UNIVERSITY AT BUFFALO • THE STATE UNIVERSITY OF NEW YORK •

1846

CSE: 587

Data Aggregation, Big Data Analysis And Visualization

AMLAN GUPTA

ATRAYEE NAG

Table of Contents

Abstract

In this project, we have Aggregated data - collected data from 3 diverse sources, such as Twitter, NY Times and CommonCrawl on our major topic - Technology with keywords from 5 subtopics of Technology. We have installed the Hadoop infrastructure in our system through Cloudera Docker and performed an analysis on the aggregated data by running a Map Reduce program for Word Count and Word Co-occurrence. Finally, to visualize our analysis, we have created a webpage powered by D3.js and generated Word cloud and Co-occurrence matrix to represent the Map Reduce output.

Introduction

Over the years, technology has revolutionized our world. Technology has created amazing tools and resources, putting useful information at their fingertips. Modern technology has made it possible for the discovery of many functional and utility devices like the smartwatch and the smartphone. With all of these revolutions, technology has also made our lives easier, faster, better… and more fun.

We have chosen Technology as our main topic and our 5 subtopics are as follows:

- **Amazon** has grown from fledgling online bookseller to one of the most valuable and powerful corporations in modern history in the course of a single generation.

- **Apple** is one of the most influential corporations of modern times, leading the way in making smartphone technology a staple of everyday life and Music in our pockets through iPod.

- **Google** is the most complete library on Earth and nobody can ever refute it. Every detail, information, appointment, question, result, historical event (or future), of any subject and temporality can be found there.

- **Facebook** is one of the top social networking sites, quickly gained popularity in terms of total users and traffic. As more people explore social media, social networking sites have become some of the key online sources they use to learn more about products, organizations, artists and world events.

- **Uber** is the rising company that uses a smartphone app to link riders with drivers. To summon a car, you just tap on the screen of your phone. There are numerous reasons uber has become so popular in our lives: Convenient and Cashless, Professional Service, Safer and more flexible drivers.

Our purpose is to collect tweets and articles which are tweeted/published on these subtopics to get an idea how these technologies are prevalent in our day-to-day lives. The three steps we will perform as a part of this project are:

1. **Data Collection**
2. **BigData Analysis**

3. **Data Visualization**

Data Aggregation

Twitter

Twitter is an American online news and social networking service on which users post and interact with messages known as "tweets". Hundreds of millions of Tweets are sent every day. They are processed, stored, cached, served and analyzed. With such massive content, we need a consequent infrastructure. Storage and messaging represents 45% of Twitter's infrastructure footprint. The storage and messaging teams provide services such as Hadoop, Manhattan, MySQL, Blobstore clusters.

Steps to Collect Data from Twitter API

1. Create an app in apps.twitter.com that allows us to collect twitter data.
2. Generate the token Consumer Key (API key), Consumer Secret (API secret), Access Token, and Access Token Secret
3. We have used R to collect the twitter data using the rtweet library to extract the tweets from United States.
4. The table below shows the keywords we have used for the 5 subtopics:

Amazon	Apple	Google	Facebook	Uber
Amazon	Iphone	Google	Facebook	Uber
Kindle	Ipad	Youtube	Oculus	-
Alexa	Ipod	Gmail	Zukerberg	-
Bezos	Steve Jobs	Adsense	Whatsapp	-
Audible	MacBook	-	Instagram	-
Twitch	Itunes	-	-	-
IMDB	Siri	-	-	-
-	Apple Watch	-	-	-

New York Times

The New York Times (sometimes abbreviated as the NYT and NYTimes) is an American newspaper based in New York City with worldwide influence and readership. The New York Times sends nearly 4 billion emails per year to its customers, ranging from daily newsletters to breaking news alerts to transactional emails. They use a platform with a serverless architecture built on top of Google Cloud Platform.

Steps to Collect Data from New York Times Article Search API

1. We have used the python library articleAPI to collect data for the 5 subtopics.
2. We used the url from the json data that articleApi returns to extract the articles using urllib library.

3. After extraction, we removed the duplicate articles and used Beautiful soup library to collect the content from each article.
4. The table below shows the keywords we have used for the 5 subtopics:

Amazon	Apple	Google	Facebook	Uber
Amazon	Iphone	Google	Facebook	Uber
Kindle	Ipad	Youtube	Oculus	-
Alexa	Ipod	Gmail	Instagram	-
-	MacBook	Adsense	-	-
-	Itunes	-	-	-
-	Apple Watch	-	-	-

Common Crawl

Common Crawl is a non-profit organization that builds and maintains an open repository of web crawl data that is, in essence, a copy of the Internet. The corpus includes web crawl data collected over the last 10 years that can be accessed and analyzed by anyone for a very low cost. This data is stored on Amazon's S3 storage service making it quick and easy to perform analysis, utilizing the cloud's scalable computing and analytics resources.

Steps to Collect Data from New York Times Article Search API

1. As common crawl only uses query by url, we have used techcrunch.com as the source to collect data.
2. We have queried data from common crawl index for articles published in 2019 and amazon s3 returned us the location where the articles are present.
3. We downloaded that chunk, which are the snapshots of pages of those articles.
4. We used Beautiful soup library to collect the content from each article.
5. The table below shows the keywords we have used for the 5 subtopics:

Amazon	Apple	Google	Facebook	Uber
Amazon	Apple	Google	Facebook	Uber

Big-Data Analysis

HDFS Infrastructure Set Up

Docker is a computer program that performs operating-system-level virtualization. It is used to run software packages called containers. Containers are isolated from each other and bundle their own application, tools, libraries and configuration files; they can communicate with each other through well-defined channels. All containers are run by a single operating system kernel and are thus more lightweight than virtual machines. Containers are created from images that specify their precise contents. Images are often created by combining and modifying standard images downloaded from public repositories.

We have used Docker Desktop for Mac and installed Cloudera image as instructed in the handout. We have connected a local folder as a shared space with the Docker.

```
docker run --hostname=quickstart.cloudera --privileged=true -t -i -v
/Users/amlan/Documents/codebase/docker_shared_space:/src --publish-all=true -p 8888
cloudera/quickstart /usr/bin/docker-quickstart
```

The following folders were created to use for executing MapReduce program.

```
hadoop fs -mkdir /user/amlan
hadoop fs -mkdir /user/amlan/MR
hadoop fs -mkdir /user/amlan/MR/input
```

We have put the Raw files in the shared folder and transfer them to HDFS

```
hadoop fs -put /src/* /user/amlan/MR/input
```

Steps for Data Clean Up

1. We have used a regex expression to keep only the words and remove everything else.

2. To convert words to their basic forms we have used the library Spacy Lemmatizer.

3. We have made our own extensive custom stop-words list from nltk and various other sources.

   The code for data clean-up can be found in the file **data_cleaning.ipynb**

Steps for Word Count

1. **mapper.py:** It takes one word at a time and sends it to the reducer to count.

2. **reducer.py:** It takes the output of all the mappers, merges them and counts the occurrence of a specific word. After the that it sorts the list in descending order according to their frequencies.

3. We have run the following command to run the word count MapReduce program. Here we have mentioned the mapper, reducer and source files and mentioned the output directory.

