Data Intensive Computing

LAB2 Report

Objectives

Data Crawling
- Collecting Data from different Sources (NY Time's, Twitter, Common crawl).

Big Data Analytics
- Processing the Data collected Using Hadoop Map Reduce.

Visualization
- Visualizing the Data Using Tableau /D3.js.

DATA COLLECTION:

We collected data for topic **Terrorist** using python from the following Sources
- 1.TWITTER-Using Twitter API.
- 2.NEW YORK TIMES- Using New York Times API.
- 3.Common Crawl- Crawling & processing warc files from Common crawl Index.

Data Magnitude : Collected both Small Data and Big Data
- Small data : Collected Over a period of 1 month.
- Big Data : Collected Over a period of last 3 and Half months.

Also collected Data for sub topics related to Main Keyword.
- Guns
- Shooting

Challenges:

Getting required data from huge common crawl data was tough but solved by indexing required .gz files using required URLS where our Keyword is frequently found.

BIG DATA ANALYTICS:

- We obtained enormous data from different Sources and this voluminous data is processed using Hadoop MR to obtain word count and Word Co-occurrence's.
- We have installed Cloudera Hadoop docker client to interact with HDFS where we process the big data parallelly.
- All the data is moved to Hadoop Infrastructure (HDFS) using cloudera Hadoop docker client.
  $hadoop fs -put /src/data/*.txt /user/ss/MR/input
- The data processing instructions are specified by our custom mapper and reducer files.
- Both Mapper and Reducer are implemented in python Language.
- Mapper and Reducer are run using hadoop streaming jar.

Mapper.py:

- All the lines in text are split line by line and stop words are removed from processing using python stemmer library and also programmatically checking for stop words in a custom stop words list.
- Words separated by spaces ae tokenized and associated with a counter of 1.
- These tokens are written to the standard output which are is as input to reducers.

Dependencies for Mapper:

- Libraries required in mapper for stemming and removing stop words can be installed using following commands
  - yum install epel-release
  - yum install python-pip
  - pip install stemming

Reducer.py

- All The stdout from mappers are collected by reducers and the reducers group similar words by incrementing counters.
- All the reducer outputs are consolidated and written into a output file which can be obtained to our local system using the following command.
  $Hadoop fs -get /user/ss/MR/output.

Observation:
As Hadoop processes the data parallelly we are able to process Big data in less time.

Word Count for 100 MB file:
Using Sequential programming:
Time:15 mins.
Using Cloudera Hadoop:
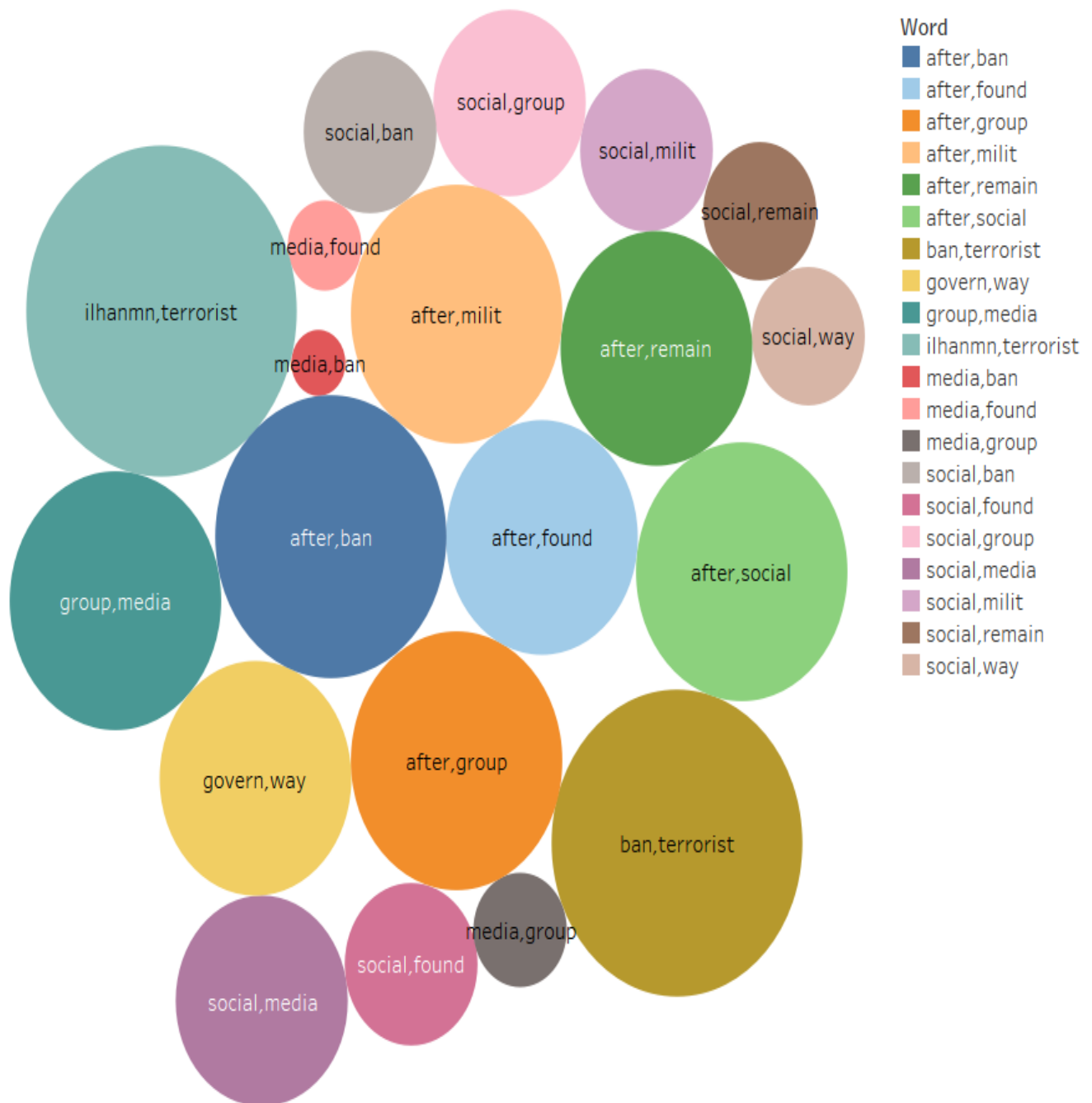Time:<5 mins.

DATA VISUALIZATION:

We have visualized both Word Count and Word Co-occurrence for inferring knowledge
on data using Tableau.

Twitter Big Data Word Count Top 20

Word. Color shows details about Word. Size shows sum of Count. The view is filtered on sum of Count, which ranges from 2,977 to 67,422.

Twitter Big Data Word Co-occurence Top 20

Word. Color shows details about Word. Size shows details about sum of Count. The marks are labeled by Word. The data is filtered on Count, which ranges from 479 to 964.

With the help of visualization, we inferred that the most frequent word, Co-occurred words in our dataset are "**Terrorist, ilhanmn**" etc.

Flow Diagram: