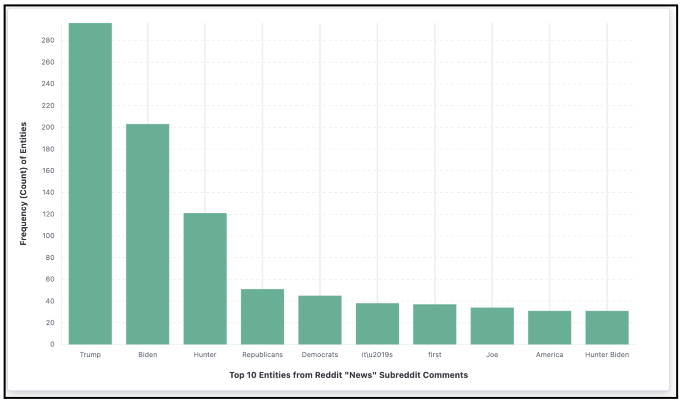


Fig. 6 After 1 hour of running.

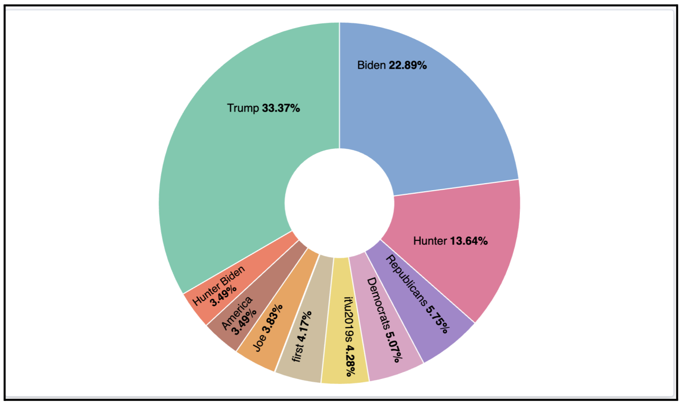


Fig. 7 Representing the data as a donut.

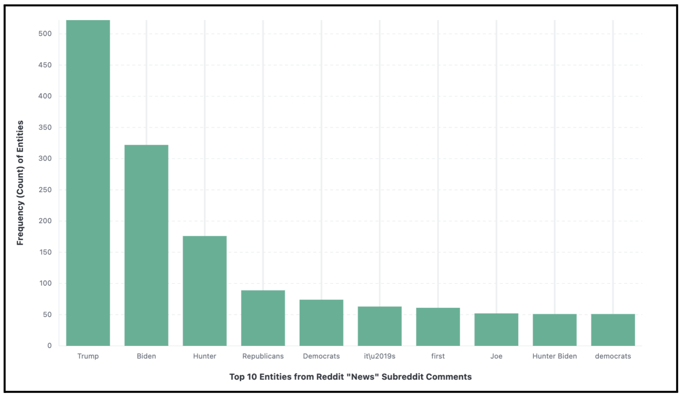


Fig. 8 After two hours of running.

VI. Conclusions

We successfully streamed comments from Reddit API's "news" subreddit into Kafka and used Spark to process the NER's with SpaCy and then used the ELK stack to collect, analyze, and visualize the data. We learned some shocking news too. Future work could involve expanding the pipeline to other data sources, optimizing performance, or exploring more advanced analytics.

https://github.com/mikimaine/structured-streaming
Video Explanation: https://youtu.be/HyR5VguchhM