Real Time Analysis of Twitter hashtags using Apache Spark Structured Streaming

Hritwik Singhal
Computer Science Engineering
LNMIIT
Jaipur, India
18ucs055@lnmiit.ac.in

Raunak Goyal
Computer Communication Engineering
LNMIIT
Jaipur, India
18ucc162@lnmiit.ac.in

Utkarsh Gupta
Computer Communication Engineering
LNMIIT
Jaipur, India
18ucc140@lnmiit.ac.in

Aarchi Gupta
Computer Science Engineering
LNMIIT
Jaipur,India
18ucs156@lnmiit.ac.in

Sameer Gupta
Computer Science Engineering
LNMIIT
Jaipur,India
18ucs008@lnmiit.ac.in

Abhay Singhal
Computer Science Engineering
LNMIIT
Jaipur,India
18ucs011@lnmiit.ac.in

***Abstract*—This project will fetch recent tweets based on “keywords” and “pages”, using Twitter API v2, filter hashtags from those tweets and give those hashtags to spark streaming for processing. After that it will launch a flask web server on localhost:5001 to view the data in a visual dashboard powered by ApexCharts.**

Keywords—Twitter API, Apache Spark, Spark Streaming, Live tweets streaming, Tweet Analysis, Flask, ApexCharts

1. Introduction

One of the biggest sources of data today are Social networks like Twitter, Facebook & Instagram. This means that for marketers, big data specialists, journalists and other professionals they are an extremely valuable asset.

Twitter is one of the biggest social media websites today, where people tweet and interact with each other about topics they like. So naturally it has a very rich source of data. We can analyze these tweets to get interesting and important insights about people’s opinions towards a particular topic/brand, sentiments of people, what people are talking about and the overall trends in general.

Also, many time-sensitive business applications today can Harness the potential of real-time Twitter data and make decisions based on it in fields of IoT application, network intrusion detection, fraud detection.

2. Problem statement

To deliver a sophisticated solution to end customers that will provide them with real-time analysis of millions of tweets and dynamic visualisation of trending twitter hashtags to assist them better sell their product and raise social awareness about a certain topic.

3. Twitter Api v2

Twitter introduced their Twitter API v2 at the end of 2020. It was built from the ground up to be faster and have access to many new endpoints. It also comes with more features and data that you can pull and analyze, and a lot of new functionalities which are useful for Academic scholars as well as for enterprise.

But it has some limitations of its own: 450 queries per 15 minutes, 500K queries per month, we can get tweets based on some keywords only.

The endpoint URL being used for queries is api.twitter.com/2/tweets/search/recent. The parameters that were given to twitter api were: the ‘keyword’, the ‘end_date’, ‘max_results’, ‘next_token’, ‘tweet fields’ which include ‘id’, ‘text’, ‘author_id’, ‘geo’, ‘conversation_id’, ‘created_at’, ‘lang’, ‘entities’.

The response we received contained the tweets as well as their metadata like ‘author’, ‘date’ ‘hashtags’ etc. We separated the hashtags from those fields and to get hashtag counts, this response is passed to Spark through Socket.io running on port number 9009.

4. Apache spark structured streaming

Apache Spark, an open-source unified analytics engine for large-scale data and stream processing, is one of the most used engines for real-time data processing and provides the option for continuous computations, as data is continuously flowing through them.

Apache Spark provides two ways of working with streaming data; Spark Streaming and Spark Structured Streaming.

Spark Structured streaming is a distributed and fault-tolerant streaming engine that enables processing data in real-time. It is built on top of Apache Spark SparkSQL engine which deals with running stream data. It is fault-tolerant, scalable, and ensures recovery of any fault as soon as possible with the help of caching and checkpoints.