   ```
   hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming-2.6.0-cdh5.7.0.jar \
   -file /src/mapper.py -mapper /src/mapper.py \
   -file /src/reducer.py -reducer /src/reducer.py \
   -input /user/amlan/MR/input/* -output /user/amlan/MR/output
   ```

4. We transferred the output of the MapReduce to our local directory by

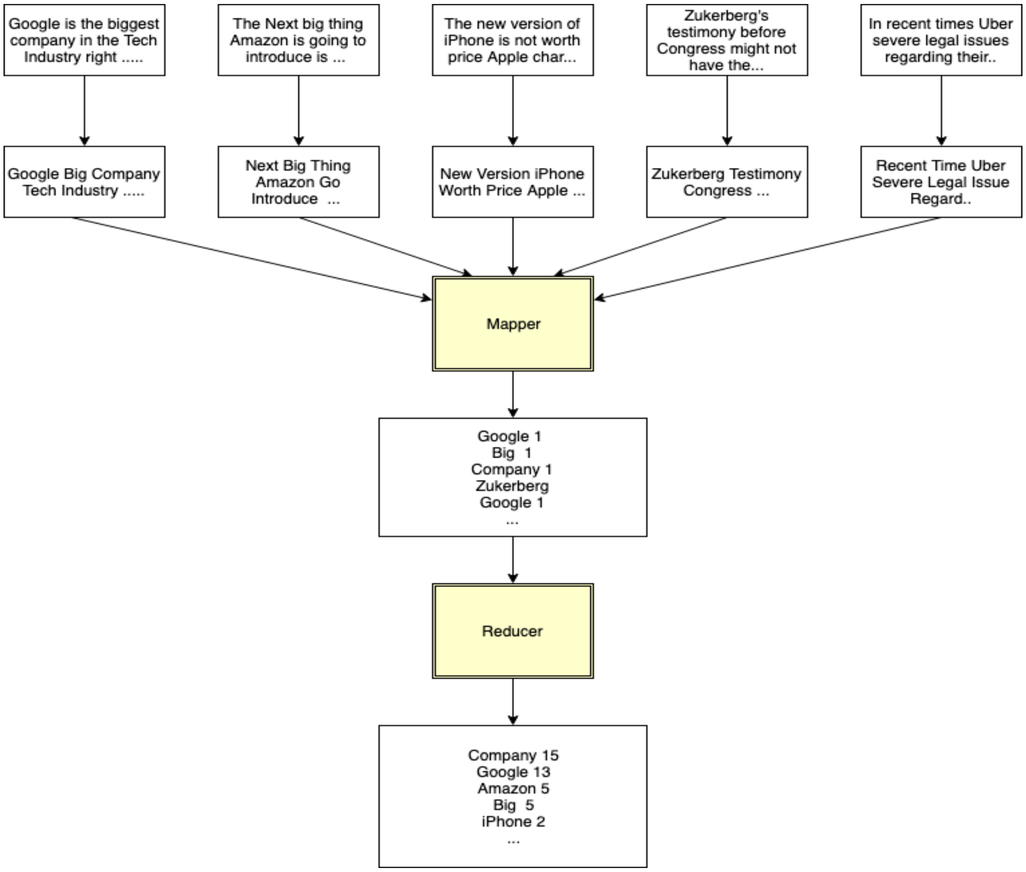
   ```
   hadoop fs -copyToLocal /user/amlan/MR/output/ /src/output
   ```

Word Count for each Subtopic for each Source

	Amazon		Apple		Google		Facebook		Uber	
NYTimes	company	1033	time	1040	time	946	time	770	company	1598
	amazon	957	apple	877	company	906	facebook	560	uber	1088
	time	891	company	877	people	740	people	543	time	935
	york	861	people	587	facebook	523	trump	529	york	834
	people	579	york	585	york	496	york	459	people	662
	trump	492	trump	401	google	488	company	435	billion	616
	city	490	change	394	call	396	american	296	city	593
	change	460	iphone	389	video	394	day	294	public	591
	president	401	china	389	write	369	president	287	lyft	511
	business	391	facebook	388	american	360	government	282	start	498
Twitter	Amazon		Apple		Google		Facebook		Uber	
	live	7552	iphone	9690	youtube	4201	facebook	2240	uber	27449
	amazon	6626	itune	4520	video	3475	day	2131	driver	6721
	twitch	5366	ipad	3648	google	2728	post	1759	lyft	4129
	stream	4038	apple	3613	live	2317	love	1579	ipo	2780
	check	2451	app	2193	watch	1773	time	1571	ride	2461
	play	2181	android	2087	check	1722	instagram	1527	eat	1975
	book	1973	watch	2086	time	1085	week	1283	drive	1870

	time	1640	play	1993	love	1072	photo	1231	car	1808
	game	1323	phone	1647	music	974	thank	1185	time	1619
	love	1247	music	1601	song	917	look	1084	billion	1594
Common Crawl	Amazon		Apple		Google		Facebook		Uber	
	company	530	company	508	company	528	facebook	1041	company	630
	amazon	525	apple	505	google	412	company	559	people	353
	people	333	app	361	app	314	app	507	million	343
	service	261	people	319	facebook	245	user	474	uber	269
	google	216	facebook	264	user	215	people	453	billion	255
	time	193	service	233	time	206	platform	271	service	254
	content	181	user	221	platform	191	time	231	business	232
	million	167	time	210	million	178	content	226	startup	228
	product	166	device	167	startup	178	google	220	time	227
	build	166	content	166	apple	169	ad	215	fund	188

Block Diagram of Word Count MapReduce


Google is the biggest company in the Tech Industry right .....
The Next big thing Amazon is going to introduce is ...
The new version of iPhone is not worth price Apple char...
Zukerberg's testimony before Congress might not have the...
In recent times Uber severe legal issues regarding their..
Google Big Company Tech Industry .....
Next Big Thing Amazon Go Introduce ...
New Version iPhone Worth Price Apple ...
Zukerberg Testimony Congress ...
Recent Time Uber Severe Legal Issue Regard..
Mapper
Google 1
Big 1
Company 1
Zukerberg
Google 1
...
Reducer
Company 15
Google 13
Amazon 5
Big 5
iPhone 2
...

Steps for Word Co-occurrence

1. **Mapper.py:** First we took Top 10 Words keywords from Word Count program and checked if they co-occur in a paragraph/tweet. If yes, we send the pair to the reducer.

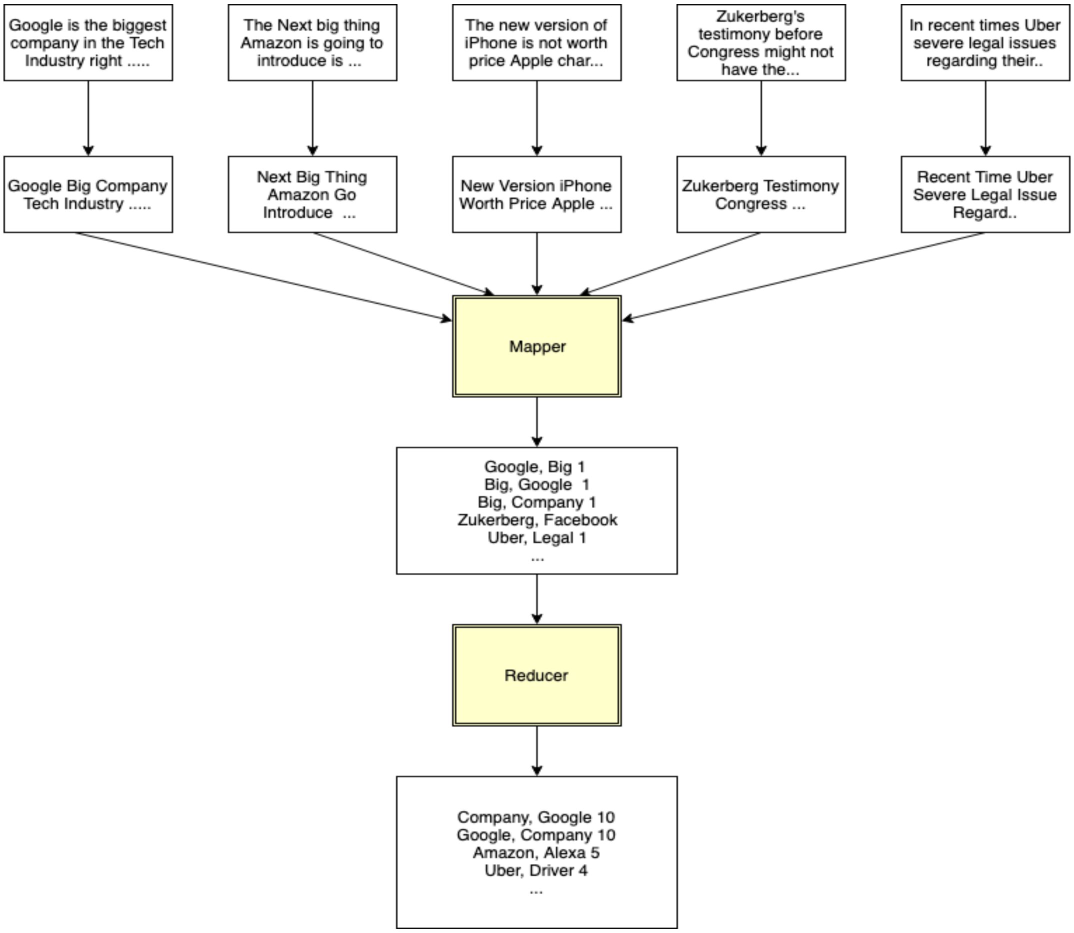
2. **reducer.py:** It takes the output of all the mappers, merges them and counts the co-occurrence of a specific pair. After the that it sorts the list in descending order according to their frequencies.

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming-2.6.0-cdh5.7.0.jar \
-file /src/mapper_co_oc.py -mapper /src/mapper_co_oc.py \
-file /src/reducer.py -reducer /src/reducer.py \
-input /user/amlan/MR/input/* -output /user/amlan/MR/output
```

3. We transferred the output of the MapReduce to our local directory by

```
hadoop fs -copyToLocal /user/amlan/MR/output/ /src/output
```

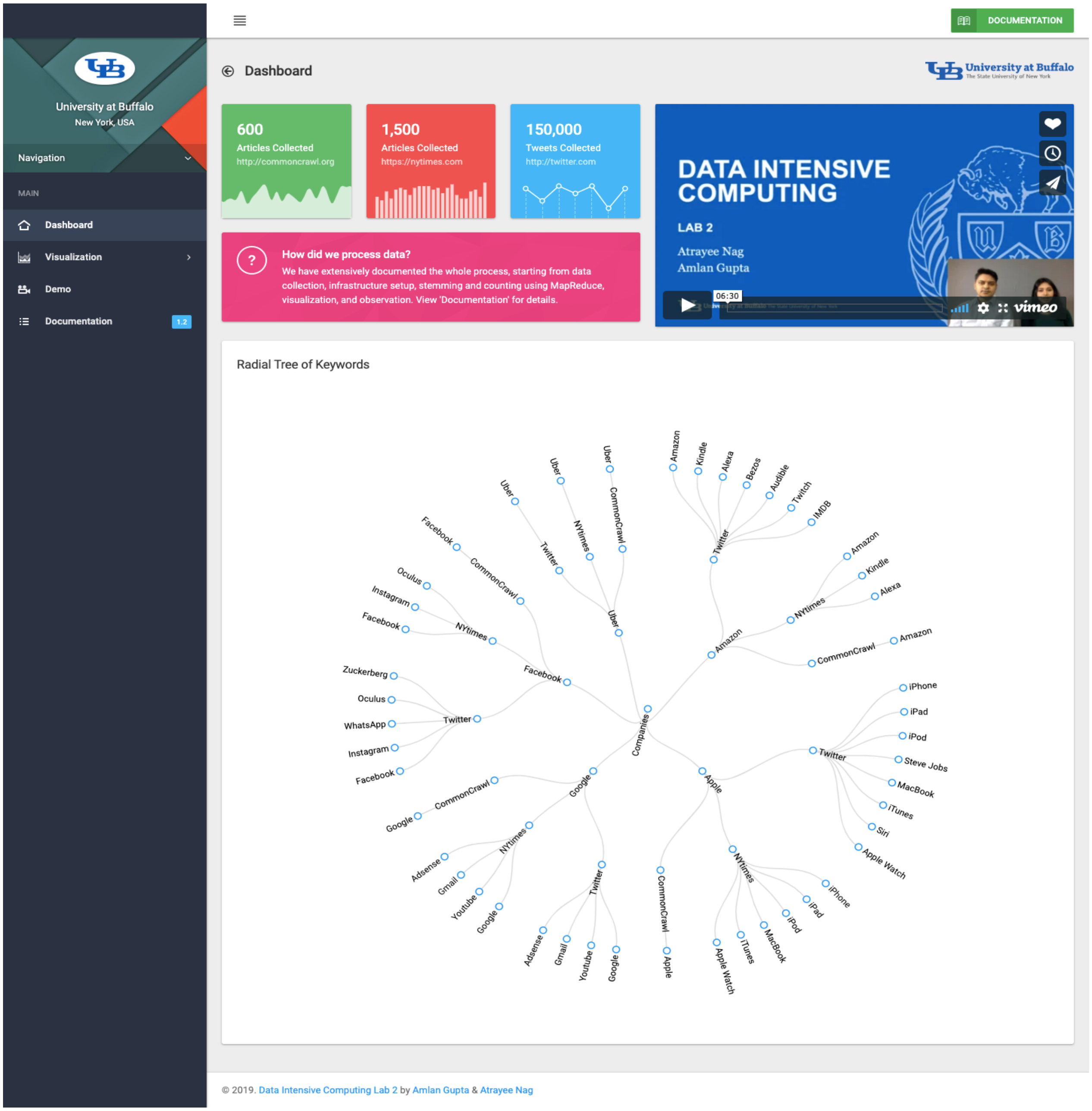
Block Diagram of Word Count MapReduce


Google is the biggest company in the Tech Industry right .....
The Next big thing Amazon is going to introduce is ...
The new version of iPhone is not worth price Apple char...
Zukerberg's testimony before Congress might not have the...
In recent times Uber severe legal issues regarding their..
Google Big Company Tech Industry .....
Next Big Thing Amazon Go Introduce ...
New Version iPhone Worth Price Apple ...
Zukerberg Testimony Congress ...
Recent Time Uber Severe Legal Issue Regard..
Mapper
Google, Big 1
Big, Google 1
Big, Company 1
Zukerberg, Facebook
Uber, Legal 1
...
Reducer
Company, Google 10
Google, Company 10
Amazon, Alexa 5
Uber, Driver 4
...

Data Visualization

We have created a web application which is powered by **d3.js** to visualize our output from MapReduce. Please note to view the application you need to run it on a server. Simplest way to do that is go to the demo folder and run the command: python3 -m http.server
We have collected the MapReduce output and converted it to appropriate json file so that it is easier for d3 to manage. We used 2 types of charts to represent our data – Word Cloud and Co-occurrence Matrix.
Deployed Url - https://amlangupta.com/misc/demo/

Dashboard



600
Articles Collected
http://commoncrawl.org

1,500
Articles Collected
https://nytimes.com

150,000
Tweets Collected
http://twitter.com

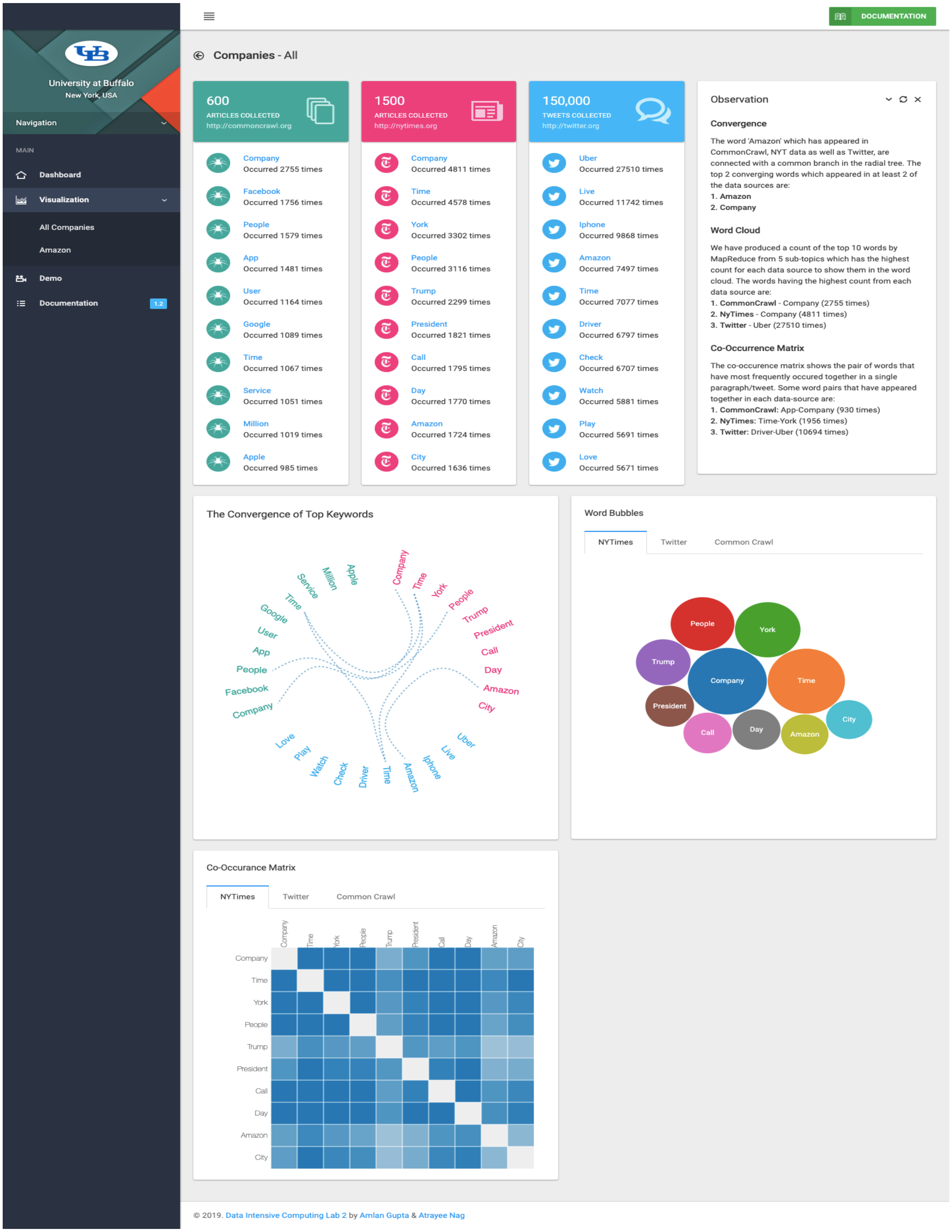
DATA INTENSIVE COMPUTING
LAB 2
Atrayee Nag
Amlan Gupta

How did we process data?
We have extensively documented the whole process, starting from data collection, infrastructure setup, stemming and counting using MapReduce, visualization, and observation. View 'Documentation' for details.

Radial Tree of Keywords

© 2019. Data Intensive Computing Lab 2 by Amlan Gupta & Atrayee Nag

Dashboard for main topic – Tech Companies



Companies - All

600
ARTICLES COLLECTED
http://commoncrawl.org

Word	Count
Company	Occurred 2755 times
Facebook	Occurred 1756 times
People	Occurred 1579 times
App	Occurred 1481 times
User	Occurred 1164 times
Google	Occurred 1089 times
Time	Occurred 1067 times
Service	Occurred 1051 times
Million	Occurred 1019 times
Apple	Occurred 985 times

1500
ARTICLES COLLECTED
http://nytimes.org

Word	Count
Company	Occurred 4811 times
Time	Occurred 4578 times
York	Occurred 3302 times
People	Occurred 3116 times
Trump	Occurred 2299 times
President	Occurred 1821 times
Call	Occurred 1795 times
Day	Occurred 1770 times
Amazon	Occurred 1724 times
City	Occurred 1636 times

150,000
TWEETS COLLECTED
http://twitter.org

Word	Count
Uber	Occurred 27510 times
