TWITTER ACTIVITY

FOR FEDERAL MEMBERS OF PARLIAMENT IN VICTORIA AND THEIR FOLLOWERS

TEAM 18

YAO PAN	777241	MIN-YING CHEN	779101
JINGFENG ZHANG	755121	SIYU FENG	745399
LIANYU ZENG	733863		

HTTP://WWW.VICMPS.COM

TWITTER ACTIVITY ANALYTICS

FOR FEDERAL MEMBERS OF PARLIAMENT IN VICTORIA AND THEIR FOLLOWERS

ACKNOWLEDGEMENTS

We would like to take this opportunity to express our special thanks and deep gratitude to Professor Richard Sinnott and Mr. Luca Morandini for their patient guidance and help during this project.

INTRODUCTION

A cloud based solution was developed using the National eResearch Collaboration Tools and Resources (NeCTAR) online infrastructure. An auto dynamic deployment script for this solution was developed in Ansible.

In total of sixteen harvesters were working in parallel and collected around 50G of Tweets from Twitter REST API and Twitter Streaming API by the harvesters. The tweets were saved to a centralized CouchDB database and replicated to the backup server.

A PHP website was created to illustrate the analytic results. Five scenarios were created to analyze the Twitter activities of the Federal Members of Parliament and their followers. Four scenarios were created to analyze the Twitter activities of the general public. The project also compares the result with data from the Australian Urban Research Infrastructure Network (Aurin), the statistic done by the Bureau of Meteorology and the research done by the Roy Morgan Research and the Galaxy Research. The analysis also covers the prediction of voting intention for the Federal Election 2016.

A backup server works alongside the main server which replicates the database from the main server. A DNS based active failover is enabled for high availability, user will be redirect to the backup server when the main server goes down, this progress is user transparent.

TWITTER

Twitter is an online social networking service that enables users to send and read short 140-character messages called "tweets".

Users access Twitter through the website interface, SMS or mobile device app. Twitter Inc. is based in San Francisco and has more than 25 offices around the world.

Twitter was created in March 2006 and launched in July 2006. The service rapidly gained worldwide popularity, with more than 100 million users posting 340 million tweets a day in 2012.The service also handled 1.6 billion search queries per day. In 2013, Twitter was one of the ten most-visited websites and has been described as "the SMS of the Internet". As of May 2015, Twitter has more than 500 million users, out of which more than 332 million are active. [1]

POSTERA CRESCAM LAUDE

THE UNIVERSITY OF MELBOURNE

SYSTEM ARCHITECTURE

Overview

The system is running on two m2.medium instances on NeCTAR. Each of them has two vCPUs, 6GB of RAM, 30G disk space and 110G volume attach to it. Apache server and CouchDB server are running on both instances, and the attached volumes are used as the storage for CouchDB and WWW root.

The Twitter REST API harvesters and the Streaming API harvesters are collecting data in parallel, the data collected are saved to the main CouchDB database and the data will be replicated to the backup CouchDB server by CouchDB replication service.

The users can use the URL http://www.vicmps.com to visit the web interface. Rage4 active failover service will resolve the domain to the IP address for the main instance (115.146.94.241) by default, if the main instance is not reachable, it will failover to the backup instance (115.146.94.24) automatically.

Figure 1 illustrates the overview of the system and its main components.

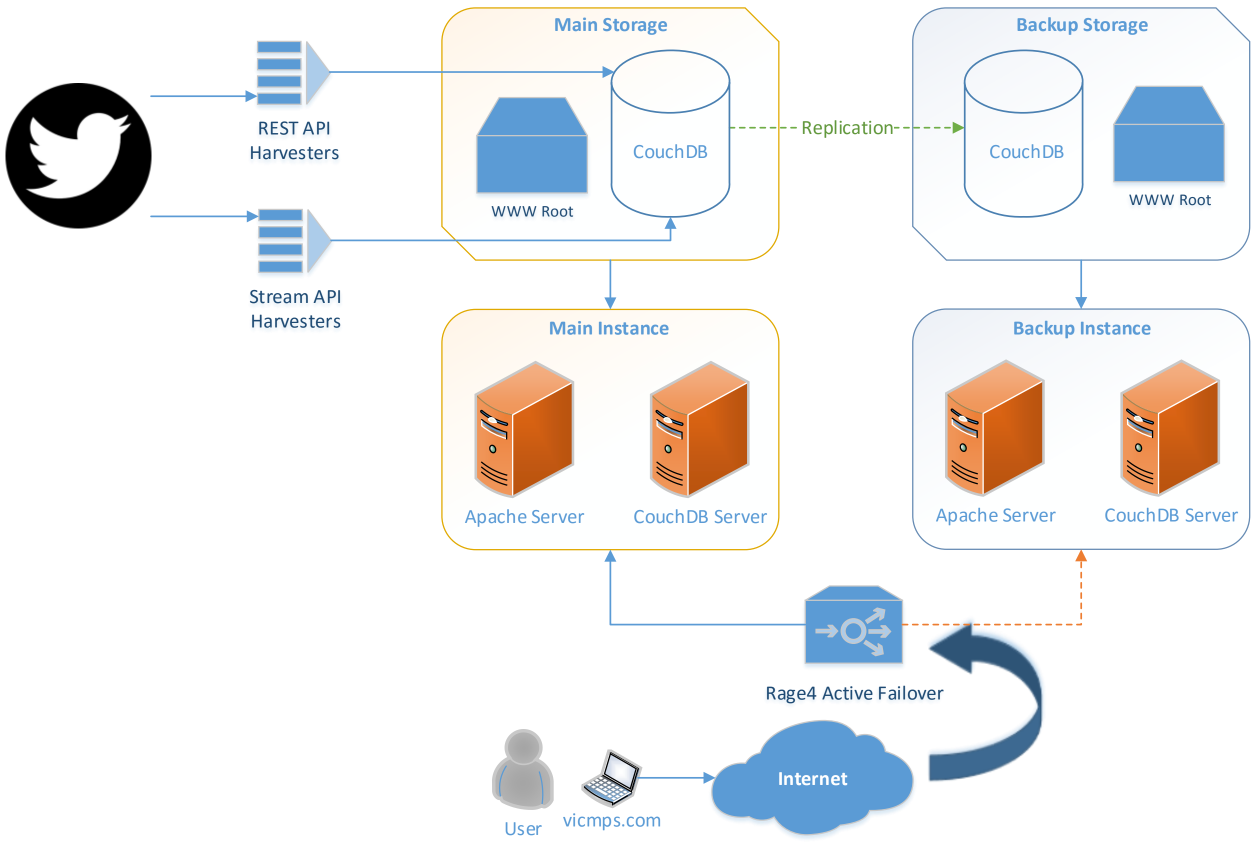
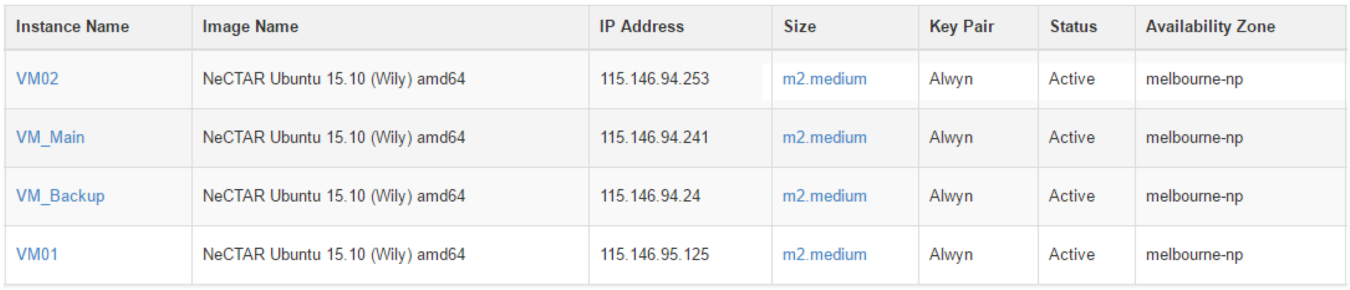


FIGURE 1 SYSTEM ARCHITECHUR OVERVIEW

NeCTAR Instances

Four m2.medium instance (2 vCPUs, 6G RAM and 30G of vHD per instance) were created in the data collection phase to run harvesters. Two of them also have a 110G volume attached for database and website storage.



Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone
VM02	NeCTAR Ubuntu 15.10 (Wily) amd64	115.146.94.253	m2.medium	Alwyn	Active	melbourne-np
VM_Main	NeCTAR Ubuntu 15.10 (Wily) amd64	115.146.94.241	m2.medium	Alwyn	Active	melbourne-np
VM_Backup	NeCTAR Ubuntu 15.10 (Wily) amd64	115.146.94.24	m2.medium	Alwyn	Active	melbourne-np
VM01	NeCTAR Ubuntu 15.10 (Wily) amd64	115.146.95.125	m2.medium	Alwyn	Active	melbourne-np

TABLE 1 INSTANCES ON NECTAR

All instances were built on NeCTAR Ubuntu 15.10 (Wily) amd64 image, and the following packages were installed on each instance,

Package Name	Version
Apache2	2.4.12
CouchDB	1.6.0
curl	7.43.0
Java	1.8.0_77
php5	5.6.11
php5-cli	5.6.11
php5-curl	5.6.11
php5-imagick	3.3.0
php5-mcrypt	5.4.6
libapache2-mod-php5	5.6.11
tux	2.0-3
vim	2:7.4.712

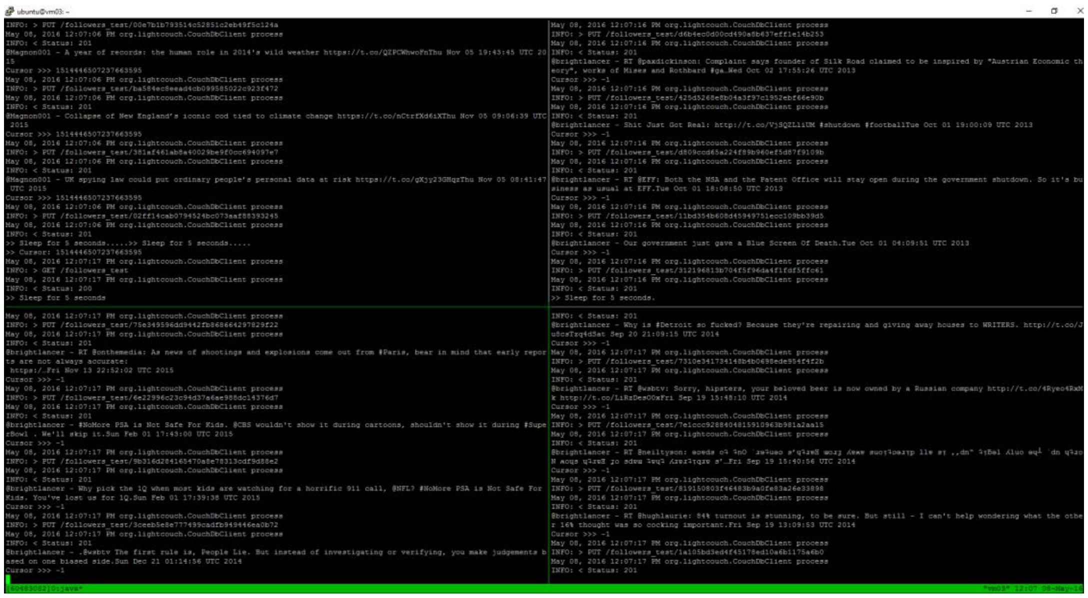
TABLE 2 PACKAGES INSTALLED ON NECTAR INSTANCE

Volumes were attached to instance VM_Main and VM_Backup, and mounted to /home/ubuntu/volume. CouchDB database and view files were saved in /home/ubuntu/volume/database folder, and the website is hosted in /home/ubuntu/volume/web folder.

Harvesters

In the data collection phase, sixteen harvesters were running on the 4 instances to collect the data from Twitter API. Each was running in a tmux terminal multiplexer session so that the harvesters could running even when the terminal shell was closed. The screenshot 1 illustrates four REST API Harvesters were running in four tmux sessions on one instance.

The Tweets collected will be preprocessed in harvester and saved to CouchDB. The details will be discussed in the Harvester section in this report.



ubuntu@vm03: ~

```
INFO: > PUT /followers_test/00e7b1b793514c52851c2eb49f5c124a
May 08, 2016 12:07:06 PM org.lightcouch.CouchDbClient process
INFO: < Status: 201
@Magnon001 - A year of records: the human role in 2014's wild weather https://t.co/QZPCWhwoFnThu Nov 05 19:43:45 UTC 20
15
Cursor >>> 1514446507237663595
May 08, 2016 12:07:06 PM org.lightcouch.CouchDbClient process
INFO: > PUT /followers_test/ba584ec8eead4cb099585022c923f472
May 08, 2016 12:07:06 PM org.lightcouch.CouchDbClient process
INFO: < Status: 201
@Magnon001 - Collapse of New England's iconic cod tied to climate change https://t.co/nCtrfXd6iXThu Nov 05 09:06:39 UTC
 2015
Cursor >>> 1514446507237663595
May 08, 2016 12:07:06 PM org.lightcouch.CouchDbClient process
INFO: > PUT /followers_test/381af461ab8a40029be9f0cc694097e7
May 08, 2016 12:07:06 PM org.lightcouch.CouchDbClient process
INFO: < Status: 201
@Magnon001 - UK spying law could put ordinary people's personal data at risk https://t.co/gXjy23GHqzThu Nov 05 08:41:47
 UTC 2015
Cursor >>> 1514446507237663595
May 08, 2016 12:07:06 PM org.lightcouch.CouchDbClient process
INFO: > PUT /followers_test/02ff14cab0794524bc073aaf88393245
May 08, 2016 12:07:06 PM org.lightcouch.CouchDbClient process
INFO: < Status: 201
>> Sleep for 5 seconds.....>> Sleep for 5 seconds.....
>> Cursor: 1514446507237663595
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: > GET /followers_test
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: < Status: 200
>> Sleep for 5 seconds
```

```
May 08, 2016 12:07:16 PM org.lightcouch.CouchDbClient process
INFO: > PUT /followers_test/d6b4ec0d00cd490a8b637eff1e14b253
May 08, 2016 12:07:16 PM org.lightcouch.CouchDbClient process
INFO: < Status: 201
@brightlancer - RT @paxdickinson: Complaint says founder of Silk Road claimed to be inspired by "Austrian Economic th
eory", works of Mises and Rothbard #ga…Wed Oct 02 17:55:26 UTC 2013
Cursor >>> -1
May 08, 2016 12:07:16 PM org.lightcouch.CouchDbClient process
INFO: > PUT /followers_test/425d5268e8b04a3f97c1952ebf66e90b
May 08, 2016 12:07:16 PM org.lightcouch.CouchDbClient process
INFO: < Status: 201
@brightlancer - Shit Just Got Real: http://t.co/VjSQZLliUM #shutdown #footballTue Oct 01 19:00:09 UTC 2013
Cursor >>> -1
May 08, 2016 12:07:16 PM org.lightcouch.CouchDbClient process
INFO: > PUT /followers_test/d809ccd65a224f89b960ef5d87f9109b
May 08, 2016 12:07:16 PM org.lightcouch.CouchDbClient process
INFO: < Status: 201
@brightlancer - RT @EFF: Both the NSA and the Patent Office will stay open during the government shutdown. So it's bu
siness as usual at EFF.Tue Oct 01 18:08:50 UTC 2013
Cursor >>> -1
May 08, 2016 12:07:16 PM org.lightcouch.CouchDbClient process
INFO: > PUT /followers_test/11bd354b608d45949751ecc109bb39d5
May 08, 2016 12:07:16 PM org.lightcouch.CouchDbClient process
INFO: < Status: 201
@brightlancer - Our government just gave a Blue Screen Of Death.Tue Oct 01 04:09:51 UTC 2013
Cursor >>> -1
May 08, 2016 12:07:16 PM org.lightcouch.CouchDbClient process
INFO: > PUT /followers_test/312196813b704f5f96da4f1fdf5ffc61
May 08, 2016 12:07:16 PM org.lightcouch.CouchDbClient process
INFO: < Status: 201
>> Sleep for 5 seconds.
```

```
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: > PUT /followers_test/75e349596dd9442fb868664297829f22
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: < Status: 201
@brightlancer - RT @onthemedia: As news of shootings and explosions come out from #Paris, bear in mind that early repor
ts are not always accurate:
 https:/…Fri Nov 13 22:52:02 UTC 2015
Cursor >>> -1
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: > PUT /followers_test/6e22996c23c94d37a6ae988dc14376d7
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: < Status: 201
@brightlancer - #NoMore PSA is Not Safe For Kids. @CBS wouldn't show it during cartoons, shouldn't show it during #Supe
rBowl . We'll skip it.Sun Feb 01 17:43:00 UTC 2015
Cursor >>> -1
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: > PUT /followers_test/9b316d284165470a8e78313cdf9d88e2
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: < Status: 201
@brightlancer - Why pick the 1Q when most kids are watching for a horrific 911 call, @NFL? #NoMore PSA is Not Safe For
Kids. You've lost us for 1Q.Sun Feb 01 17:39:38 UTC 2015
Cursor >>> -1
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: > PUT /followers_test/3ceeb5e8e777499cadfb949446ea0b72
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: < Status: 201
@brightlancer - .@wsbtv The first rule is, People Lie. But instead of investigating or verifying, you make judgements b
ased on one biased side.Sun Dec 21 01:14:56 UTC 2014
Cursor >>> -1
```

```
INFO: < Status: 201
@brightlancer - Why is #Detroit so fucked? Because they're repairing and giving away houses to WRITERS. http://t.co/J
u8csTxq4dSat Sep 20 21:09:15 UTC 2014
Cursor >>> -1
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: > PUT /followers_test/7310e341734148b4b0698ede954f4f2b
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: < Status: 201
@brightlancer - RT @wsbtv: Sorry, hipsters, your beloved beer is now owned by a Russian company http://t.co/4Ryeo4RxM
k http://t.co/LiRzDes0OxFri Sep 19 15:48:10 UTC 2014
Cursor >>> -1
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: > PUT /followers_test/7e1ccc9288404815910963b981a2aa15
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: < Status: 201
@brightlancer - RT @neiltyson: ǝɔɐds oʇ ʇnO ˙ɹǝʇuǝɔ s,ɥʇɹɐƎ ɯoɹɟ ʎɐʍɐ suoıʇɔǝɹıp llɐ sı ,,dn,, ʇɐɥʇ ʎluo ǝɥ┴ ˙dn ɥʇɹo
N ʍoɥs ɥʇɹɐƎ ɟo sdɐɯ ʇɐɥʇ ʎɹɐɹʇıqɹɐ s,…Fri Sep 19 15:40:56 UTC 2014
Cursor >>> -1
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: > PUT /followers_test/819150803f46483b9a0fe83a26e33898
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: < Status: 201
@brightlancer - RT @hughlaurie: 84% turnout is stunning, to be sure. But still - I can't help wondering what the othe
r 16% thought was so cocking important.Fri Sep 19 13:09:53 UTC 2014
Cursor >>> -1
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: > PUT /followers_test/1a105bd3ed4f45178ed10a6b1175a6b0
May 08, 2016 12:07:17 PM org.lightcouch.CouchDbClient process
INFO: < Status: 201
```

```
[60483082]0:java*                                                        "vm03" 12:07 08-May-16
```

SCREENSHOT 1 FOUR HARVESTERS ARE COLLECTING THE DATA IN ONE INSTANCE

CouchDB Servers

There are two CouchDB servers running on VM_Main (115.146.94.241:5984) and VM_Backup (115.146.94.24:5984). The database files and view files are saved in /home/ubuntu/volume/database folder. There are five databases running on each server. The databases on VM_Main are replicated to VM_Backup for backup and failover purpose. The details will be discussed in the CouchDB section in this report.

Database Name	Purpose
politician	Tweets collected from the Members
followers	Tweets collected from the Members' followers
streaming	Tweets collected from the Streaming API
non_english_area	Language data from Aurin
income_area	Income data from Aurin

TABLE 3 DATABASE LIST

Apache Servers

There are two Apache servers running on VM_Main (115.146.94.241) and VM_Backup (115.146.94.24). The web application is hosted on both servers for failover purpose.

Domain and DNS

Domain name vicmps.com was registered for this project. The DNS is hosted in Rage4, DNS based active failover is enabled for A record. The details will be discussed in the Error Handling section in this report.

Version Control

A Github private repositories was used for version control purpose.

HARVESTER

Overview

There are two types of harvesters were collecting data from Twitter API, the REST API harvesters and the Streaming API harvesters. Both of the harvesters were developed in JAVA and used **Twitter4J** library to access the Twitter APIs and used the **lightcouch** library to operate the CouchDB. The data collected will be processed in the harvester and saved in the CouchDB.

Rate Limits

Twitter REST APIs have API Rate Limits. Rate limits are divided into 15 minute intervals. Additionally, all endpoints require authentication, so there is no concept of unauthenticated calls and rate limits. There are two initial buckets available for GET requests: 15 calls every 15 minutes, and 180 calls every 15 minutes. [2] Once any one of the limits is reached, the application will be disconnected from the Twitter API.

REST API Harvester

The REST API harvesters use the following APIs to collect the Tweets.

- GET users/show, to return a variety of information about the user specified by the required user_id. [2]
- GET followers/ids, to return a cursored collection of user IDs for every user following the specified user, results are ordered with the most recent following first and are given in groups of 100 user IDs and multiple "pages" of results is navigated through using the next_cursor value in subsequent requests. [2]
- GET statuses/user_timeline, to collect most recent Tweets posted by the user. This method can only return up to 3200 of a user's most recent Tweets. [2]

The challenge is the GET statuses/user_timeline rate limits, which is 180 requests per 15 minutes. Our solution is to break the timeline to pages (200 Tweets per page), and put the thread to sleep for 5 seconds after 200 Tweets on each page are collected.

Reverse Geocoding

The Harvester uses Daniel Glasson's OfficeReverseGeocode library (https://github.com/AReallyGoodName/OfflineReverseGeocode) and placenames dump file from Geonames.org (http://download.geonames.org/export/dump/) to do offline reverse geocoding.

REST API HARVESTER - VICMPS.JAR

There are 37 Members of Parliament in Victoria, 28 of which have an active Twitter account. The REST API Harvester VicMPs.jar collected data from the 28 Members of Parliament's timeline, due to the Twitter's limits, only up to 3200 Tweets were collected from each Member.

The Harvester finds the Twitter user's timeline by Twitter ID, then breaks the timelines into "pages", there are 200 Tweets on each page. The Harvester processes the Tweets one by one, analyze the sentiment, and put the result in "emotion2" field. The Harvester also processes the "status.createAt" field, gets the day, month, year and time, and save in relevant fields for fast view processing. If the Tweet is geo enabled, the Harvester also does reverse geocoding based on "status.geoLocation", and save the result in "city" field. The Tweet will be saved in "politician" database after processing. After the Harvester processed all 200 Tweets in one page, it will pause 5 seconds and then process the next page until all pages are processed. The figure 2 illustrates the flowchart for Harvester VicMPs.

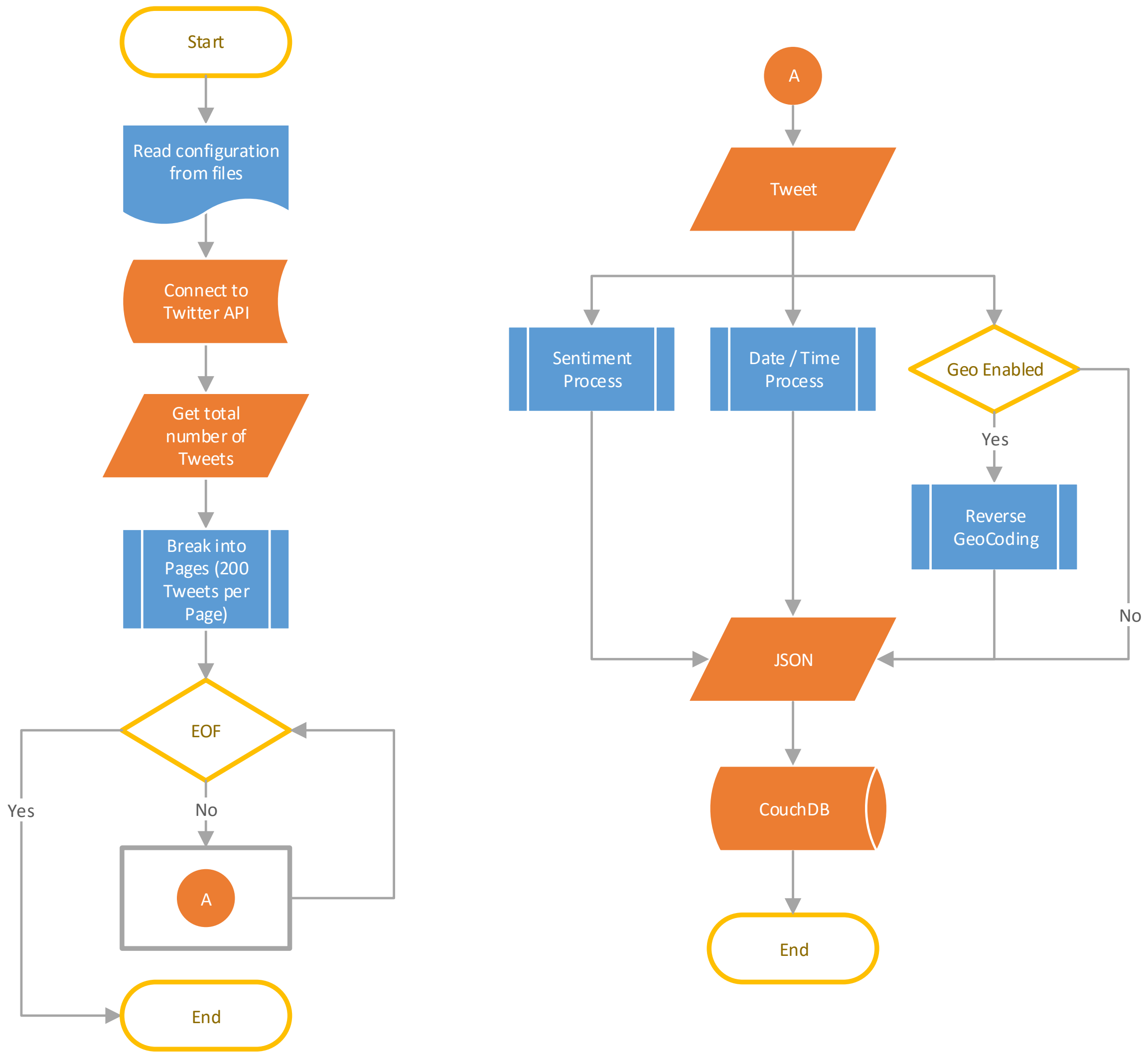


FIGURE 2 FLOWCHART FOR HARVESTER VICMPS

REST API HARVESTER - VICMPFOLLOWERS.JAR

We selected 14 Members who has the largest number of followers in their party. Five Members are from the Australian Labor Party, five Members are from Liberal Party of Australia, two Members are from The Nationals, one Member is from the Australian Greens and one independent Member. The REST API Harvester VicMPFollowers.jar collected data from the followers' timeline of the 14 Members of Parliament, 400 Tweets were collected from each follower's timeline.

Name	Party	Number of Followers
Hon Bill Shorten MP	Australian Labor Party	137,000
Hon Richard Marles MP	Australian Labor Party	14,600
Hon Catherine King MP	Australian Labor Party	12,600
Mr Tim Watts MP	Australian Labor Party	11,100
Hon David Feeney MP	Australian Labor Party	6,407
Hon Greg Hunt MP	Liberal Party of Australia	27,800
Hon Kevin Andrews MP	Liberal Party of Australia	22,300
Hon Andrew Robb AO, MP	Liberal Party of Australia	22,200
Hon Josh Frydenberg MP	Liberal Party of Australia	15,300
Hon Kelly O'Dwyer MP	Liberal Party of Australia	14,100
Hon Darren Chester MP	The Nationals	5,416
Mr Andrew Broad MP	The Nationals	2,060
Mr Adam Bandt MP	Australian Greens	92,200
Ms Cathy McGowan AO, MP	Independent	16,600

TABLE 4 LIST OF 14 MEMBERS WE COLLECTED DATA FROM THEIR FLLOWRS

The Harvester finds the Twitter user's followers by Twitter ID, then breaks the followers into "groups", there are 100 followers in each group. The Harvester processes the follower's timeline one by one, same as the Harvester VicMPs.jar. The Tweets will also be tagged with the name of the party and the Member. The Tweet will be saved in "followers" database after processing. The Harvester will navigate to the next "group" using the next_cursor value in subsequent requests. The figure 3 illustrates the flowchart for Harvester VicMPFollowers.

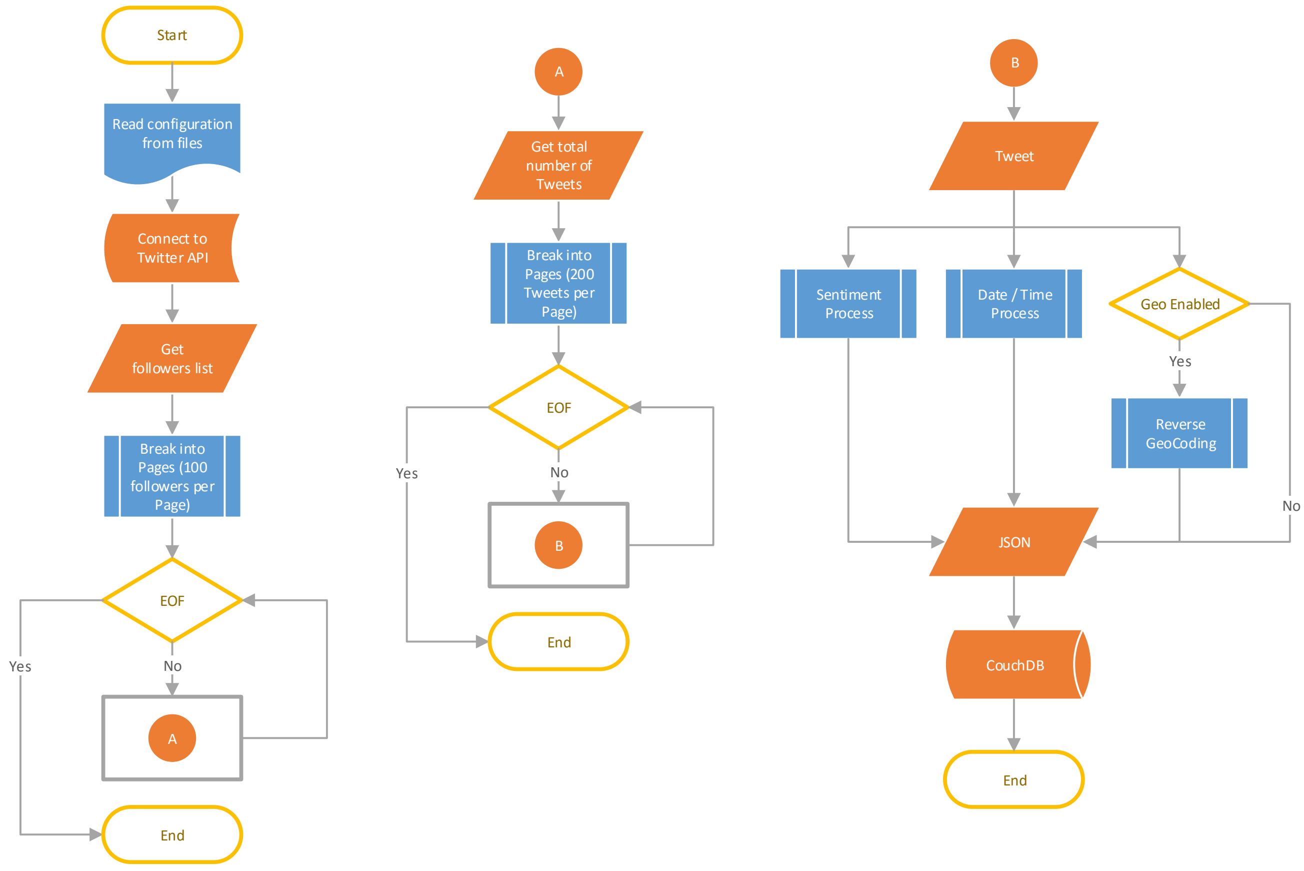


FIGURE 3 FLOWCHART FOR HARVESTER VICMPFOLLOWERS

Streaming API Harvester

The Streaming API Harvester uses Twitter Steaming APIs to collect Tweets from Twitter Public Stream. Currently, there is no rate limits for Twitter Streaming APIs.

STREAMING API HARVESTER – STEAMING.JAR

The Streaming Harvester listens to the Twitter Steaming API and collect the Tweets created in Australia from Twitter Public Stream. When a Tweet is collected it will be processed in the same way as the REST API Harvester. The Tweet will be saved in "streaming" database after processing.

The figure 4 illustrates the flowchart for Harvester Streaming.

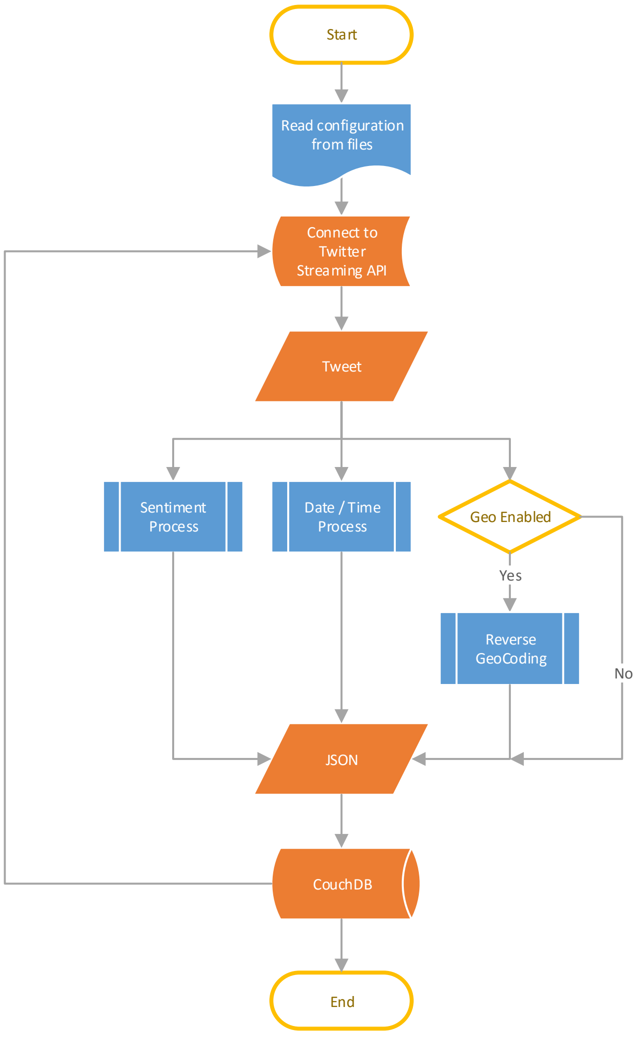


FIGURE 4 FLOWCHART FOR HARVESTER STREAMING

Deployment

In the data collection phase, there was one REST API Harvester VicMPs running on instance VM02 to collect the data from Members' timelines. Fourteen REST API Harvester VicMPFollowers were deployed on Instance VM_Main, VM_Backup, VM01 and VM02 to collect the data from the Members' followers' timelines. One Streaming API Harvester Streaming was also running on instance VM02 to collect the data from the Twitter Public Stream.

Data Collected

- In total of **62,214** Tweets were collected from the Member's timelines. Table 5 illustrates a list of the Members who has a Twitter account, and numbers of Tweets collected from each account.

Name	Party	Screen Name	Tweets	Collected
Hon Kevin Andrews MP	Liberal Party of Australia	@kevinandrewsmp	1480	1478
Mr Adam Bandt MP	Australian Greens	@AdamBandt	8177	3215
Hon Bruce Billson MP	Liberal Party of Australia	@BruceBillsonMP	3273	3214
Mr Andrew Broad MP	The Nationals	@broad4mallee	450	443
Hon Anthony Byrne MP	Australian Labor Party	@AnthonyByrne_MP	2093	2083
Hon Darren Chester MP	The Nationals	@DarrenChesterMP	5759	3240
Ms Lisa Chesters MP	Australian Labor Party	@LMChesters	6453	3197
Hon David Feeney MP	Australian Labor Party	@Feeney4Batman	9178	3247
Hon Josh Frydenberg MP	Liberal Party of Australia	@JoshFrydenberg	2668	2664
Mr Andrew Giles MP	Australian Labor Party	@andrewjgiles	4936	3216
Ms Sarah Henderson MP	Liberal Party of Australia	@SHendersonMP	2286	2282
Hon Greg Hunt MP	Liberal Party of Australia	@GregHuntMP	2101	2096
Hon Catherine King MP	Australian Labor Party	@CatherineKingMP	3969	3236
Hon Richard Marles MP	Australian Labor Party	@RichardMarlesMP	2391	2377
Ms Cathy McGowan AO, MP	Independent	@Indigocathy	3549	3220
Mr Rob Mitchell MP	Australian Labor Party	@RobMitchellMP	7509	3234
Hon Kelly O'Dwyer MP	Liberal Party of Australia	@KellyODwyer	876	876
Ms Clare O'Neil MP	Australian Labor Party	@ClareONeilMP	1046	1040
Hon Andrew Robb AO, MP	Liberal Party of Australia	@AndrewRobbMP	2428	2418
Ms Joanne Ryan MP	Australian Labor Party	@JoanneRyanLalor	5127	3201
Hon Bill Shorten MP	Australian Labor Party	@billshortenmp	3605	3219
Hon Tony Smith MP	Liberal Party of Australia	@TonySmithMP	0	0
Hon Sharman Stone MP	Liberal Party of Australia	@SharmanStone	902	902
Hon Dan Tehan MP	Liberal Party of Australia	@DanTehanWannon	1329	1325
Hon Kelvin Thomson MP	Australian Labor Party	@KelvinThomsonMP	1866	1849
Hon Alan Tudge MP	Liberal Party of Australia	@AlanTudgeMP	1155	1152
Ms Maria Vamvakinou MP	Australian Labor Party	@MariaVamvakinou	182	1811
Mr Tim Watts MP	Australian Labor Party	@TimWattsMP	40310	3243
Mr Jason Wood MP	Liberal Party of Australia	@JasonWood_MP	351	351

TABLE 5 AMOUNT OF TWEETS COLLECTED MEMBER OF PARLIAMENTS

- In total of **10,600,890** Tweets were collected from the Members' followers' timelines.
- In total of **215,125** Tweets were collected from the Twitter Public Stream.

COUCHDB

CouchDB is a document-oriented Non-SQL database which uses JSON to store data, JAVAScript to query and HTTP for an API. CouchDB doesn't use schema (table) to store data and relationship, it is a collection of independent documents, and each documents maintains its own data and self-contained schema. [3]

Views and List function

In this system, view part is also important to analyze data harvested from Twitter and Aurin. MapReduce function can be used to sum or count needed data, and sorted data or changed data format can be retrieved from a view by list function. There is no show function used in this system.

MAPREDUCE FUNCTION

As mentioned at the previous part, there are two parts data obtained from Twitter, the first part is all politicians' tweets, and another part is some of politicians' followers' tweets.

ATSIGN VIEW

This view is used to acquire all the mentioned userId from both politicians' tweets s and followers' tweets (only count at sign tweeted in Melbourne), then reduce these data and sort them, so the top 10 politicians' mentioned userId and followers' userId. There is a field called 'userMentionEntities' in the 'status' field acquired from Twitter data, and it is easy to check whether this tweet mentions some user and how many users are mentioned.

HASHTAG VIEW

This view is similar with the atSign view, which is also used to obtain top 10 mentioned topics from politicians' and followers' tweets (only count hashtag tweeted in Melbourne). Twitter API also provides a field named 'hashtagEntities' in 'status' of each tweet.

EMOTION_AM VIEW & EMOTION_MONTH VIEW

These two views are used to show both politicians' and followers' emotion during a day (morning, afternoon and night) and a year (12 months). Before inserting data to CouchDB, the created time and sentiment of a tweet has been analyzed. Many time data, such as 'year', 'timeOfDay', 'month', 'day' and 'dayOfMonth' are stored in each piece of data. However, because of time limitation, there are only 'timeOfDay' and 'Month' used. With these two views, we can know that politicians or followers are happier in which part of a day or a year.

EMOTION_MP_AM VIEW & EMOTION_MP_MONTH VIEW

Each politician's sentiment analysis from their tweets during a day or a year can be obtained through these two views. It is uncomplicated to know whether a politician is much happier in the morning, afternoon or at night. These data also can be compared with the data of the circumstance of his election. For example, if he got few votes in this month, then we can compare the result of this view with his circumstance, and maybe in this month most of his tweets are negative.

GEOLOCATION VIEW

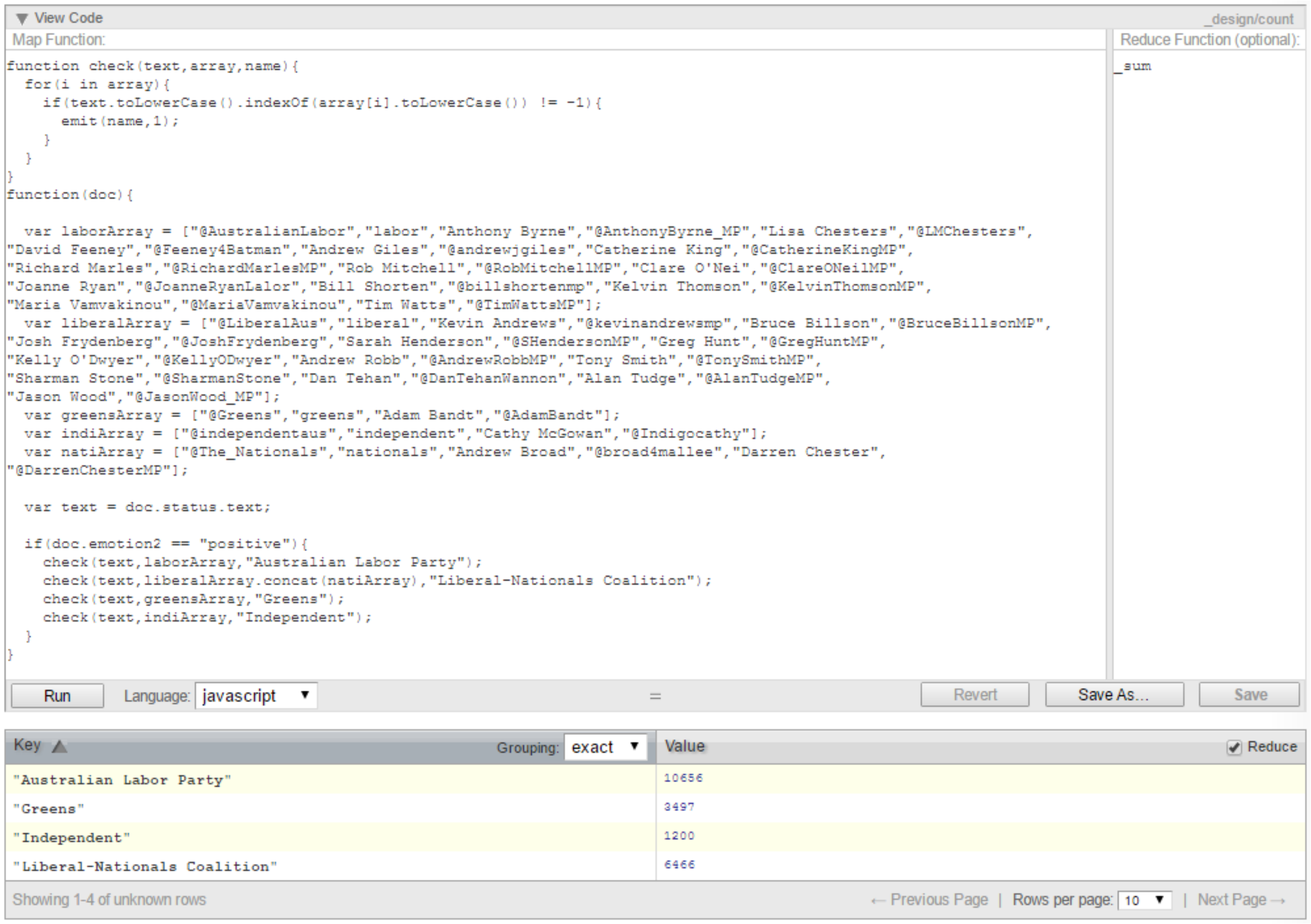
This view is only used in politician part to create heat map. There is a field named city which is added to each of data if when getting the data from Twitter. With this view, all the locations of tweets tweeted by this politician can be made to heat map in the website.

LANG VIEW

Language view is used to calculate the amount of all tweets' languages except English (EN) from all followers. With order list function, we can analyze the relationship between the most used foreign languages and the most foreigners in Victoria. Furthermore, with geoLocation, Aurin data from all suburbs can be analyzed too.

VOTING_INTENTION VIEW

This view is designed to analyze tweets from all the politicians' followers, and acquire the voting intention among four parties. When there is a positive mention (the value of sentiment analysis is positive) including all the Victoria politicians' user ids, full name of these politicians or key words of four parties, then with Reduce function, it can be obviously seen that which party is the most popular. Although the result may not 100% match the real data, it can also show the voting intention in Twitter.



Language: javascript

```
function check(text,array,name){
  for(i in array){
    if(text.toLowerCase().indexOf(array[i].toLowerCase()) != -1){
      emit(name,1);
    }
  }
}
function(doc){

  var laborArray = ["@AustralianLabor","labor","Anthony Byrne","@AnthonyByrne_MP","Lisa Chesters","@LMChesters",
"David Feeney","@Feeney4Batman","Andrew Giles","@andrewjgiles","Catherine King","@CatherineKingMP",
"Richard Marles","@RichardMarlesMP","Rob Mitchell","@RobMitchellMP","Clare O'Nei","@ClareONeilMP",
"Joanne Ryan","@JoanneRyanLalor","Bill Shorten","@billshortenmp","Kelvin Thomson","@KelvinThomsonMP",
"Maria Vamvakinou","@MariaVamvakinou","Tim Watts","@TimWattsMP"];
  var liberalArray = ["@LiberalAus","liberal","Kevin Andrews","@kevinandrewsmp","Bruce Billson","@BruceBillsonMP",
"Josh Frydenberg","@JoshFrydenberg","Sarah Henderson","@SHendersonMP","Greg Hunt","@GregHuntMP",
"Kelly O'Dwyer","@KellyODwyer","Andrew Robb","@AndrewRobbMP","Tony Smith","@TonySmithMP",
"Sharman Stone","@SharmanStone","Dan Tehan","@DanTehanWannon","Alan Tudge","@AlanTudgeMP",
"Jason Wood","@JasonWood_MP"];
  var greensArray = ["@Greens","greens","Adam Bandt","@AdamBandt"];
  var indiArray = ["@independentaus","independent","Cathy McGowan","@Indigocathy"];
  var natiArray = ["@The_Nationals","nationals","Andrew Broad","@broad4mallee","Darren Chester",
"@DarrenChesterMP"];

  var text = doc.status.text;

  if(doc.emotion2 == "positive"){
    check(text,laborArray,"Australian Labor Party");
    check(text,liberalArray.concat(natiArray),"Liberal-Nationals Coalition");
    check(text,greensArray,"Greens");
    check(text,indiArray,"Independent");
  }
}
```

```
_sum
```

Grouping: exact

Reduce ✓

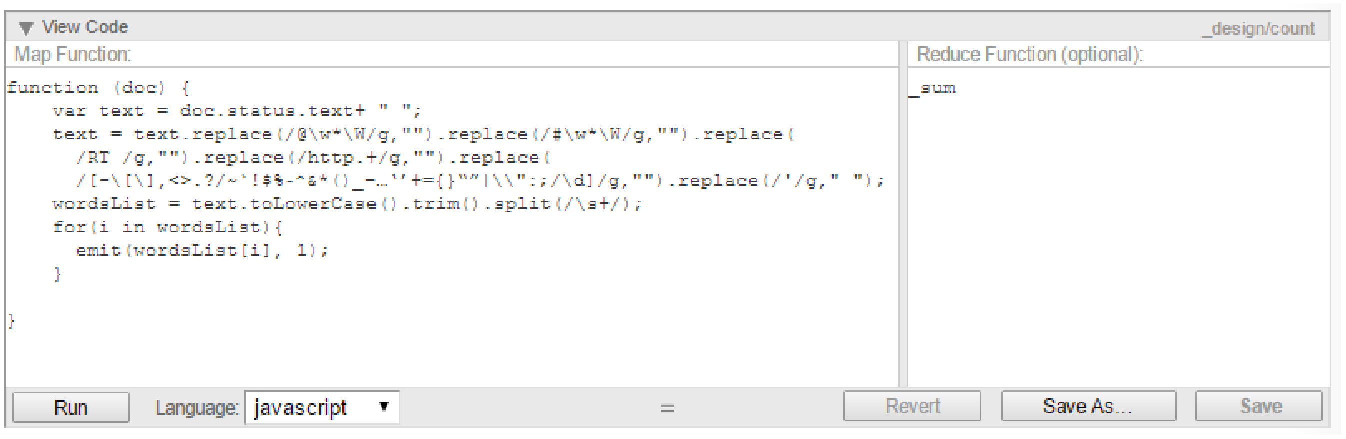
Key	Value
"Australian Labor Party"	10656
"Greens"	3497
"Independent"	1200
"Liberal-Nationals Coalition"	6466

Showing 1-4 of unknown rows ← Previous Page | Rows per page: 10 | Next Page →

SCREENSHOT 2 VOTING_INTENTION VIEW CODE AND RESULT

WORD VIEW

This view is one of the complicated views in this system. The purpose of words view is to use top 10 or top 20 words used in both politicians' and followers' tweets (only count words tweeted in Melbourne) to make word cloud. This view first cleans the string, then splits the string and get each word, finally reduce them. With order list function which will be introduced later, top 20 words can be obtained.



```
function (doc) {
    var text = doc.status.text+ " ";
    text = text.replace(/@\w*\W/g,"").replace(/#\w*\W/g,"").replace(
      /RT /g,"").replace(/http.+/g,"").replace(
      /[-\[\],<>.?/~`!$%-^&*()_-…''+={}""|\\":;/\d]/g,"").replace(/'/g," ");
    wordsList = text.toLowerCase().trim().split(/\s+/);
    for(i in wordsList){
      emit(wordsList[i], 1);
    }

}
```

```
_sum
```

SCREENSHOT 3 WORDS VIEW CODE

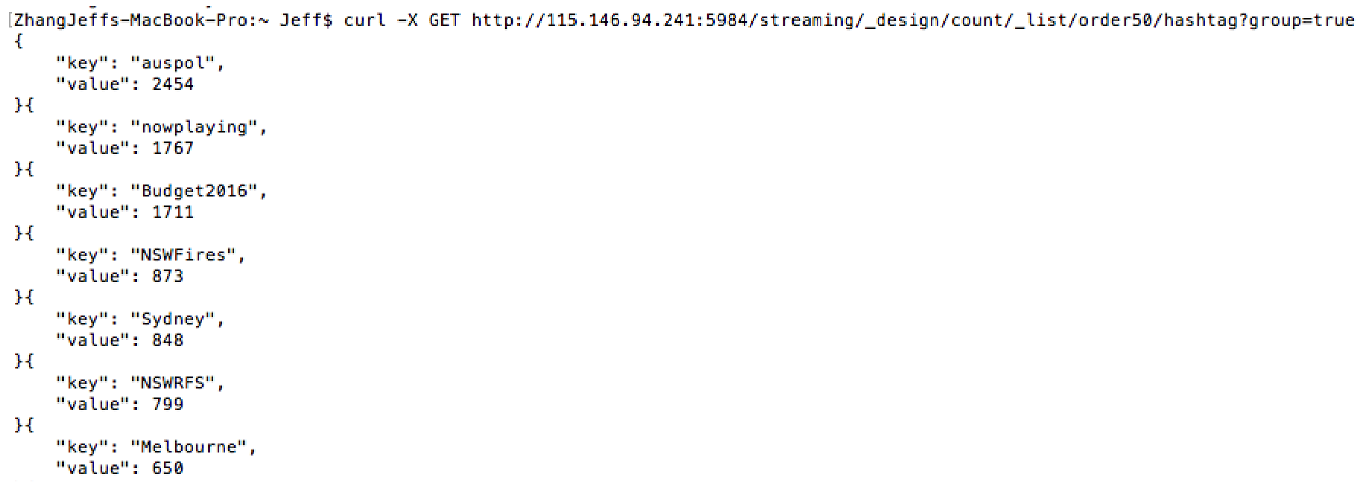
LIST FUNCTION

There are three list functions will be introduced in this report, but only one of them is used in the final system.

order

Sometimes when we get the view, there are too many keys, and we only care about those keys with high value, such as top10 or top50 values. For view function, data can be only sorted by key instead of value. For example, in word view, there are too many words used in all tweets, some of them only occur one or two times, which is meaningless for the word cloud. With order list function, views can be sorted by value.

This order list function is used with many views in this system, such as hashtag, atSign and word view.



```
[ZhangJeffs-MacBook-Pro:~ Jeff$ curl -X GET http://115.146.94.241:5984/streaming/_design/count/_list/order50/hashtag?group=true
{
    "key": "auspol",
    "value": 2454
}{
    "key": "nowplaying",
    "value": 1767
}{
    "key": "Budget2016",
    "value": 1711
}{
    "key": "NSWFires",
    "value": 873
}{
    "key": "Sydney",
    "value": 848
}{
    "key": "NSWRFS",
    "value": 799
}{
    "key": "Melbourne",
    "value": 650
```

SCREENSHOT 4 HASHTAG VIEW WITH ORDER LIST FUNCTION

addempty

In the CouchDB view, when we use an array as a key, occasionally we need some keys which has not occurred in the data, but we also want that key with value 0 to be showed in the view. For example, when we analyse politicians' emotion (normal, positive and negative) in a day (morning, afternoon and night), few MPs do not tweet at night, or there are no negative tweets in the morning. With this list function, it is convenient for our website to retrieve all the data no matter whether they are empty or not, because the keys do not occur with value 0 are also important to analyse.

week

There is also a problem with CouchDB view is that it will take large amount of time and storage to calculate the view, if the data is huge. In order to save time and space, this list function is designed to use multiple levels grouping in view to create a view one time, and list function can obtain other reduce results in one view rather than creating another view. For instance, there is a view to obtain the emotion of MPs' tweets by month and day of a week (Monday, Tuesday etc.). In this view, only month amount valued can be reduced, we can't sum all the days of a week (the emotion of a politician on Thursdays), if there is no another view or list function. Utilizing week list function can easily get the list we want instead of costing time and storage to create another view.

SENTIMENT

In this assessment, one of our purpose is to analyse the emotion of each tweet, and we analyse the emotion of each tweet before it is saved in the CouchDB.

As we know, the content of a tweet is short, and normally contains some meaningless text, for example, mentioned person (@xxx), mentioned topic (#xxx) and short link (https://xxx). In order to avoid the influence by these meaningless text, it is important to filter this information by algorithm 1.

```
Algorithm 1: Filter Meaningless Information
   Input: The content of a tweet;
   Output: The edited content that without meaningless information;
1  do
2      replace the meaningless information to blank space by two regular
3      expressions
4      [@#]([a-zA-Z0-9])+    replace to blank space     and
5      http(s)?://[a-zA-Z]+(\.[a-zA-Z]+)+/[a-zA-Z0-9]+     replace to blank space;
6      return edited text;
```

After having the clear content, we can use algorithms to analyse the emotion. The classification of emotion can be normally regarded as six parts, which are happiness, sadness, angry, disgust, fear and surprise. In this case, this application is designed to transfer tweet to the six vectors of emotion, and selects the most outstanding vector as its emotion.

On the one hand, because of the short length of the tweet and the situation that people often use several words to show their altitudes, we decide to use the comparison of the count of different types of emotion words to show the emotion of tweet, and some typical words in each type of emotion are picked out, for example, happy, lucky and awesome can show the emotion of happiness; hate and damn can show the emotion of angry. In addition, people sometimes use negation words to show different meanings, for example, no, not and don’t, and it is necessary to change the result if there is a negation word before a typical emotion word. However, our purpose is not only analyzing the emotion of tweet, but also taking advantage of these emotion to analyse the public altitude of different political parties. Thus, we synthesize the six vectors to three parts: positive, normal and negative.

On the other hand, it is the fact that people prefer to use emoji to show their altitude. Thus, we highlight the influence of emoji that it is easier to find, and it has more accurate meaning than word. As we have mentioned in the first part, there are six vectors of emotion that we can easily define the meaning the emoji into six vectors. After having the meaning of typical words and emoji, it is possible to analyse the emotion of tweet by using Algorithm 2.

```
Algorithm 2: Analyse Emotion
  Input: The content of a tweet;
  Output: The type of this tweet, normal, negative or positive;
1   Initialize Word Map Word(i).word and Word(i).emotion from .txt file;
2   Initialize Emoji Map Emoji(i).emoji and Emoji(i).emotion from .txt file;
3   Initialize Count.nega = 0, and Count.posi = 0;
4   foreach Emoji(i).emoji do
5     if tweet contains this Emoji(i).emoji then
6       change Count.nega or Count.posi according to Emoji(i).emotion;
7       delete the Emoji(i).emoji from the tweet;
8   foreach word in tweet do
9     foreach element in Word do
10      if Word(i).word equal with word then
11        change Count.nega or Count.posi according to Word(i).emotion;
12  if Count.nega > Count.posi then
13    return negative;
14  if Count.nega < Count.posi then
15    return positive;
16  return normal;
```

RESULT ANALYSIS & COMPARISON

In order to compare the analysis result of tweet with true data, we download some statistical data from Australian Urban Research Infrastructure Network (AURIN), Bureau of Meteorology (BOM) and Roy Morgan.

Firstly, average rainfall of every month is downloaded from BOM to analyse the relationship between people's emotion and weather. For the reason that the BOM don not have the data of 2015 or 2016, we use the mean rainfall data of 2014 as our dataset (Figure 5).

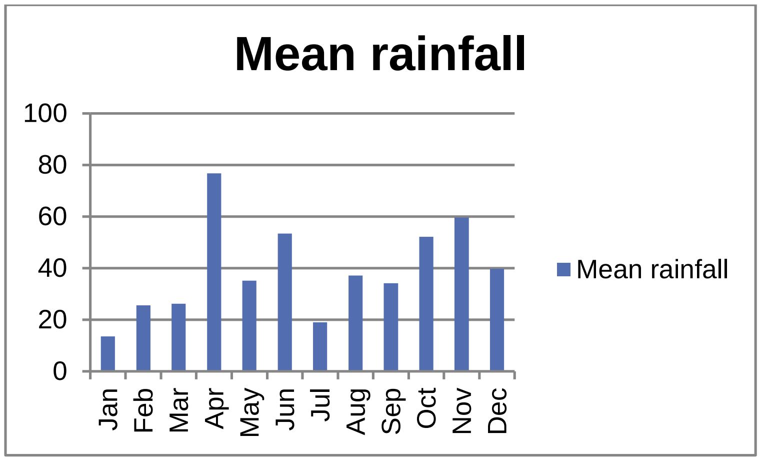


FIGURE 5 MEAN RAINFALL IN 2014 FROM BOM

Secondly, because of the purpose of our application that statistics prefer rating of each party, we get the real time prefer rating of the two parties ALP and L-NP in Figure 2 and Figure 3. Thirdly, in order to analyse the relationship between the lives of residents and preferred party, we download some statistical data in different areas of Victoria about average income from the dataset “Suburb Working Age Employment and Income 2011”, age distribution and percentage of gender from the dataset “SA2 Age Distribution – Females/Males/Persons”, and percentage of people from different countries from the dataset “SA2 Non-English speaking countries of birth”.

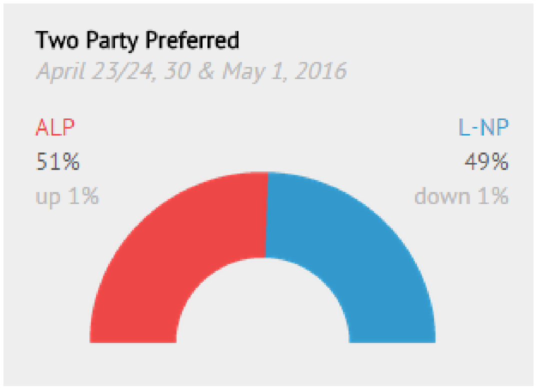


FIGURE 6 TWO PARTIES PREFER RATING (FROM HTTP://WWW.ROYMORGAN.COM/MORGANPOLL)

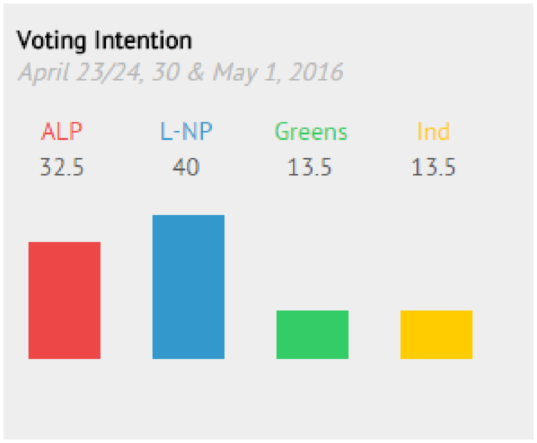


FIGURE 7 VOTING INTENTION (FROM HTTP://WWW.ROYMORGAN.COM/MORGANPOLL)

DYNAMIC DEPOLYMENT

A Boto / Ansible automation script was developed in Python and YAML to deployment the system dynamically. Boto was used to create EC2 instance on NeCTAR Research Cloud and Ansible Playbook was used to deployment applications on the EC2 instance.

Boto

NeCTAR provides extensive OpenStack APIs, allowing programmatic access to each of its services. There are a number of libraries for using this API, and for Python, we chose Boto. Boto provides a Python interface to nearly all of the OpenStack APIs. Boto is mature, well documented and easy to use.

Ansible

Ansible is an automation tool which provides the ways of creating scripts to configure systems and deploy software automatically for simplicity, ease-of-use, security and reliability. Ansible is decentralized and manages machines in an agent-less manner. [4]

Installation for tools

The installation of these two automation tools can be done via "pip", which is the Python package manager, though other options are also available. The "pip" itself can be installed by using,

sudo easy_install pip

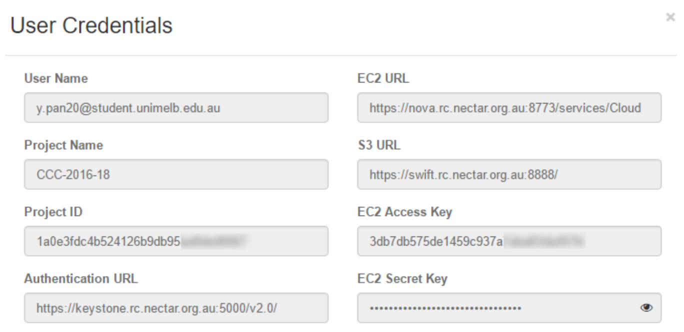
Then, the latest release of Boto and Ansible can be installed by using,

sudo pip boto *sudo pip ansible*

Implementation

CREATE EC2 INSTANCE

To deploy the nodes in Nectar cloud, we used the Boto, which provides an integrated interface for Python to current and future infrastructural services offered by NeCTAR OpenStack. Firstly, we made a connection to the EC2 cloud using our EC2 credentials (access_key and sercret_key).



User Credentials

User Name	EC2 URL
y.pan20@student.unimelb.edu.au	https://nova.rc.nectar.org.au:8773/services/Cloud
Project Name	S3 URL
CCC-2016-18	https://swift.rc.nectar.org.au:8888/
Project ID	EC2 Access Key
1a0e3fdc4b524126b9db95[redacted]	3db7db575de1459c937a[redacted]
Authentication URL	EC2 Secret Key
https://keystone.rc.nectar.org.au:5000/v2.0/	••••••••••••••••••••••••••••••••

SCREENSHOT 5 EC2 USER CREDENTIALS

Then we created a security group which allows incoming traffic on port 80, 22 and 5984 for HTTP, SSH and CouchDB service respectively. After that, we initialized two instances and attach a volume to each of them. By monitoring the status of each instance, we can make sure that all instances have been launched successfully before taking the next step. The final step in this part is to record the IP address of each instance into a file, which works as the inventory for the second part. This list file may looks as follow, with the group name in brackets:

```
[cloudHost]
115.146.94.241
115.146.94.24
```

APPLICATION DEPLOYMENT

Since the two nodes have identical configurations, it is not necessary to differentiate group or role between them. A single script is sufficient. For this part, we will use Ansible to deploy our remote nodes automatically, including install required packages, mount volume and modify configurations. When the deployment of our system is done, the web application and harvesters will be uploaded to remote nodes from local machine for immediate execution.

All of these actions are achieved by a single script called Playbook which is written in YAML and composed a serial publication of module calls (tasks) which are executed on remote machines. A brief description on the goals of each task is listed as follows

- Install all required packages (CouchDB, JAVA, PHP5, Apache2, curl and etc.)
- Mount the volume (/dev/vdb) to the instance
- Modify the configuration file for CouchDB and Apache2
- Upload web application and harvester from local machine and execute them on remote node

The Playbook can be executed by the following command,

```
sudo ansible-playbook -u ubuntu –private-key=feng.pem -I testHost playbook.yml
```

where –u specifies the remote user, **--**private-key specifies the **key** file to authenticate the connection, and –i specifies the inventory host path, all of which can also be determined in the playbook.yml file.

The EC2 instance creation can also be achieved by using Ansible script. Use a single script to deploy numbers of nodes simultaneously can achieve great efficiency and simplicity, especially when a number of nodes need to be deployed.

Scalability of the application and Infrastructure

Within NeCTAR cloud, we can launch new instances, attach volumes to our instances and even request for more resources as long as such authorities are given. All of these actions can be achieved by execute a simple script for the purpose of increasing scalability and extending infrastructure. However, according to our current design of the cloud based solution where the entire solution falls on a single node, it might not meet the desirable scalability. We are also aware of the bottleneck when a high demand of users accessing the application. Ideally, it would be better to have some resource manager nodes. For example, we can build a Hadoop cluster, where the master node manages the source information of data storing on each worker node and workers focus on the computation and analysis of specific data. Since the data are split into blocks which distributed locate in HDFS and using MapReduce to do the computation work, there is no need to assign high configuration on hardware to each node. What’s more, the Resource Manager and Node Manager within Hadoop help us to manage the relationship between nodes as well as allocating jobs and gathering results. Under such condition, scalability can be easily achieved by modifying the configuration file of Hadoop. Although the above are just original concepts when taking project periods into account.

ERROR HANDLING

Duplication Removal

Some Tweets can be collected twice during the data collection. Although they have different database id and revision, these tweets data are essentially the same and should be removed from the database.

In this system, the way to remove duplicate tweets is manipulated after saving the data in CouchDB. The first approach we have taken was to identify and remove the identical tweets before saving them into database in the harvester. With the growing of the database size, it costed huge amount time and resource to iterate the entire database. It decreased the speed of harvesting dramatically.

The approach to remove the duplicate tweets in the database is to a view to list all duplicate Tweets, and then use Java application to delete corresponding data in the database.

COUCHDB VIEW

In database “politician”, if two sets of data have the same Twitter id, only one set of data will be kept and the other will be deleted.

However, in database “followers”, Tweets cannot be removed only by the same Twitter id. As mentioned in the Harvester section, during the data pre-process, the Tweets from the Member’ followers will be tagged with the name of the Party (doc.fParty) and the Member (doc.fName) for fast view processing as some people may follow different Members. In database “followers”, the Tweets would only be identified as duplicate and removed if they have the same Twitter id and fName.

There is a hash map created in this view, if one set of data has the same Twitter id (in “politician”) or the same Twitter id and fName (in “followers”), then emit the id and revision of this data.

JAVA APPLICATION

After having a view, we use a Java program to remove duplicate. In Java program, the Lightcouch API is chosen to do the connection and operation with CouchDB. Firstly, we use the function *CouchDbClient.view()* to get the view result from CouchDB, and then transform to a *List* which contains *JSONObject.class* by *query(JSONObject.class)* that is a sub function of *view()*. As we know, the CouchDB use _id and _rev to identify the data, and the removing data function of Lightcouch is *CouchDbClient.remove(id, rev)*. In this case, we read the id and rev from the view, and use this two values to remove relevance data in CouchDB by algorithm 3.

```
Algorithm 3: Removing data
  Input: The ip address of the CouchDB;
  Output: null;
1     Loading the result of view from CouchDB in List<JSONObject>(i);
2     foreach JSONObject List(i) do
3         initialize id = List(i).get("key");
4         initialize rev = List(i).get("value");
5         CouchDbClient.remove(id, rev);
```

Fault Tolerance

The system has the following layers of fault tolerance.

MULTI HARVESTERS

During the data collection phase, there were 16 harvesters working on four virtual machines, utilizing both Twitter REST API and Twitter Streaming API to collect the data. In the event of a single node or harvester outage, the rest of the nodes or harvesters will keep collecting the data.

BROKEN-POINT RESUMING

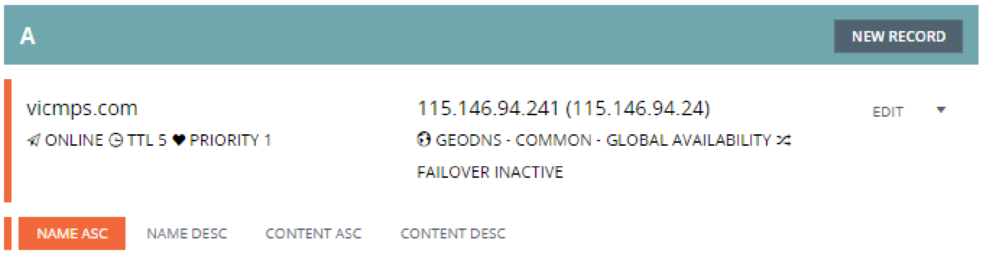
Harvesters often stopped due to the temporary outage of the Twitter API (error code 130) during the data collection. We break the follower's list into small pages (the smaller the better, however, due to the API Rate Limits, we found the 100 followers per page is the smallest we can go.), then use cursor to navigate on the page and between pages. The current cursor is printed out for resuming purpose. In the event a harvester stopped working, we can easily found the broken-point and restart the harvester to collecting the data from the broken-point to minimize the duplication.

DATABASE REPLICATION

The databases on the main server are replicated to the backup server for backup and failover purpose.

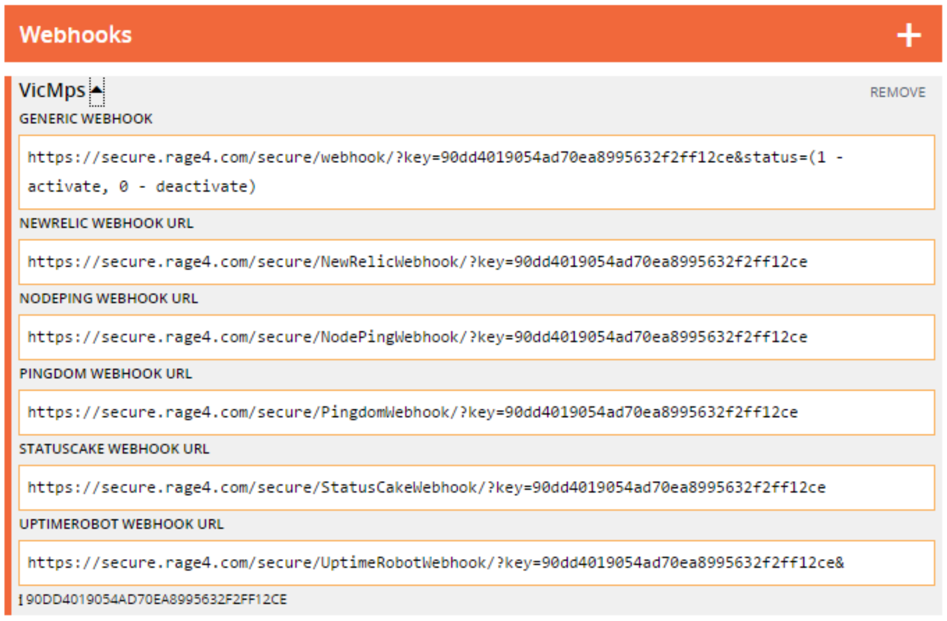
Active Failover

The domain name vicmps.com is hosted on Rage4 DNS Manager. For high availability purpose, DNS level active failover is enabled for A record.



