Lab 2- Part 2:

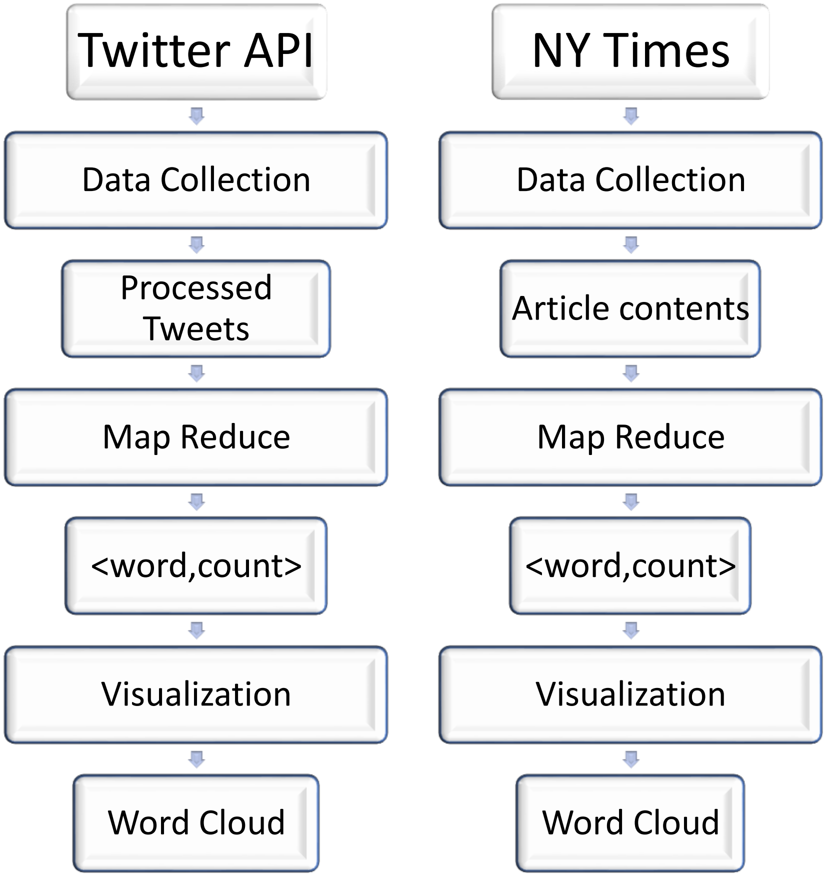
Map Reduce

Team Members:

Vigneshwaran Vasanthakumar #50248708

Siddharth Selvaraj #50247317

Process Flow:



Data Collection:

The topics chosen for data collection are **“Cambridge Analytica”** and **“Driverless cars”.**

Tweets were collected for the related topics using the package **“twitteR”.** The collected tweets were then cleaned by removing the non-ascii letters, symbols and URLs.

URL of the articles were collected using the package **“rtimes”** and the contents of the article were extracted from the respective URLs using the package **“Rcrawler”.**

The contents of the article were then cleaned by removing the non-ascii letters, symbols and URLs.

Map-reduce:

The tweets and articles collected are given as inputs to the mapper in the form of text files.

The mapper function cleans the input data and parses them into words and removes the stop words.