Live	Occurred 11742 times
Iphone	Occurred 9868 times
Amazon	Occurred 7497 times
Time	Occurred 7077 times
Driver	Occurred 6797 times
Check	Occurred 6707 times
Watch	Occurred 5881 times
Play	Occurred 5691 times
Love	Occurred 5671 times

Observation

Convergence

The word 'Amazon' which has appeared in CommonCrawl, NYT data as well as Twitter, are connected with a common branch in the radial tree. The top 2 converging words which appeared in at least 2 of the data sources are:
1. Amazon
2. Company

Word Cloud

We have produced a count of the top 10 words by MapReduce from 5 sub-topics which has the highest count for each data source to show them in the word cloud. The words having the highest count from each data source are:
1. CommonCrawl - Company (2755 times)
2. NyTimes - Company (4811 times)
3. Twitter - Uber (27510 times)

Co-Occurrence Matrix

The co-occurence matrix shows the pair of words that have most frequently occured together in a single paragraph/tweet. Some word pairs that have appeared together in each data-source are:
1. CommonCrawl: App-Company (930 times)
2. NyTimes: Time-York (1956 times)
3. Twitter: Driver-Uber (10694 times)

The Convergence of Top Keywords

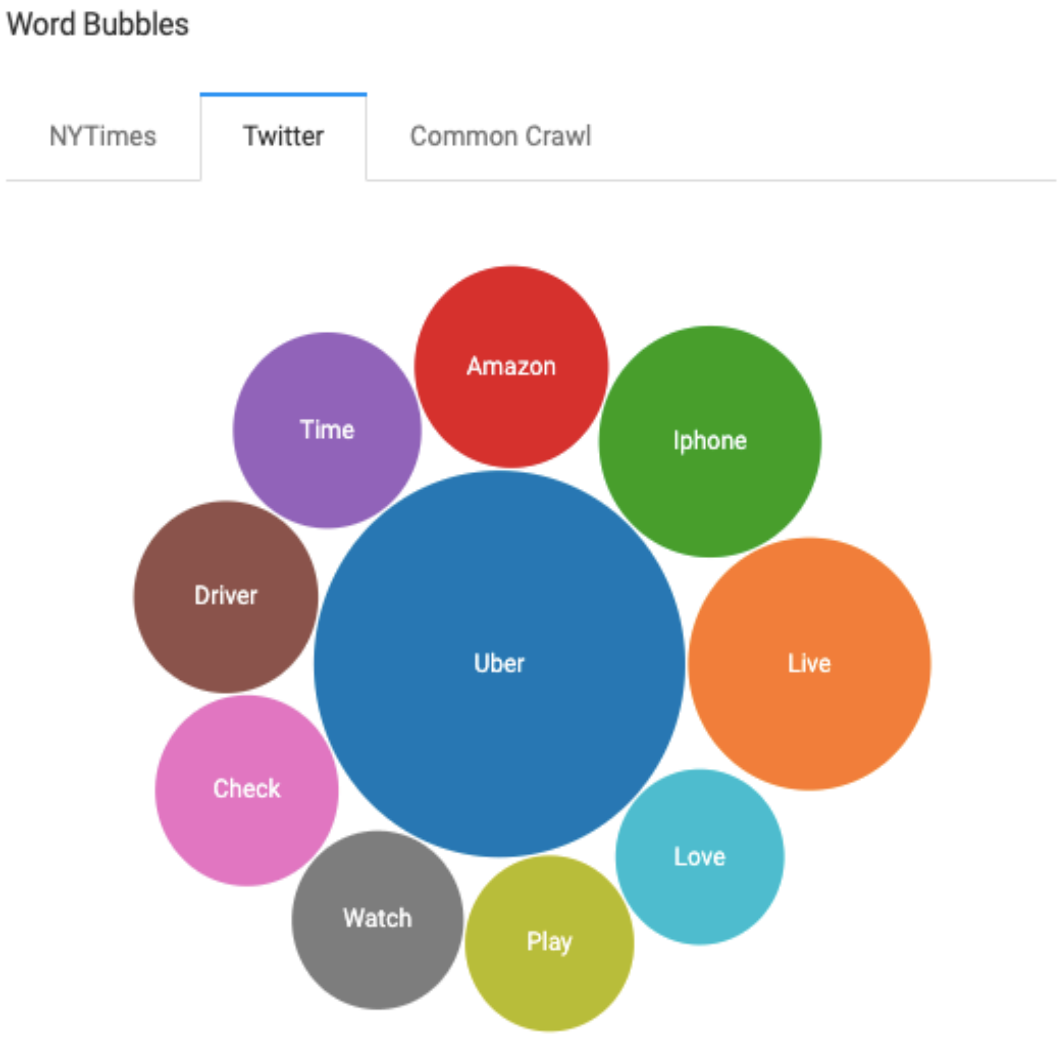
Word Bubbles

Co-Occurance Matrix

© 2019. Data Intensive Computing Lab 2 by Amlan Gupta & Atrayee Nag

