Sentiment Analysis and Clustering: Spark Streaming

Ragi Bhatt, Irene Torrijos Robles, Haolin Chen

May 2023

1 Introduction

The aim of this project is to tackle the challenge of real-time analysis of data streams and enhance our practical skills in handling and analyzing streaming data. Our focus is on integrating Apache Spark Streaming, a fault-tolerant stream processing framework, with a Python-based natural language processing toolkit, to perform sentiment analysis and clustering on a data stream in real time.

All the code used for this project is available on Github: `https://github.com/irenetorrijos/sparkproject`

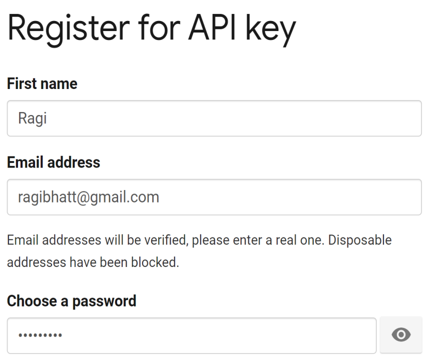
2 Implementation

In this section, we discuss the implementation of how to connect to a streaming API, parse the textual input, and use Spark Streaming to read the data and TextBlob to tag it for sentiment analysis. Our goal is to combine these technologies to process and analyze data in real time. Additionally, we will employ machine learning techniques such as clustering and window operations on the data stream to identify valuable insights such as sentiment trends over time or abrupt mood shifts.

To showcase our findings, we will create a dynamic dashboard that will continuously update with the latest data, enabling us to visually monitor and interpret the real-time analysis.

2.1 API Connection and Access

Here we fetch the top news headlines by using the NewsAPI every 2 minutes and sending them to a Spark Streaming application using a socket connection. The API key and port number are defined as constants, and a server socket is created and bound to the specified address and port. The script listens for incoming connections and accepts them, printing a message to indicate the Spark Streaming application has connected.


Register for API key

First name: Ragi

Email address: ragibhatt@gmail.com

Email addresses will be verified, please enter a real one. Disposable addresses have been blocked.

Choose a password: ••••••••••

Figure 1: Registering to get API key

To connect to the API we use the API key that we receive by registering on the NEWSAPI website as shown in 1 After successfully registering, there is an HTTP GET request to the API URL to fetch the top headlines. The response is then parsed as JSON, and the 'articles' list is extracted. The script then iterates through the list, converts each article to a JSON string, and sends it to the client using the socket connection. The script waits for 1 second before sending the next message.

The articles can be found by looking for a specific author, publication date, language, etc. The way we connect to the API is shown in Figure 2.

After all the articles are sent, the script waits for 2 minutes before fetching the headlines again, and the process repeats.

2.2 Methodology

After receiving the input stream, we pre-process the data by cleaning the text where we remove digits, emoticons, URL's, mentions, etc. to proceed with analyzing the sentiment of the text with the TextBlob. Post this we perform clustering by grouping the sentiment scores

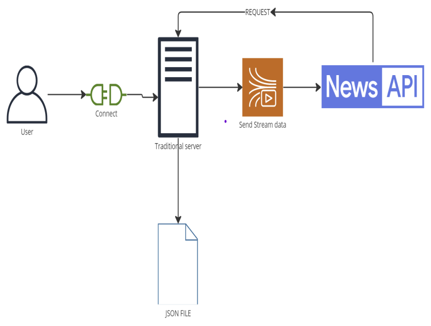


Figure 2: Diagram representation of connection to the API

using the StreamingKMeans algorithm. The sentiment data is then vectorized and saved to a CSV file, which can be used for further analysis.

2.3 Sentiment Analysis

After we acquire the text as input, using the TextBlob library we return the list of two computed values - polarity and subjectivity scores. It extracts the title and content fields from the message, concatenates them with a period, and passes the resulting text to sentiment analysis to compute the sentiment scores. If the message does not have a content field, it computes the sentiment scores using only the title field. The function then returns a dictionary containing the title and sentiment scores.

The computed scores are returned as a dictionary containing the title and sentiment scores. The sentiment orientation is provided by polarity, and it can be stated as a float with a value between [-1.0, 1.0] and be either positive, negative, or neutral. Subjectivity, on the other hand, measures the degree of subjective opinion, feeling, or judgment inside the text and is represented as a float between [0.0, 1.0].