SCREENSHOT 6 RAGE4 DNS RECORD

Rage4 active failover uses UptimeRobot, ScaleXtreme, NewRelic, NodePing and StatusCake as monitoring service to detect downtime and adjust the DNS configuration for failover purpose. These monitors are enabled in Webhooks.



Webhooks

VicMps

REMOVE

GENERIC WEBHOOK

```
https://secure.rage4.com/secure/webhook/?key=90dd4019054ad70ea8995632f2ff12ce&status=(1 - activate, 0 - deactivate)
```

NEWRELIC WEBHOOK URL

```
https://secure.rage4.com/secure/NewRelicWebhook/?key=90dd4019054ad70ea8995632f2ff12ce
```

NODEPING WEBHOOK URL

```
https://secure.rage4.com/secure/NodePingWebhook/?key=90dd4019054ad70ea8995632f2ff12ce
```

PINGDOM WEBHOOK URL

```
https://secure.rage4.com/secure/PingdomWebhook/?key=90dd4019054ad70ea8995632f2ff12ce
```

STATUSCAKE WEBHOOK URL

```
https://secure.rage4.com/secure/StatusCakeWebhook/?key=90dd4019054ad70ea8995632f2ff12ce
```

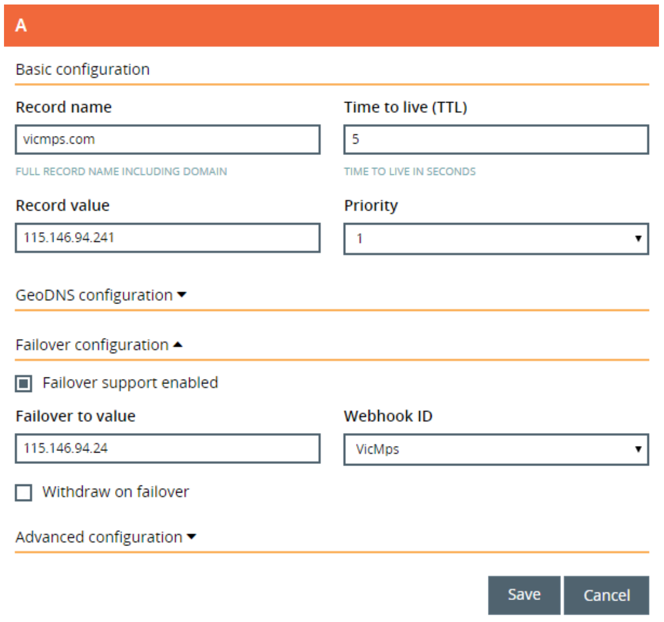
UPTIMEROBOT WEBHOOK URL

```
https://secure.rage4.com/secure/UptimeRobotWebhook/?key=90dd4019054ad70ea8995632f2ff12ce&
```

90DD4019054AD70EA8995632F2FF12CE

SCREENSHOT 7 RAGE4 WEBHOOK

Rage4 failover supports A, AAAA and CNAME records. For our system, failover is enabled on A record.



A

Basic configuration

Record name: vicmps.com
FULL RECORD NAME INCLUDING DOMAIN

Time to live (TTL): 5
TIME TO LIVE IN SECONDS

Record value: 115.146.94.241

Priority: 1

GeoDNS configuration

Failover configuration

[✓] Failover support enabled

Failover to value: 115.146.94.24

Webhook ID: VicMps

[ ] Withdraw on failover

Advanced configuration

SCREENSHOT 8 FAILOVER SETTINGS

WEB APPLICATION

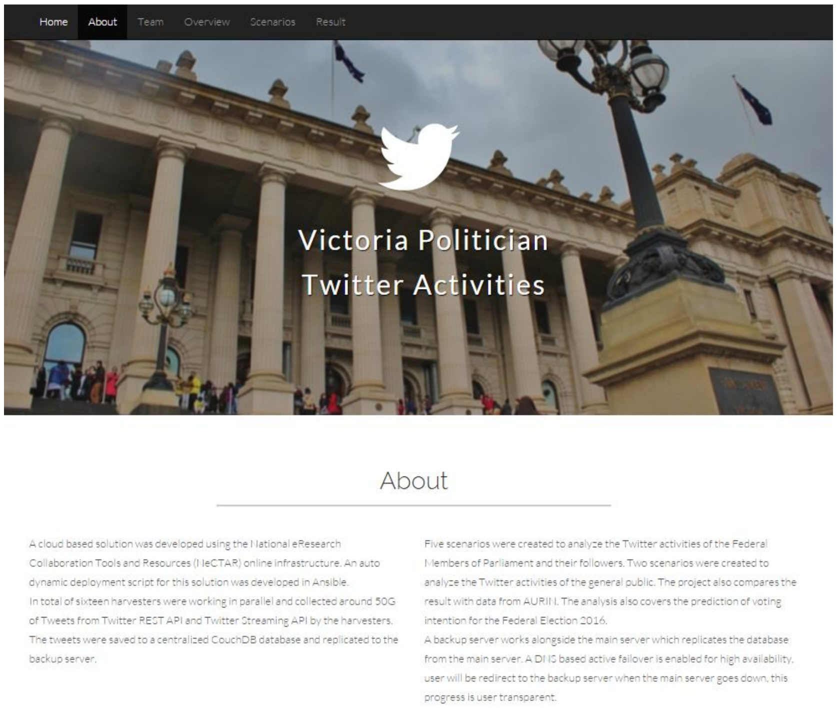
Overview

The web application was developed in PHP and JAVAScript, it connects to the CouchDB server and demonstrates the scenarios created in this project. The following libraries are used in the web application,

Library	Purpose
bootstrap	Generate HTML theme, animation and styling based on bootstrap CSS and JS
PHP-On-Couch	Map all actions the website can do on the CouchDB server, responsible for low level dialog between PHP and CouchDB and manipulation of document to allow uses PHP getters and setters.
Google Chart API	To visualize data of Twitter result on the website. Implemented via JavaScript and external Google APIs
Google Maps API	To visualize Twitter data via the map on the website. Reverse Geo-coding
Google Fonts	Optimize website font style
CanvasJS	Generate dynamic chart

TABLE 6 LIBRARIES USED

The web application can be accessed from http://www.vicmps.com.



About

A cloud based solution was developed using the National eResearch Collaboration Tools and Resources (NeCTAR) online infrastructure. An auto dynamic deployment script for this solution was developed in Ansible.
In total of sixteen harvesters were working in parallel and collected around 50G of Tweets from Twitter REST API and Twitter Streaming API by the harvesters. The tweets were saved to a centralized CouchDB database and replicated to the backup server.

Five scenarios were created to analyze the Twitter activities of the Federal Members of Parliament and their followers. Two scenarios were created to analyze the Twitter activities of the general public. The project also compares the result with data from AURIN. The analysis also covers the prediction of voting intention for the Federal Election 2016.
A backup server works alongside the main server which replicates the database from the main server. A DNS based active failover is enabled for high availability, user will be redirect to the backup server when the main server goes down, this progress is user transparent.

SCREENSHOT 9 WEB APPLICATION LADNING PAGE

Scenario one – Top 10 User Mentions

DATA SOURCE

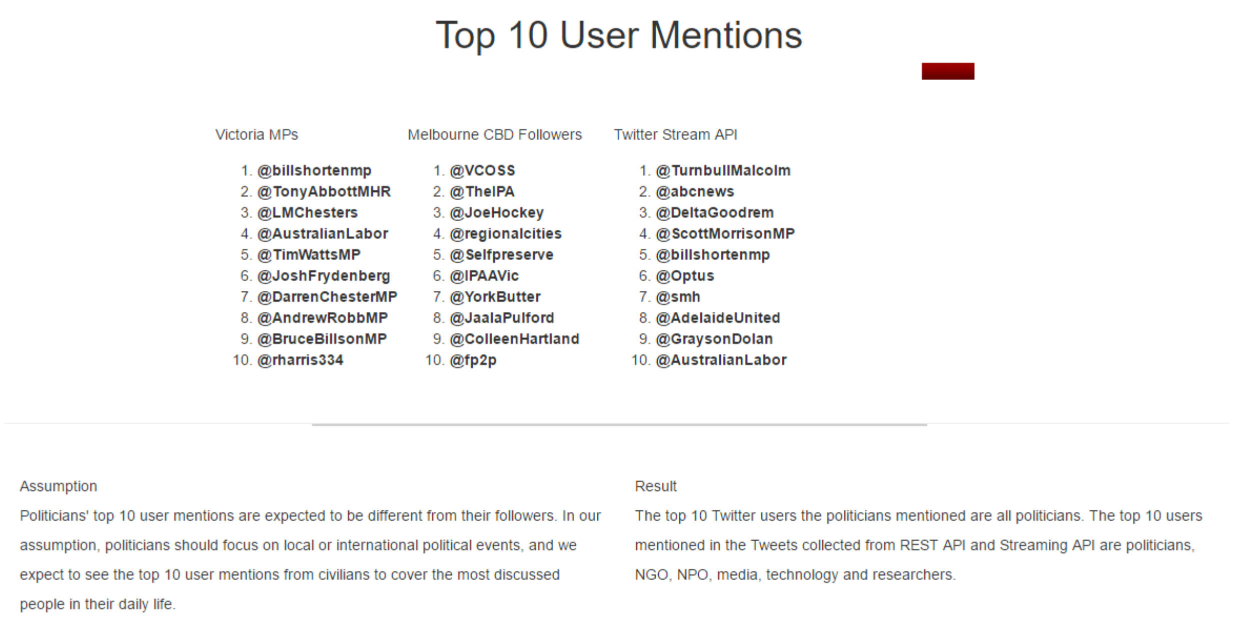
Database	View
politician	count/atsign
followers	count/atsign_melbourne
streaming	count/atsign

ASSUMPTION

Politicians' top 10 user mentions are expected to be different from their followers. In our assumption, politicians should focus on local or international political events, and we expect to see the top 10 user mentions from civilians to cover the most discussed people in their daily life.

RESULT

The top 10 Twitter users the politicians mentioned are all politicians. The top 10 users mentioned in the Tweets collected from REST API and Streaming API are politicians, NGO, NPO, media, technology and researchers



Top 10 User Mentions

Victoria MPs	Melbourne CBD Followers	Twitter Stream API
1. @billshortenmp	1. @VCOSS	1. @TurnbullMalcolm
2. @TonyAbbottMHR	2. @TheIPA	2. @abcnews
3. @LMChesters	3. @JoeHockey	3. @DeltaGoodrem
4. @AustralianLabor	4. @regionalcities	4. @ScottMorrisonMP
5. @TimWattsMP	5. @Selfpreserve	5. @billshortenmp
6. @JoshFrydenberg	6. @IPAAVic	6. @Optus
7. @DarrenChesterMP	7. @YorkButter	7. @smh
8. @AndrewRobbMP	8. @JaalaPulford	8. @AdelaideUnited
9. @BruceBillsonMP	9. @ColleenHartland	9. @GraysonDolan
10. @rharris334	10. @fp2p	10. @AustralianLabor

Assumption

Politicians' top 10 user mentions are expected to be different from their followers. In our assumption, politicians should focus on local or international political events, and we expect to see the top 10 user mentions from civilians to cover the most discussed people in their daily life.

Result

The top 10 Twitter users the politicians mentioned are all politicians. The top 10 users mentioned in the Tweets collected from REST API and Streaming API are politicians, NGO, NPO, media, technology and researchers.

SCREENSHOT 10 TOP 10 USER MENTIONS

Scenario two – Top 10 Hashtags

DATA SOURCE

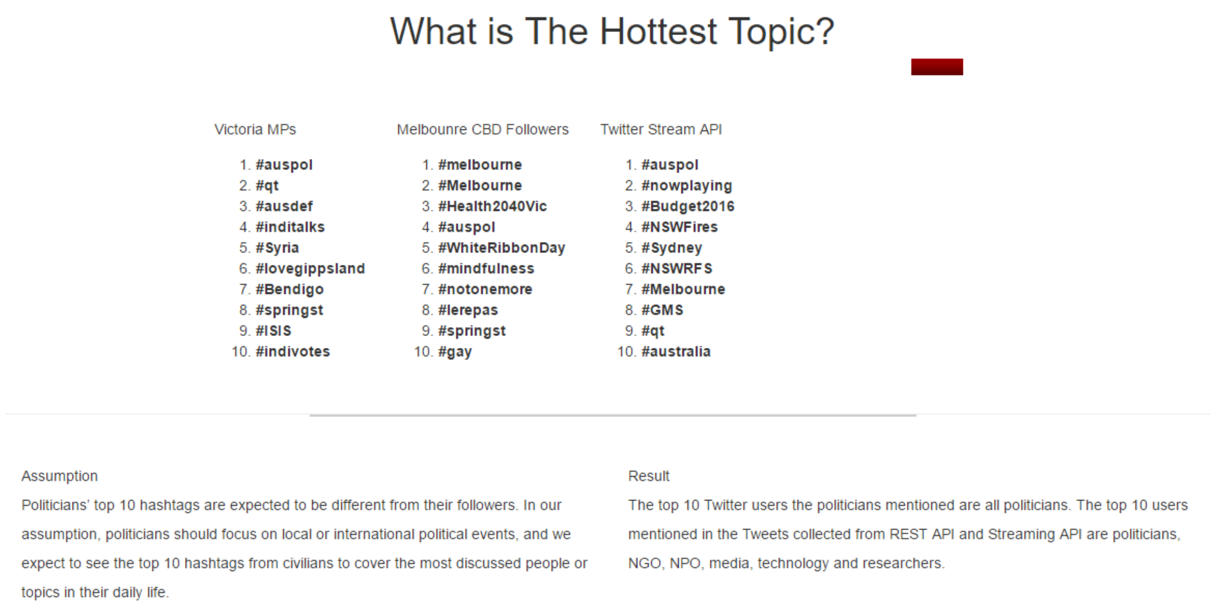
Database	View
politician	count/hashtag
followers	count/hashtag_melbourne
streaming	count/hashtag

ASSUMPTION

Politicians' top 10 hashtags are expected to be different from their followers. In our assumption, politicians should focus on local or international political events, and we expect to see the top 10 hashtags from civilians to cover the most discussed people or topics in their daily life.

RESULT

The top 10 Twitter users the politicians mentioned are all politicians. The top 10 users mentioned in the Tweets collected from REST API and Streaming API are politicians, NGO, NPO, media, technology and researchers.



What is The Hottest Topic?