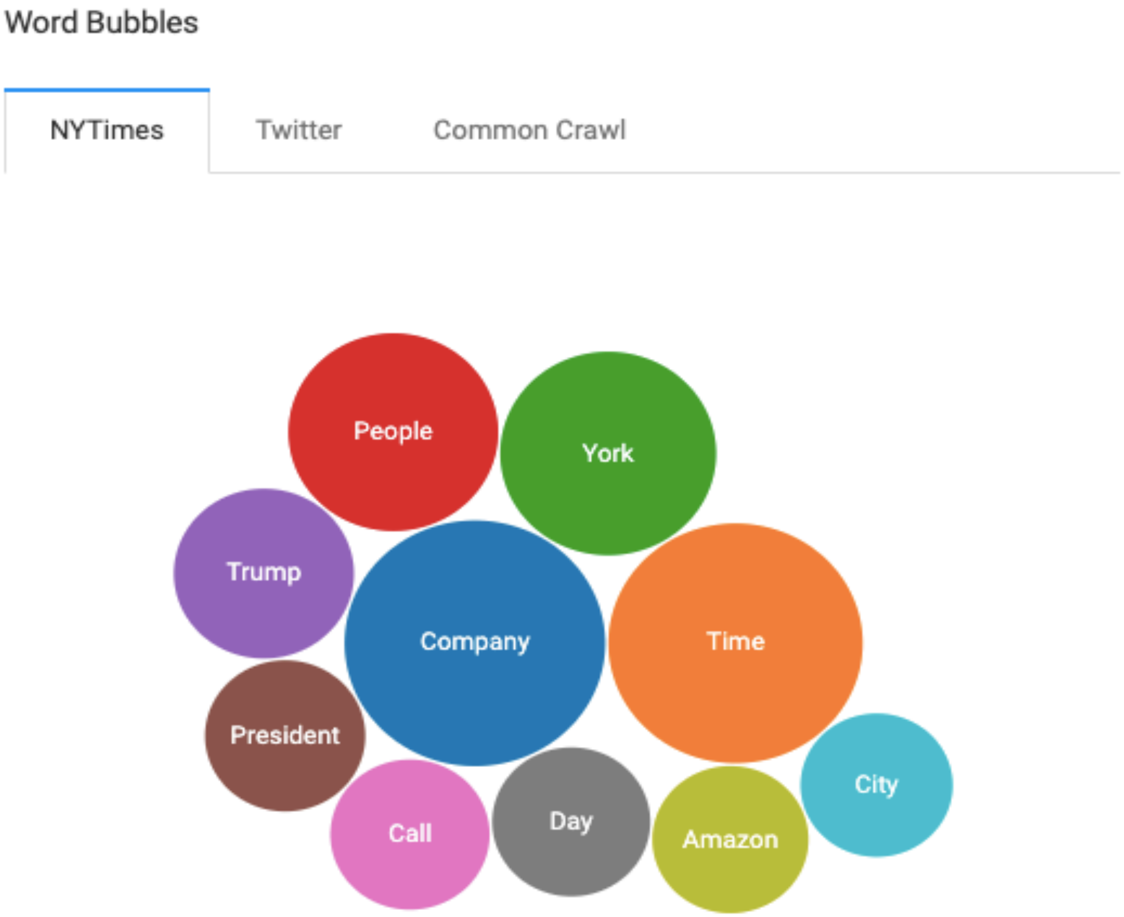
Word Count

Twitter

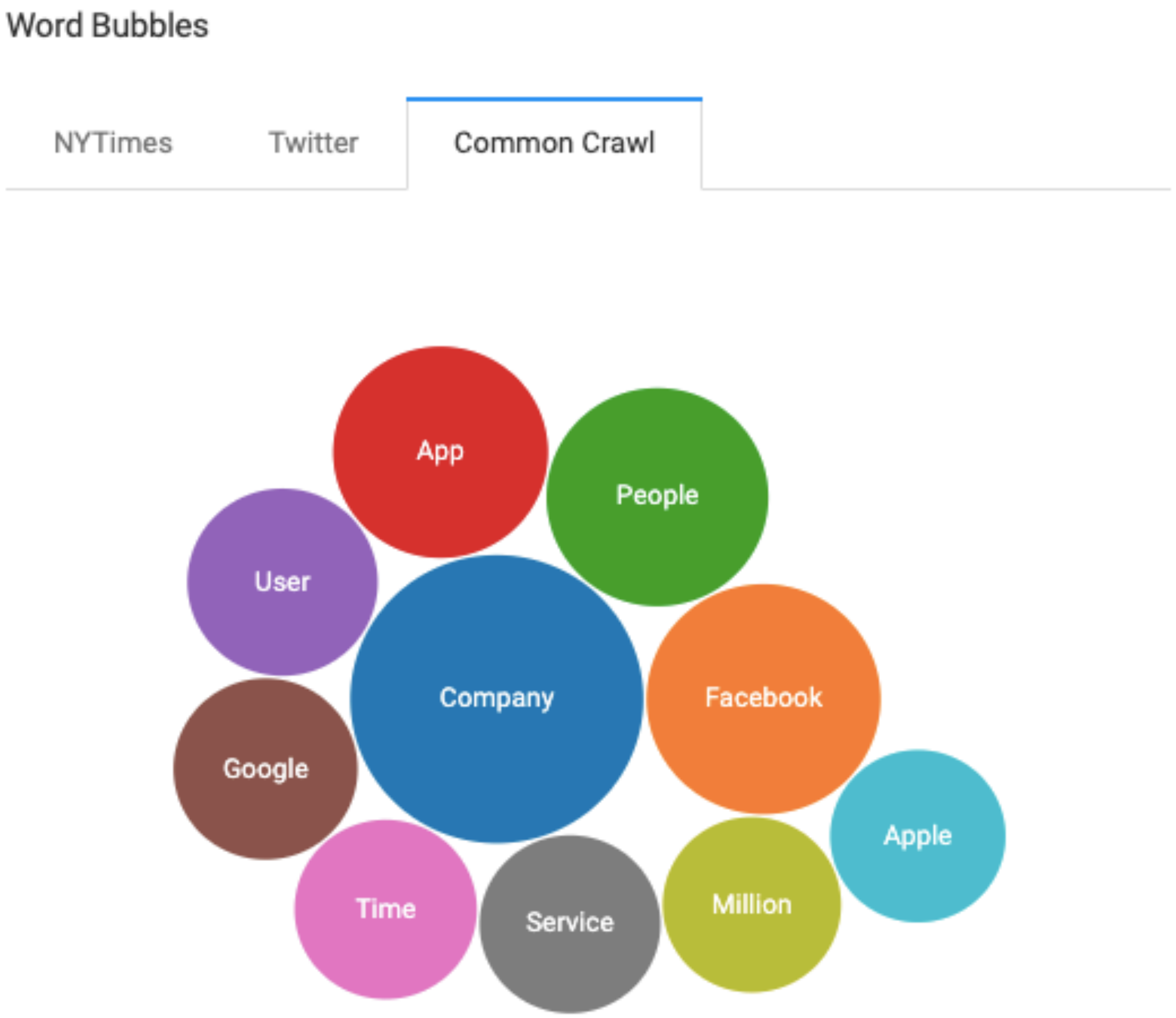


Word Bubbles

New York Times



Common Crawl



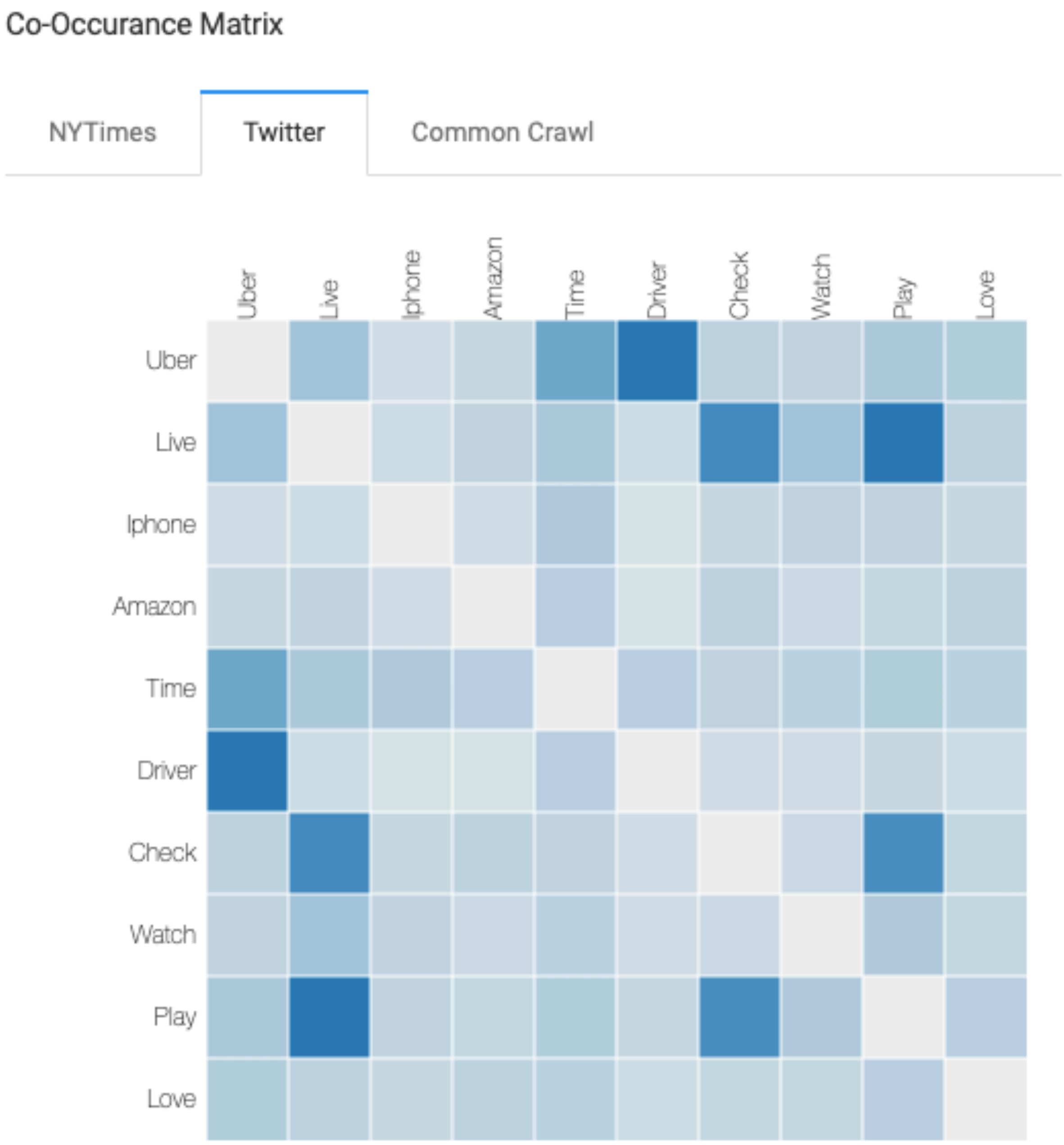
Observation

We have produced a count of the top 10 words by MapReduce from 5 sub-topics which has the highest count for each data source to show them in the word cloud. The words having the highest count from each data source are:

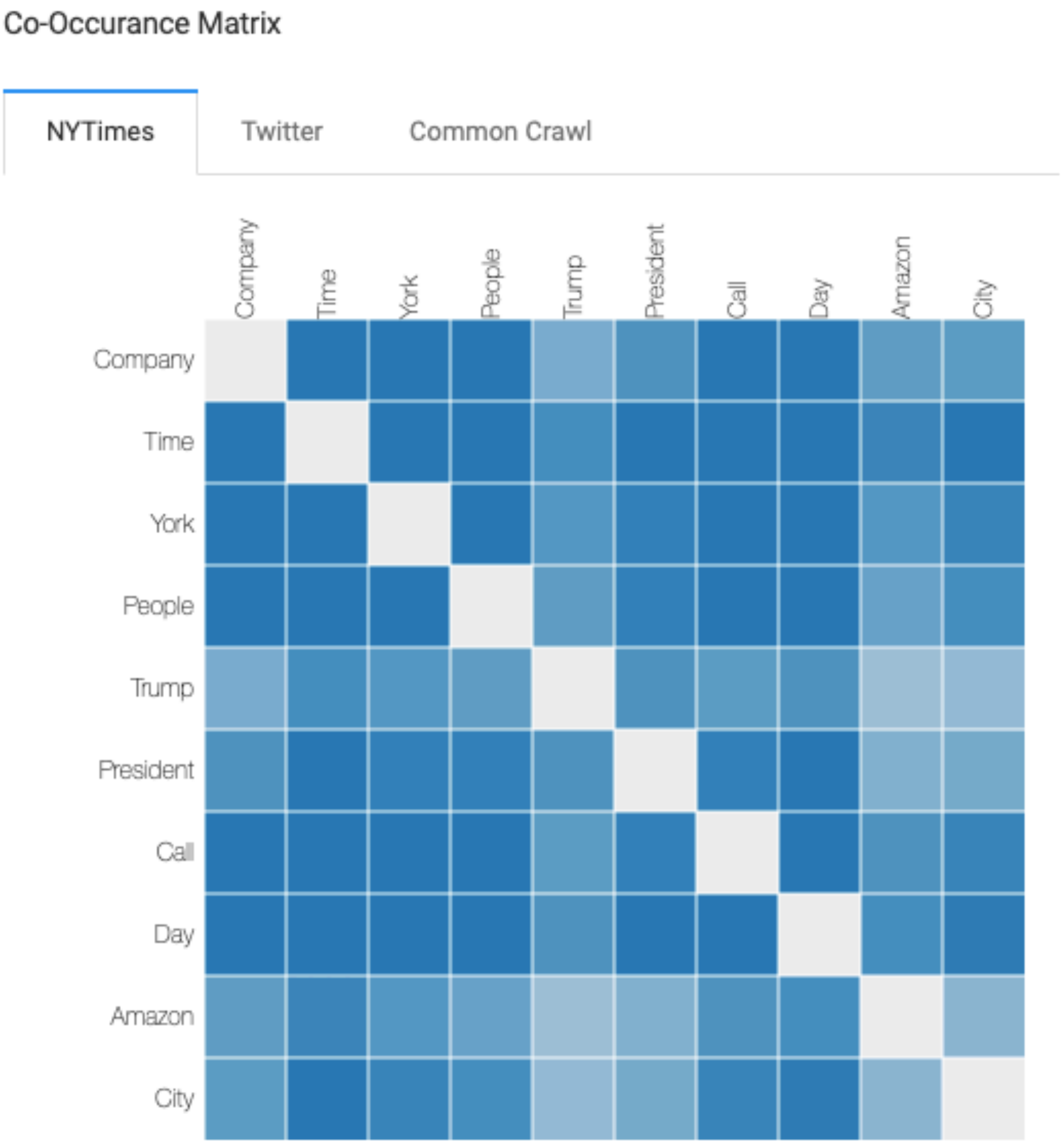
1. CommonCrawl - Company (2755 times)
2. NyTimes - Company (4811 times)
3. Twitter - Uber (27510 times)

Word Co-occurrence

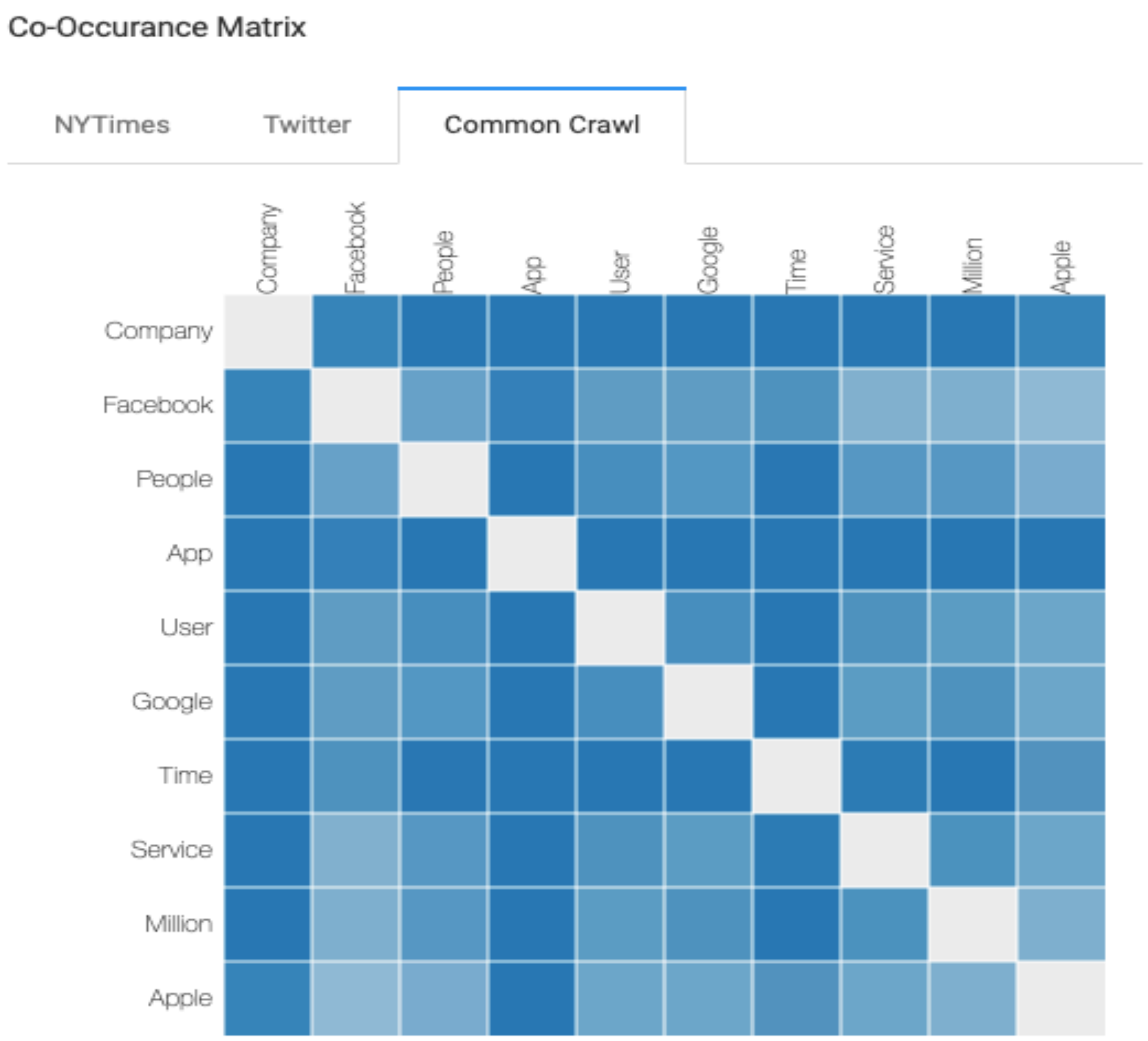
Twitter



New York Times


Co-Occurance Matrix

Common Crawl

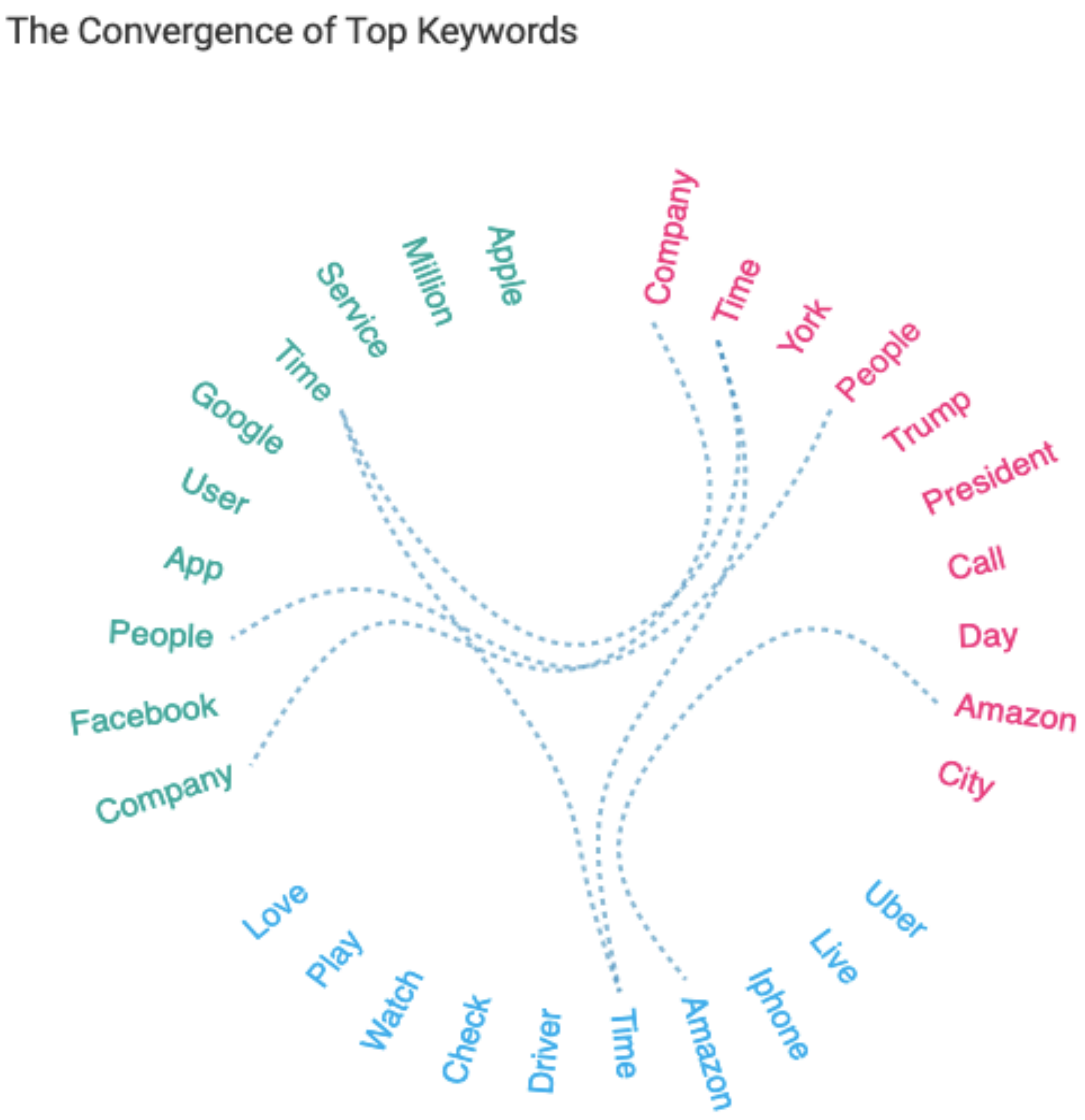


Observation

The co-occurrence matrix shows the pair of words that have most frequently occurred together in a single paragraph/tweet. Some word pairs that have appeared together in each data-source are:

1. CommonCrawl: App-Company (930 times)
2. NyTimes: Time-York (1956 times)
3. Twitter: Driver-Uber (10694 times)

Convergence


The Convergence of Top Keywords

Observation

The word 'Amazon' which has appeared in CommonCrawl, NYT data as well as Twitter, are connected with a common branch in the radial tree. The top 2 converging words which appeared in at least 2 of the data sources are:

1. Amazon
2. Company

File Structure