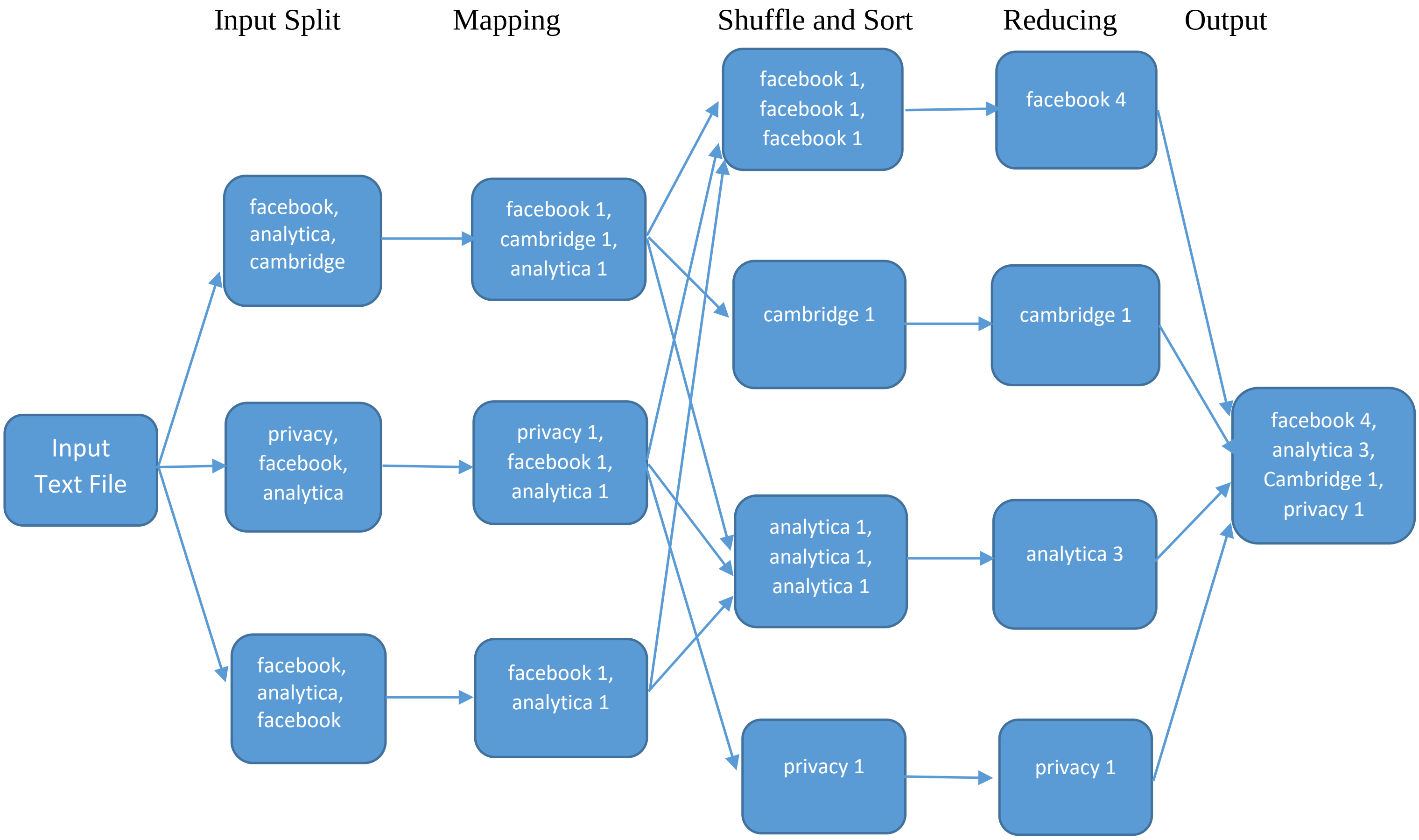
It produces the <key, value> pair as the mapper output where key is the word and value will be one. Hadoop then sorts these <key,value> pairs and gives them as the input to the reducer.

The reducer function will count the number of times the words have occurred and produces the <word, count> as the output which is stored as csv file and the top ten words with highest count are selected from that.

Co-occurrence:

For the top 10 words with highest count in the tweets and articles, the co-occurrence is found with other words in that tweet or that paragraph in the article.

Sample Map-reduce process



Input Split
Mapping
Shuffle and Sort
Reducing
Output
Input Text File
facebook, analytica, cambridge
privacy, facebook, analytica
facebook, analytica, facebook
facebook 1, cambridge 1, analytica 1
privacy 1, facebook 1, analytica 1
facebook 1, analytica 1
facebook 1, facebook 1, facebook 1
cambridge 1
analytica 1, analytica 1, analytica 1
privacy 1
facebook 4
cambridge 1
analytica 3
privacy 1
facebook 4, analytica 3, Cambridge 1, privacy 1

Visualization:

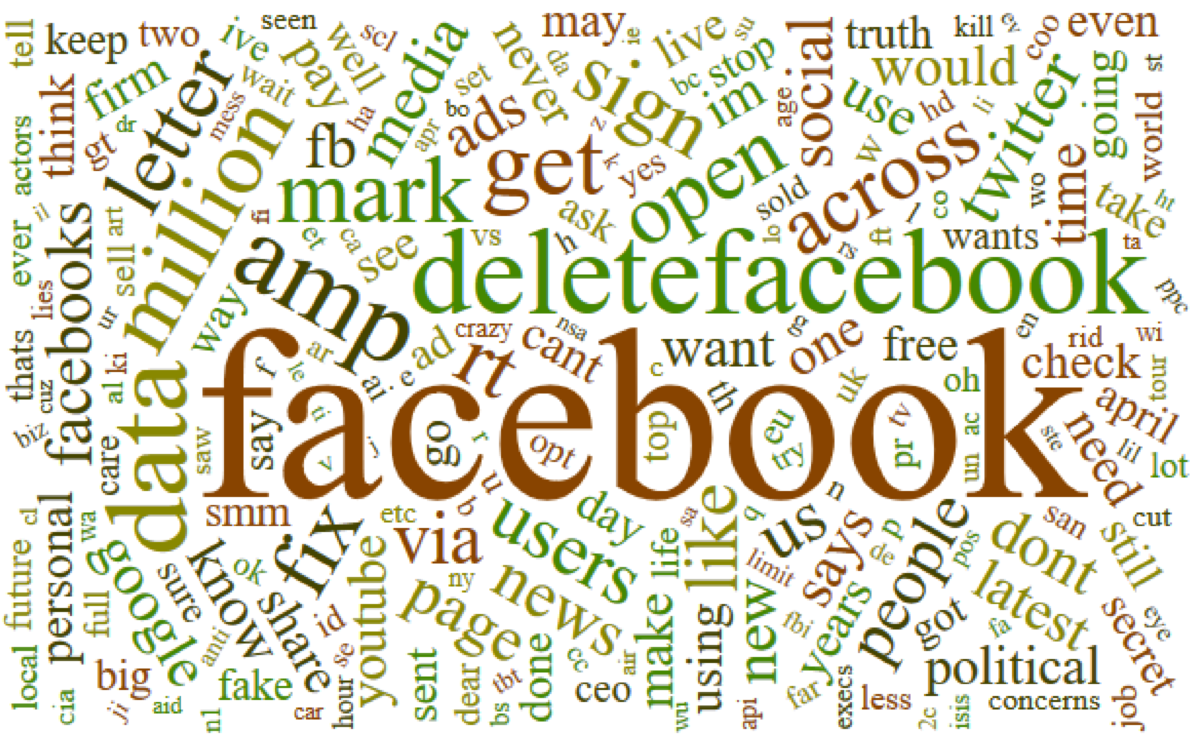
Word clouds are generated for the map-reduce output using d3.js.

d3.csv() function is used to read the data from csv files. This function returns objects in the form of {text,size} pairs. These objects are then passed to **d3.worcloud()** function to generate the corresponding word cloud. Then word cloud is generated for the outputs obtained from the Hadoop map-reduce process. The word clouds were generated for the following topics:

- Cambridge Analytica (Tweets) – one day data, large data and co-occurrence word clouds.
- Cambridge Analytica (Articles) - one day data, large data and co-occurrence word clouds.
- Driverless Cars (Tweets) - one day data, large data and co-occurrence word clouds.
- Driverless Cars (Articles) - one day data, large data and co-occurrence word clouds.

Cambridge Analytica

Twitter word cloud


facebook
deletefacebook
data
million
amp
mark
get
letter
open
across
twitter
sign
media
social
fb
ads
facebooks
fix
via
users
like
people
news
page
dont
latest
secret
political
want
free
check
one
rt
cant
us
says
years
new
make
using
youtube
share
know
google
personal
smm
day
time
going
would
use
truth
may
even
keep
two
think
firm
way
pay
well
never
live
stop
im
wants
take
world
april
need
still
lot
see
ad
ask
say
go
top
uk
fake
big
sent
dear
done
ceo
less
concerns
job
local
future
thats
ever
actors
tell
seen
kill
wait
ive
sell
care
sure
crazy
etc
opt
try
eu
oh
rid
cut
san
life
limit
execs
isis
car
hour
aid
cia
full
ok
id
ny
air
api
fbi
far
got