Victoria MPs	Melbounre CBD Followers	Twitter Stream API
1. #auspol	1. #melbourne	1. #auspol
2. #qt	2. #Melbourne	2. #nowplaying
3. #ausdef	3. #Health2040Vic	3. #Budget2016
4. #inditalks	4. #auspol	4. #NSWFires
5. #Syria	5. #WhiteRibbonDay	5. #Sydney
6. #lovegippsland	6. #mindfulness	6. #NSWRFS
7. #Bendigo	7. #notonemore	7. #Melbourne
8. #springst	8. #lerepas	8. #GMS
9. #ISIS	9. #springst	9. #qt
10. #indivotes	10. #gay	10. #australia

Assumption

Politicians' top 10 hashtags are expected to be different from their followers. In our assumption, politicians should focus on local or international political events, and we expect to see the top 10 hashtags from civilians to cover the most discussed people or topics in their daily life.

Result

The top 10 Twitter users the politicians mentioned are all politicians. The top 10 users mentioned in the Tweets collected from REST API and Streaming API are politicians, NGO, NPO, media, technology and researchers.

SCREENSHOT 11 TOP 10 HASHTAGS

Scenario three – Most Frequently Used Words

DATA SOURCE

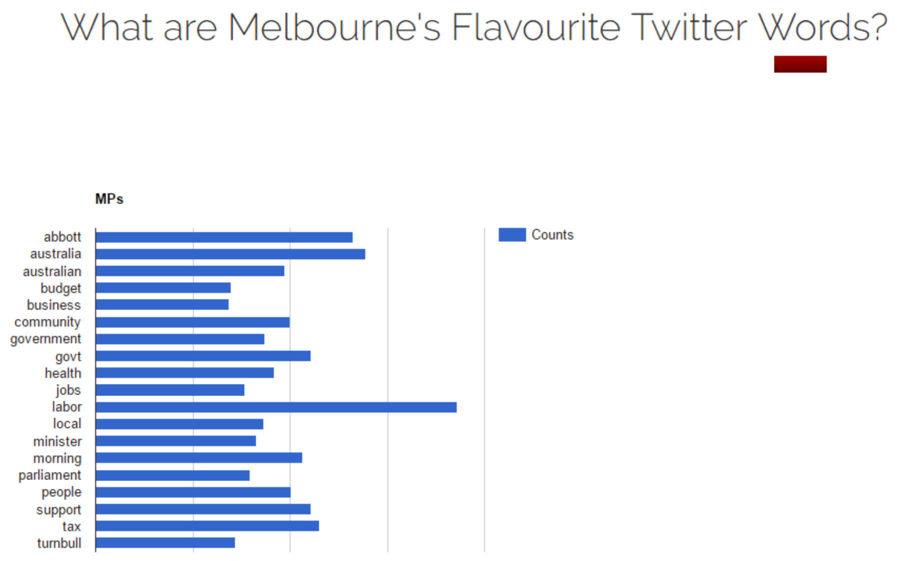
Database	View
politician	count/words
followers	count/word_melbourne
streaming	count/word

ASSUMPTION

We expect the Tweets the politicians have tweeted to be well worded and refined. We don't expect any swearing language or extreme wording to be appeared often. However, the most frequently used words are expected to be close to daily language.

RESULT

As expected, the most frequently used the words in politicians Tweets are political words or topic. The most frequently used words in civilians are more generic and random.



SCREENSHOT 12 TOP 10 WORDS FROM MPS

Scenario four – Emotion Map

DATA SOURCE

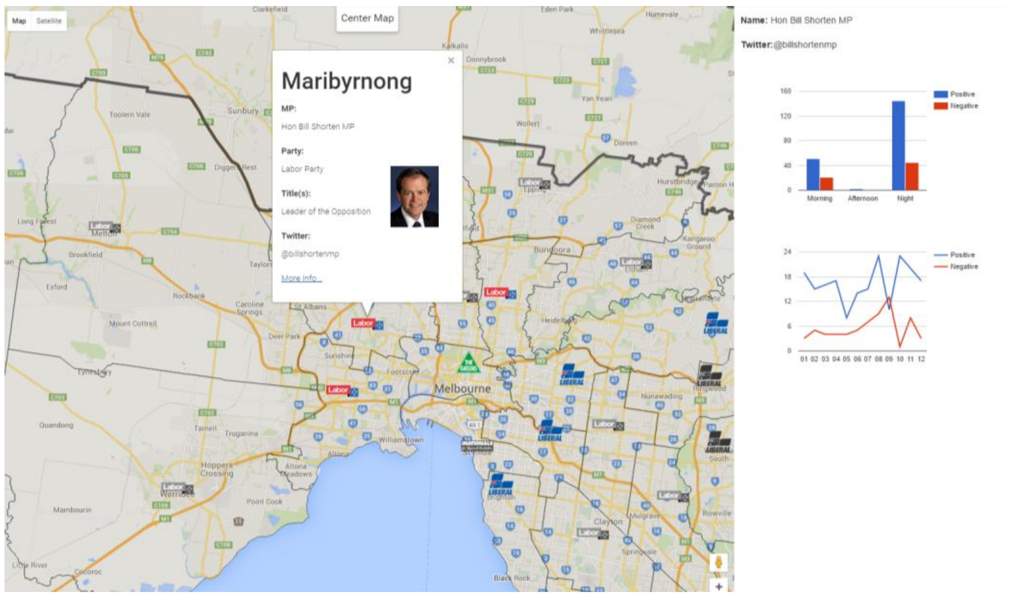
Database	View
politician	count/emotion_mp_am
politician	count/emotion_mp_month

ASSUMPTION

We expect to see the Tweets from the politicians to be positive at most of times.

RESULT

As expected, most of the Tweets from the politicians are positive. And in general, the MPs are happier in the night time.



Maribyrnong
MP: Hon Bill Shorten MP
Party: Labor Party
Title(s): Leader of the Opposition
Twitter: @billshortenmp
More info...

Name: Hon Bill Shorten MP
Twitter: @billshortenmp

SCREENSHOT 13 EMOTION MAPS

Scenario five – Location Heat Map

DATA SOURCE

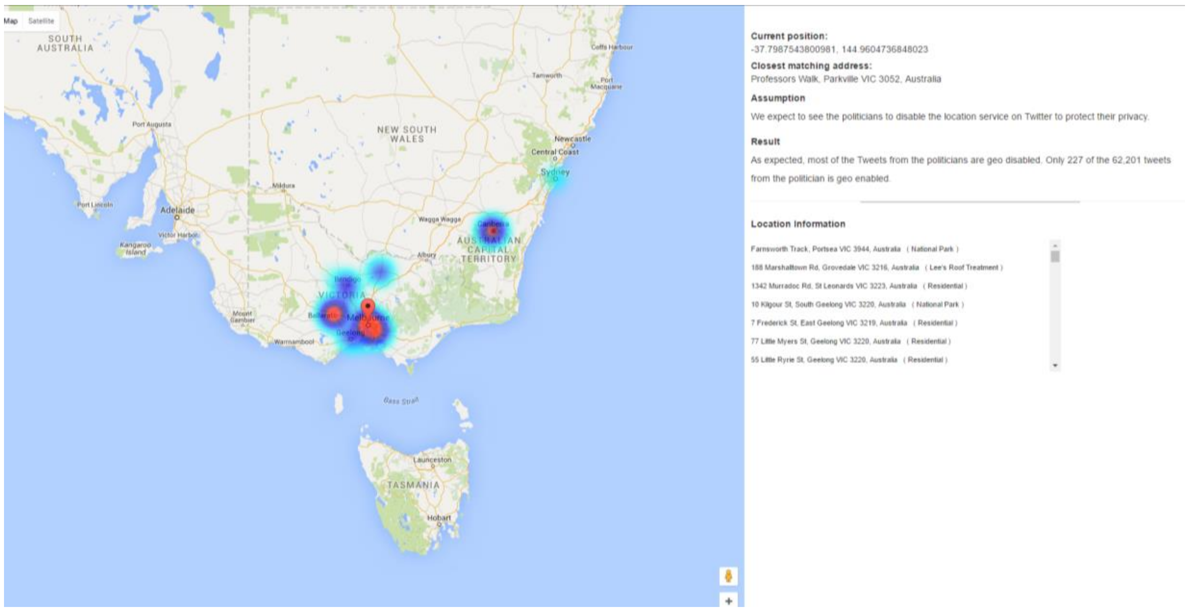
Database	View
politician	location/geoLocation

ASSUMPTION

We expect to see the politicians to disable the location service on Twitter to protect their privacy.

RESULT

As expected, most of the Tweets from the politicians are geo disabled. Only 227 of the 62,214 tweets from the politician is geo enabled.



Current position:
-37.7987543800981, 144.9604736848023

Closest matching address:
Professors Walk, Parkville VIC 3052, Australia

Assumption

We expect to see the politicians to disable the location service on Twitter to protect their privacy.

Result

As expected, most of the Tweets from the politicians are geo disabled. Only 227 of the 62,201 tweets from the politician is geo enabled.

Location Information

Farnsworth Track, Portsea VIC 3944, Australia (National Park)
188 Marshalltown Rd, Grovedale VIC 3216, Australia (Lee's Roof Treatment)
1342 Murradoc Rd, St Leonards VIC 3223, Australia (Residential)
10 Kilgour St, South Geelong VIC 3220, Australia (National Park)
7 Frederick St, East Geelong VIC 3219, Australia (Residential)
77 Little Myers St, Geelong VIC 3220, Australia (Residential)
55 Little Ryrie St, Geelong VIC 3220, Australia (Residential)

SCREENSHOT 14 HEAT MAP

Scenario six – Voting Intention

DATA SOURCE

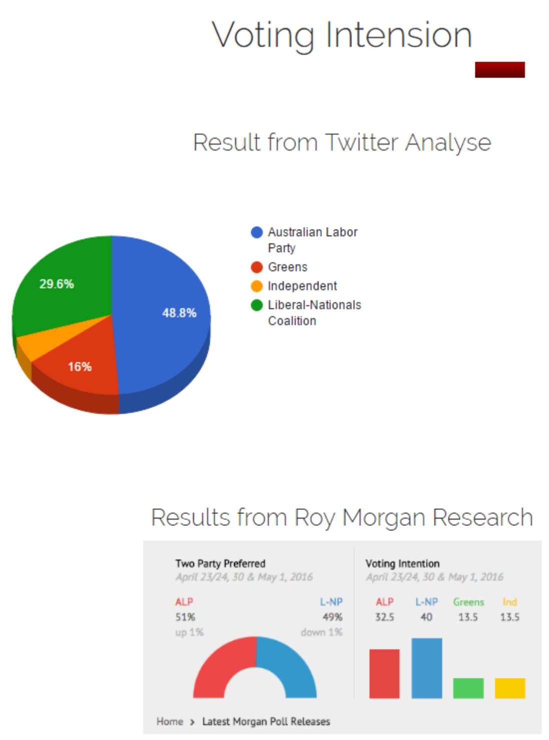
Database	View
followers	count/voting_intention
poll data (Roy Mogan / Galaxy)	

ASSUMPTION

We expect to see the voting intention we generated by analysing the Twitter activities should match the polls done by third party.

RESULT

According to our analysis, the support rate for L-NP in Victoria is lower than the support poll done by Roy Morgan Research and Galaxy Research. While the support rate for the Greens is higher. We believe the reason is the Green party has a significantly higher followers base.



SCREENSHOT 15 VOTING INTENTIONS

Scenario seven - Sentiment Analyse

DATA SOURCE

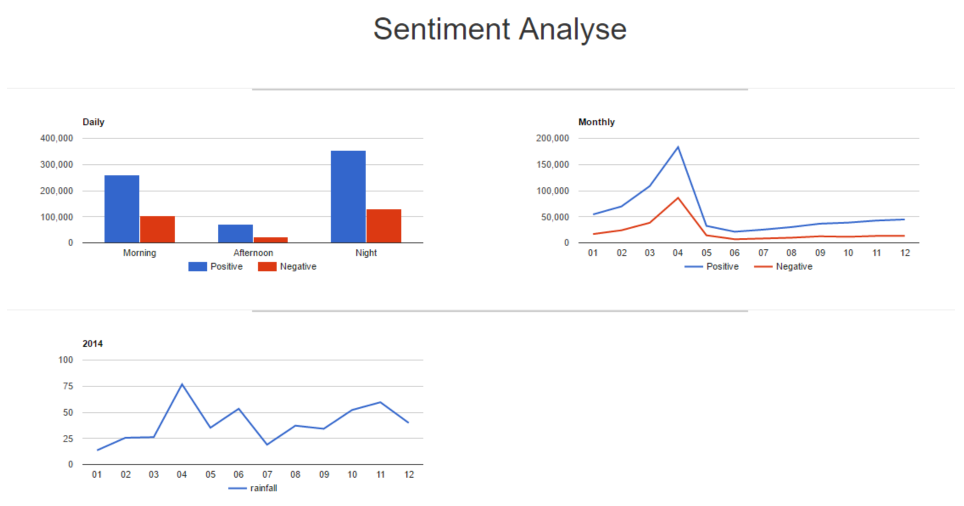
Database	View
politician	count/emotion
Mean rainfall (BoM)	

ASSUMPTION

From the civilians, we expect to see the positive Tweets are much higher than the negative ones. We also see more negative Tweets on raining days.

RESULT

As we expected, the number of positive Tweets are always double the number of the negative ones. Compare with the Mean Rainfall data from BoM, we can see high negative Tweets if the rainfall in the month is higher. In general, people are happier during the night time.



SCREENSHOT 16 SENTIMENT ANALYSE

Scenario eight – Privacy

DATA SOURCE

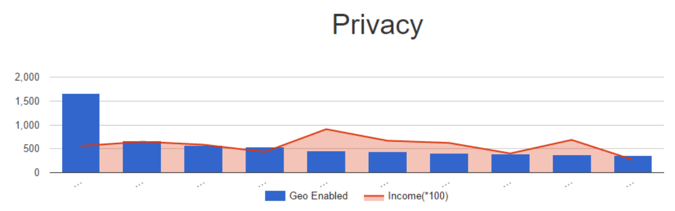
Database	View
followers	location/byCity
income_area	income/income_top10Geo

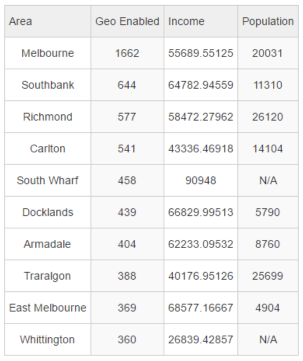
ASSUMPTION

We expect to see less geo enabled Tweets from high-income area than the low-income area, as we believe the high-income people pay more attention to privacy.

RESULT

We compared the result from the Twitter data with the data from Aurin. The result shows the number of geo enabled Tweets in an area is income irrelevant.