```
├── Report.pdf
├── demo
├── part1
│   ├── code
│   │   ├── collect_cc.ipynb
│   │   ├── collect_nyt.ipynb
│   │   └── collect_tweet.r
│   └── data
│       ├── cc
│       │   ├── amazon.csv
│       │   ├── apple.csv
│       │   ├── facebook.csv
│       │   ├── google.csv
│       │   └── uber.csv
│       ├── nyt
│       │   ├── amazon.csv
│       │   ├── apple.csv
│       │   ├── facebook.csv
│       │   ├── google.csv
│       │   └── uber.csv
│       └── twitter
│           ├── amazon.csv
│           ├── apple.csv
│           ├── facebook.csv
│           ├── google.csv
│           └── uber.csv
├── part2
│   ├── mapper.py
│   ├── reducer.py
│   └── testMR_output.csv
├── part3
│   ├── code
│   │   ├── data_cleaning.ipynb
│   │   ├── mapper.py
│   │   ├── mapper_co_occ.py
│   │   ├── mapreduce_op_to_d3_format.ipynb
│   │   ├── reducer.py
│   │   └── stop_words
│   └── data
│       ├── cleaned_raw_data
│       │   ├── cc
│       │   │   ├── amazon.csv
│       │   │   ├── apple.csv
│       │   │   ├── facebook.csv
│       │   │   │
│   │   │   ├── google.csv
│   │   │   └── uber.csv
│   │   ├── nyt
│   │   │   ├── amazon.csv
│   │   │   ├── apple.csv
│   │   │   ├── facebook.csv
│   │   │   ├── google.csv
│   │   │   └── uber.csv
│   │   └── twitter
│   │       ├── amazon.csv
│   │       ├── apple.csv
│   │       ├── facebook.csv
│   │       ├── google.csv
│   │       └── uber.csv
│   ├── co_occurance_op
│   │   ├── cc
│   │   ├── nyt
│   │   └── twitter
│   ├── top_keywords_by_count.xlsx
│   └── word_count_op
│       ├── cc
│       ├── nyt
│       └── twitter
└── video.txt
```

Reference

https://developer.twitter.com/en/apps
https://developer.nytimes.com/
https://www.bellingcat.com/resources/2015/08/13/using-python-to-mine-common-crawl/
https://github.com/d3/d3/wiki
https://riptutorial.com/pandas/example/16715/dataframe-into-nested-json-as-in-flare-js--files-used-in-d3-js