Hasib Ziai
Michael Rodriguez
Aaron Gumabong
Saytu Singh

Dataset Ingest and Analysis Service
Final Report

Project Description

The main objective of our project is to create a cloud-based service that analyzes and provides an easy-to-use interface with detailed and meaningful results about a dataset. We decided to use responses to a tweet as our dataset to collect a large amount of text that could be used in our analysis service. The benefits of creating a service like ours would be to help a user use the information gained to better understand the discourse of a thread of responses. The benefits of us deploying a system like this on the cloud is scalability. This will allow us the processing power needed to scale up to tracking and storing many response threads in the future.

Project Inspiration

In 2012, a security researcher discovered an ‘exploit’ which allowed the researcher to gain access to a victim’s network via Linksys routers and certain internet connected devices. They wrote a script which scanned all IPv4 addresses and patched the vulnerability, then scanned the internet for more devices to patch, effectively creating a botnet. The results of the scan was then sent to the researcher’s ‘command’ servers, resulting in the “Internet Census of 2012”.

Simple Use Case

A user’s goal will be to view detailed analytics about a response thread in a simple interface. And the user will specify the dataset beforehand, and the type of analysis for the dataset will be defined preemptively. They will then be able to view the selected information from a simple website.

Design and Implementation

The overall design of our project is a linear system that starts with a script hosted on the cloud that will monitor, track, and retrieve a list of Twitter user’s tweet threads. Within the script the text will be cleaned of all unnecessary and meaningless characters and words. The retrieved text will then be saved into a cloud container that will be used as the pulling area for Spark to use. The processed text will then be sent to another container where a database will store the dataset to be used in the front end UI. We utilize the cloud platform to host our front end on a website that will handle user interaction and display the datasets.

For the implementation of our Twitter text script, we are using a python library, Tweepy, to get all the tweets from a certain user or users. The script can be run in the background using ‘nohup’, which ignores the hangup signal and allows the script to run indefinitely as a background process on the Virtual machine (VM) environment. This python script is running on a flexible VM environment, allowing it to allocate more resources as needed, such as if we want to use the script on a larger set of users. The output data from this VM is then stored into a Google Cloud storage bucket, which is advantageous for scalability reasons. This data can then be used by the Spark cluster for jobs, and the data output from these jobs are then stored into another separate bucket. This final data can then be put into a simple database, such as Google Cloud Firestore or if we were working with a large dataset, we could use Google’s BigTable storage. Now that this data is easily accessible through the Firestore database, we can use a very simple web interface to output the data for the user.

Data Collection

Twitter API allows for easy collection of a large amount of tweets with some rate restrictions based on time. Using the Twitter API allows us to pull a great deal of user generated content that can be used in a Spark application to possibly lead to some meaningful analysis. The data collected is a response thread to a user’s tweet.

Tasks to be Completed - > Completed

For the final phase of the project, now that we have the data from Twitter, our main objective is to do some meaningful analysis of the data using our Spark/Hadoop cluster. A suggestion that I am looking into is the use of Machine Learning algorithms to create some sort of predictive search or analysis about the Twitter user(s) from whom we are pulling the data from. We are thinking of using the random forest algorithm and applying it to our data to see if we can find anything meaningful. In addition to this, we still need to complete a simple interface to view the data. This will not be anything elaborate, and will just be the bare necessities needed for the user to view the analyzed data. We have decided to use Spark rather than Hadoop for our clusters due to the fact that Spark can allow real-time processing of data, and we could use Twitter’s Stream API to allow us real-time analysis of data coming from our set of twitter users. These are all subject to change based upon the constraints placed on our usage of the Twitter API, but they should be considered for future projects.