2.4 K-Means Clustering

After sentiment analysis, we cluster our data using Streaming K-means, a K-means variant made for streaming data. Three clusters have been formed, which could, for example, stand for pleasant, unpleasant, and neutral sentiments.

To give more weight to recent data while making predictions, the decay factor of the model is set to 0.1, which is the forgetting factor of the data stream.

We save the prediction findings to a CSV file, along with the news story's title, polarity, subjectivity, and cluster prediction. We are given an ongoing record of the sentiment analysis of the news data as a result of this procedure, making it feasible to track sentiment patterns in real-time.

The diagram here summarizes the methodology. shown in Figure3

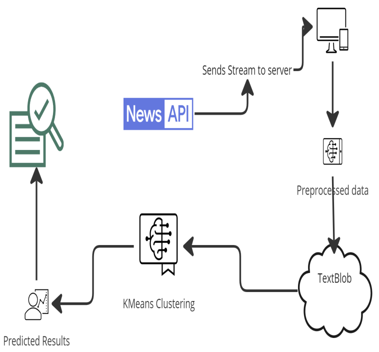


Figure 3: Connection to API while performing the analysis

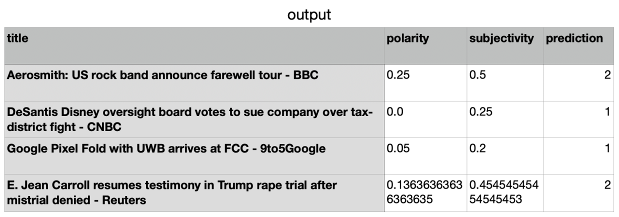
3 Results

In addition to the obtained results, Spark Streaming and TextBlob work together to create a solid framework for real-time sentiment analysis of news data that has the potential to offer insightful analysis of evolving public sentiment patterns.

However, the clustered articles can not be considered as good clusters because the features used for clustering might not be enough. In addition to that ambiguous titles of news can heighten different emotions which are subjective to the user when exposed to the content.

3.1 Visualization

As seen in Figure **??**, the data frame contains the results of the analysis of 18 news headlines along with labels for their polarity, subjectivity, and anticipated sentiment. The labels for the sentiments range from 0 to 2, with 0 denoting negativity, 1 denoting neutrality, and 2 denoting positivity. Positive values indicate positive sentiment, negative values indicate negative sentiment, and values close to zero indicate neutral sentiment. The polarity scores range from -1 to 1. Higher values denote greater subjectivity or a more personal opinion. The subjectivity scores range from 0 to 1. Seven of the 18 news headlines were expected to be neutral, six to be favorable, and five to be unfavorable.



output

title	polarity	subjectivity	prediction
Aerosmith: US rock band announce farewell tour - BBC	0.25	0.5	2
DeSantis Disney oversight board votes to sue company over tax-district fight - CNBC	0.0	0.25	1
Google Pixel Fold with UWB arrives at FCC - 9to5Google	0.05	0.2	1
E. Jean Carroll resumes testimony in Trump rape trial after mistrial denied - Reuters	0.13636363636363635	0.45454545454545453	2

Figure 4: Example DataFrame of the predictions from K-means clustering

3.2 Dynamic Dashboard

The plot.ipynb file generates a plot of the analyzed data every second and shows it as a scatter plot. Figure **??** an example of a random selected plot, where we can see the groups 0, 1 and 2. To create a dynamic plot, we stored some of these plots in a gif, where the evolution in time can be seen. The dynamic plot is in our github

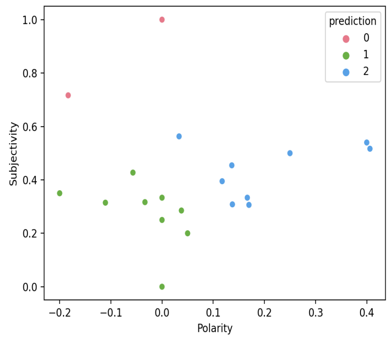


Figure 5: Scatter plot of a sentiment analysis

4 Conclusion

The goal of this project was to analyze and group different News from the most popular news articles in different categories by utilizing sentiment analysis and K-means clustering. Despite the model's inability to effectively distinguish between the categories, the project shed light on the difficulties involved in clustering chat data and provided valuable insights into the process of working with Spark Streaming and APIs more broadly.