We have used the Pyspark interface for Apache Spark in Python. It allows us to write Spark applications using

Python APIs. We first created a local SparkSession, the starting point of all functionalities related to Spark. Then we create a streaming DataFrame that represents text data received from a server listening on Localhost using sockets. After receiving the data from socket, we used two spark built-in functions namely 'Split' and 'explode' to split each line into multiple rows and store them in Dataframe. Then we Generate running word count using the 'groupBy' function of spark. At last we send the results from spark to flask Using REST API.

5. Flask

Flask is a micro web framework based on the Werkzeug WSGI toolkit and the Jinja2 template engine. It permits extensions that add application functionalities as if they were built into Flask itself. We chose Flask as the server since it is a lightweight web application framework that is designed to deliver results quickly while also allowing for future expansion. With Flask, the code is always simply what the developers put in it, with no extra code in charge of things we don't use. In our case, the Flask server collects hashtag data from Spark and directs it to the frontend.

We have created three REST APIs in the server out of which one is the default GET API that is used to render HTML template for the dashboard whenever the server starts. After processing Twitter's data, Spark will send hashtag data to the server through POST API "/updateData". Then the frontend will call GET API "/refreshData" to get hashtag data from the server. These two APIs are repeatedly called by Spark and frontend respectively to update hashtag count on the dashboard and the dynamic updation on the frontend is done by the Jinja2 template engine.

6. Apexcharts

ApexCharts is a modern, free and open-source charting library that has its integrations with Vue, React and Angular frameworks. This library has beautiful and interactive charts and aids developers in creating attractive and dynamic web page visualisations. It's MIT-licensed and free to use in commercial applications. ApexCharts has NPM support and it is better, with bigger datasets, but not by much.

In our case, Javascript calls the server API to GET hashtag data and then it passes to ApexCharts API to get a responsive and dynamic bar graph.

In our implementation of the trending hashtag dashboard we created a dynamic apex bar chart which is loaded with the options to provide responsive visualizations on frontend. Frontend will call the backend server's GET API to fetch the trending(Real Time) top 15 twitter hashtags with their corresponding tweet count. These data values are then passed into the bar chart through the series option.To get real time visualization on frontend , the GET API will regularly be called in an interval of 2 seconds. The bar chart is also packed with the option of responsiveness which means that if the screen resolution changes , the bar chart remains visually readable.

7. Future Prospects

This project can be useful to market a particular product by associating it to a trending hashtag. It is also useful to promote contests and giveaways to boost user engagement. It can also be useful to raise awareness about a particular topic, also to discover trending topics and also to build a community around a hashtag.

Also it can be used in many time-sensitive applications today who can use this real-time data and make decisions in fields of IoT application, network intrusion detection, fraud detection and improve our lifestyles in many ways.

8. Project Location

The complete source code of the Project is provided under GPLv3 license and is currently hosted on Github.com. Instructions to run the project are provided in the README.md file. You are free to modify and redistribute the source code as well as binaries, provided you follow GPL terms. Link to the project: Spark-tweet

9. Acknowledgement

We would like to express our sincere thanks to Dr. Animesh Chaturvedi, for his valuable guidance and support in completing this project in the field of Cloud Computing. His immense knowledge, profound experience and professional expertise in Cloud Computing has enabled us to complete this Project successfully. Without his support and suggestions, this project would not have been Successful. It helped us in doing a lot of Research and we learned a lot of things related to this topic

10. References

[1] Twitter API v2 tools & libraries | Docs | Twitter Developer Platform

[2] Search Tweets - Twitter Official Github Examples

[3] Make-your-first-request Twitter API

[4] GET /2/tweets/search/recent | Docs | Twitter Developer Platform

[5] Apache Spark - Unified Engine for large-scale data analytics

[6] Overview - Spark 3.2.0 Documentation

[7] Spark Streaming - Spark 3.2.0 Documentation

[8] Structured Streaming Programming Guide - Spark 3.2.0 Documentation

[9] PySpark Documentation — PySpark 3.2.0 documentation

[10] PySpark - RDD

[11] Welcome to Flask — Flask Documentation (2.0.x)

[12] ApexCharts.js - Open Source JavaScript Charts for your website