Area	Geo Enabled	Income	Population
Melbourne	1662	55689.55125	20031
Southbank	644	64782.94559	11310
Richmond	577	58472.27962	26120
Carlton	541	43336.46918	14104
South Wharf	458	90948	N/A
Docklands	439	66829.99513	5790
Armadale	404	62233.09532	8760
Traralgon	388	40176.95126	25699
East Melbourne	369	68577.16667	4904
Whittington	360	26839.42857	N/A

SCREENSHOT 17 PRIVACY

Scenario nine – Language

DATA SOURCE

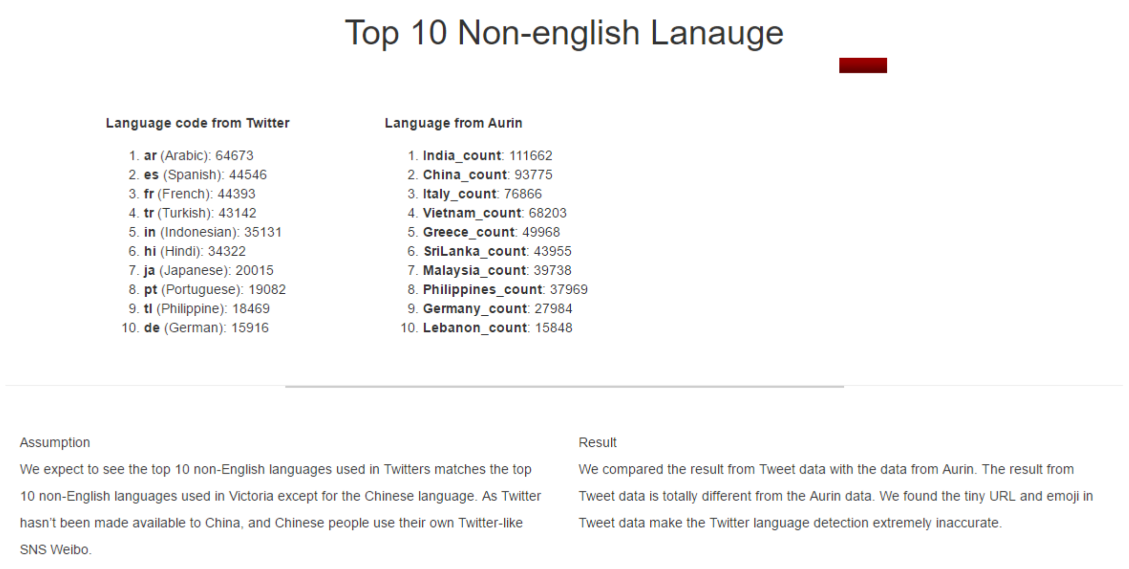
Database	View
followers	language/byCity
non_english_area	language/countAll
non_english_area	population/totalpopulationwithGeo

ASSUMPTION

We expect to see the top 10 non-English languages used in Twitters matches the top 10 non-English languages used in Victoria except for the Chinese language. As Twitter hasn't been made available to China, and Chinese people use their own Twitter-like SNS Weibo.

RESULT

We compared the result from Tweet data with the data from Aurin. The result from Tweet data is totally different from the Aurin data. We found the tiny URL and emoji in Tweet data make the Twitter language detection extremely inaccurate.



Top 10 Non-english Lanauge

Language code from Twitter

1. **ar** (Arabic): 64673
2. **es** (Spanish): 44546
3. **fr** (French): 44393
4. **tr** (Turkish): 43142
5. **in** (Indonesian): 35131
6. **hi** (Hindi): 34322
7. **ja** (Japanese): 20015
8. **pt** (Portuguese): 19082
9. **tl** (Philippine): 18469
10. **de** (German): 15916

Language from Aurin

1. **India_count**: 111662
2. **China_count**: 93775
3. **Italy_count**: 76866
4. **Vietnam_count**: 68203
5. **Greece_count**: 49968
6. **SriLanka_count**: 43955
7. **Malaysia_count**: 39738
8. **Philippines_count**: 37969
9. **Germany_count**: 27984
10. **Lebanon_count**: 15848

Assumption

We expect to see the top 10 non-English languages used in Twitters matches the top 10 non-English languages used in Victoria except for the Chinese language. As Twitter hasn't been made available to China, and Chinese people use their own Twitter-like SNS Weibo.

Result

We compared the result from Tweet data with the data from Aurin. The result from Tweet data is totally different from the Aurin data. We found the tiny URL and emoji in Tweet data make the Twitter language detection extremely inaccurate.

SCREENSHOT 18 LANGUAGE

Technical Aspects

PASS DATA FROM PHP TO JAVASCRIPT

The result above is retrieved directly from CouchDB and PHP On Couch library by dready92 have been implemented. Data will be sorted in the PHP file using usort() method and the time assumed are showing below:

- 3 – 4 seconds to sort 719 rows from database.
- 15 – 17 seconds to sort 68,849 rows from database
- 30 seconds or timeout to sort 100,000 rows from database

By echoing the data from atsign.php, the front end atsign.html will display the sorted real-time Twitter data and use JavaScript to get the information from the DOM.

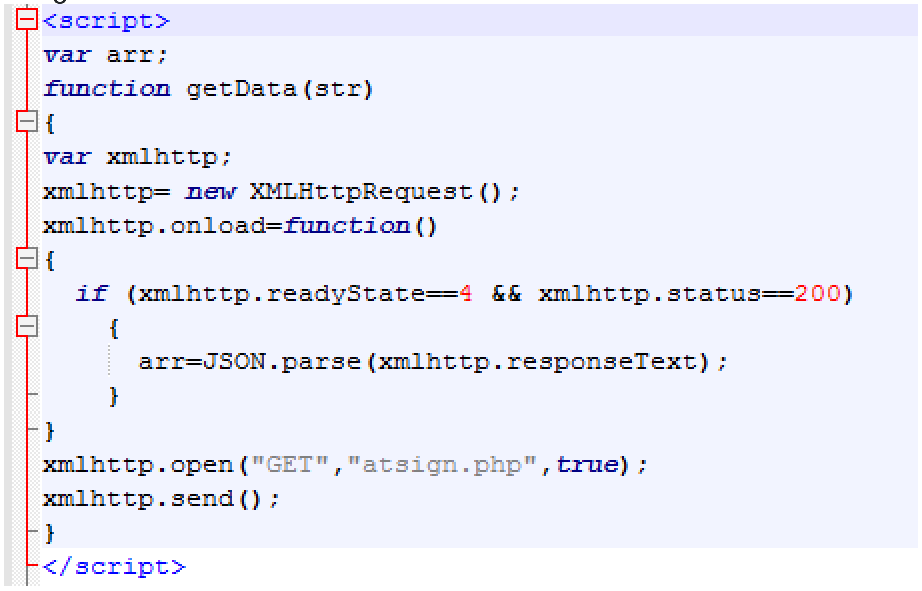
PROS

- Fast – DOM operations are quick. No extra latency as AJAX, which needs to create an HTTP request in the first stage.

CONS

- Harder separation between layers – If the system moves to use REST API, servlet or other service, JAVAScript and HTML would need to be rewritten.
- Mess-up the code – JavaScript, HTML and PHP are all mixed up together in one file. It would increase the difficulty for future expansions and harder for maintenance.

AJAX is not used in this project, because of its cons of extra latency thermionically. Below is a sample code of implementing AJAX



```
<script>
var arr;
function getData(str)
{
var xmlhttp;
xmlhttp= new XMLHttpRequest();
xmlhttp.onload=function()
{
  if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
      arr=JSON.parse(xmlhttp.responseText);
    }
}
xmlhttp.open("GET","atsign.php",true);
xmlhttp.send();
}
</script>
```

SCREENSHOT 19 AJAX CODE

NeCTAR Pros and Cons

NeCTAR is the acronym of National eResearch Collaboration Tools and Resources [5], which provides researchers in Australian and around the world with powerful information and technology infrastructure (ICT) that helps them to promote the research collaboration and research outcomes. In this project, our cloud based system is hosted in the NeCTAR Research Cloud which is an OpenStack based computational cloud. The Research Cloud provides its user up to 8 high-performing nodes that can be used to host research applications. As one of the biggest OpenStack based clouds, it benefits from the strengths of OpenStack in following aspects.

- **Dashboard** - Dashboard provides users with an interactive interface to manage the computation, storage and networking resources, allowing users or administrators to have an overview of the current resources usage and current VM instances. All operations can be easily done in the interface rather than editing complex configuration scripts.
- **Security** - NeCTAR allows users to customize different security groups to access the created instances. It supports SSH communication with reliable RSA public-key scheme for system management and applications uploading and HTTP communication for web services. For authentication, it provides robust access control. The access to the instance can be controlled in user, role and project levels and it can be done via username/password based authentication (OpenStack) or token based authentication (EC2).
- **Storage** - NeCTAR gives users freedom to customize the storage of each instance. When an instance is created, the cloud will automatically allocate an ephemeral storage (10GB) to it. User could also allocate persistent volumes to different nodes by need. Volumes are independent from instances, they can exist even if you deleted the instance that they have been attached to, which guarantee the safety and reliability of data storage.
- **APIs** - A range of graphic, command line, application interfaces are available in NeCTAR Research Cloud. User can use these APIs to better support the research applications.

Despite the immense benefits in Cloud computing provided by the Nectar Research cloud, some of the challenges/disadvantages applicable to the project were:

- The lack of elasticity available for sudden load increases. Due to the manual setup of the nodes, the Nectar infrastructure does not respond effectively to a sudden rise in the access to the system unlike EC2[6] provided by Amazon. This was not a direct issue encountered in the project, however it is problematic for other works with a combination of a high reliability requirement as well as high network demand.
- Does not provide Microsoft Windows as an operating system for the nodes. Although it did not affect this project in a significant way, other works requiring specialized Windows software will be affected by the licensing restrictions on Nectar.
- Reliability issues were encountered throughout the harvesting process, with the nodes going down multiple times due to system issues on Nectar. This affected the schedule of the harvesting process and the project as a whole.
- Performance issue were also encountered during the data processing phase. The speed of view generating was slow due to the low disk I/O.

NeCTAR vs AWS

NeCTAR aims to enhance research collaboration and research outcomes by providing a "private cloud" service operated by participating universities to Australian researchers. [7]

Amazon Web Services (AWS), is a collection of cloud computing services that make up the on-demand computing platform offered by Amazon, providing services range from compute, storage, networking, database, analytics, application services, deployment, management and mobile. [8]

Key differences between the NeCTAR and AWS:

- **Access:** NeCTAR is a private cloud service, provides free service to Australian Access Federation users based on a project quota. AWS is a public cloud service, provides a wide range of cloud services based on a range of pricing models.
- **Compute:** NeCTAR supports up to 16 vCPU per virtual machine, limited flavours are available and supports Linux based instances only. AWS supports up to 32 vCPU per virtual machine, a range of flavours are available and a wide range of OS are supported.
- **Storage and Database:** NeCTAR supports Volume storage, Object storage and RDSI storage. Users need to install database themselves. AWS supports a wide range of storage services and database solutions.
- **Networking:** A public IP address is provided to each instance on NeCTAR, any other services need to be installed by the users themselves. AWS offers a wide range of networking services including VPC, VPN and load balancing.
- **Security:** Both NeCTAR and AWS use a shared responsibility model, i.e. the provider only responsible for physical security and virtualization infrastructure security, the users are responsible for the security for applications and OS.
- **Other:** NeCTAR offers HEAT (in beta) and Ceilometer API (in beta). AWS supports a much wider range of services such as CDN, SNS, mobile service, FS, RDS and much more.

NeCTAR is a government funded Super Science project which provides the research in Australia and around the world with private eResearch cloud with limited resources and services. It may be less user friendly, not flexible and lack of ease of use than most of the commercial public cloud like AWS or Azure, but it is sufficient for research purpose and the service is free for all AFF users.

When a research project demands higher resources, more service and improved availability and reliability. Or a commercial project which requires out of box services and improved availability and reliability, AWS may become a better choice.

USER MANUAL

CREATE EC2 INSTANCE

The following command can be used to create EC2 instance,

python finalBOTO.py

DEPLOY APPLICATIONS

The Playbook can be executed by the following command,

sudo ansible-playbook -u ubuntu –private-key=feng.pem -I testHost playbook.yml

where **–**u specifies the remote user, **--**private-key specifies the key file to authenticate the connection, and –i specifies the inventory host path, all of which can also be determined in the playbook.yml file.

START HARVESTER

The following command can be used to start the REST API Harvester VicMPs.jar,

java -jar VicMPs.jar -d politician -h 115.146.94.241 -t 408200652 -a 1 -p labor

where -d specifies the database name, -h specifies the server IP address, -t specifies the Twitter ID, -a specifies the authentication credential sequence number, -p specifies the name of the party.

The following command can be used to start the REST API Harvester VicMPFollowers.jar,

java -jar VicMPFollowers.jar -d followers -h 115.146.94.241 -a 9 -t 252331819 -c -1 -p liberal

where -d specifies the database name, -h specifies the server IP address, -t specifies the Twitter ID, -a specifies the authentication credential sequence number, -c specifies the cursor, -p specifies the name of the party.

The following command can be used to start the Streaming API Harvester Streaming.jar,

java -jar Streaming.jar -d followers -h 115.146.94.241

where -d specifies the database name, -h specifies the server IP address.

GITHUB LINK:

HTTPS://GITHUB.COM/GEMINEAN/COMP90024

YOUTUBE LINK:

Ansible: https://youtu.be/6nxc_yrdW4E

Web application: https://youtu.be/rEYCmZZ1EWU

REFERENCE

[1] Wikipedia. Twitter - Wikipedia, the free encyclopedia. [ONLINE] Available at: https://en.wikipedia.org/wiki/Twitter. [Accessed 07 May 2016].

[2] Twitter Developers. Twitter Developers. [ONLINE] Available at: https://dev.twitter.com/. [Accessed 07 May 2016].

[3] Wikipedia. 2016. CouchDB - Wikipedia, the free encyclopedia. [ONLINE] Available at: https://en.wikipedia.org/wiki/CouchDB. [Accessed 10 May 2016].

[4] Ansible, Inc.. 2016. Ansible Documentation. [ONLINE] Available at: http://docs.ansible.com/. [Accessed 10 May 2016].

[5] Nectar. 2016. About - Nectar. [ONLINE] Available at: https://nectar.org.au/about/. [Accessed 10 May 2016].

[6] Amazon Web Services, Inc.. 2016. Elastic Compute Cloud (EC2) Cloud Server & Hosting – AWS. [ONLINE] Available at: http://aws.amazon.com/ec2/. [Accessed 10 May 2016].

[7] Differences to NeCTAR — AWS On Boarding Guide. [ONLINE] Available at: https://espaces.edu.au/aws/differences-to-nectar. [Accessed 10 May 2016].

[8] Wikipedia. 2016. Amazon Web Services - Wikipedia, the free encyclopedia. [ONLINE] Available at: https://en.wikipedia.org/wiki/Amazon_Web_Services. [Accessed 10 May 2016].