Social Mining Harvester, using Twitter

Alberto F. Jové Iguina
Master in Computer Science
Jeffrey Duffany, Ph.D.
Electrical & Computer Engineering and Computer Science Department
Polytechnic University of Puerto Rico

Abstract — *This article showcases phase one of a system that is able to analyze the Social Web. Here we present how to create a harvester for the Social Network Twitter. We explain the different technologies used and how to proceed to improve this first phase, as well as to provide ideas for future phases. Social Networks are the biggest buzzword on the Internet, and a tool like this harvester will provide us a method to easily acquire data. In order to execute some data mining techniques that will help understand what is going on to companies, governments and people. The wealth of data readily available is of interest to many, and we have been able to create a platform where to get or view the data from Twitter.*

Key Terms — *CouchDB, Python, Redis, Tweets.*

WHAT IS TWITTER?

Social Networking sites like Facebook, MySpace, Google Plus, LinkedIn and Twitter have been gaining more and more attention since 2010. This concept of Social Networks started close to 2005, with each passing year since then these sites have created and cemented their foothold in the Internet. Currently it goes as far as to having TV viewers interact with their services while watching TV.

The Social Network Twitter has been selected as the network to acquire data because of the current popularity it has gained and keeps gaining. At one hundred and forty characters [1] it is considered a micro blogging service; Twitter is an option inside the Internet called Social Networks. Currently it seems to be the latest Social Network where everyone is flocking into, looking at what is the latest trend on the Internet and participate.

It is seen as a place where revolutions are taking place [2]:

- Whether it is the entertainment industry promoting their latest work
- Or an organization campaigning for their preferred charity
- Athletes being criticized or praised for their performances
- Politicians campaigning for their political positions
- The Pope and other religious figure blessing their followers
- Citizens believe they can be the new News Reporters
- Citizens have united and mobilize in protests against their governments

There are so many possible uses for the information inside twitter, that it is hard not to invest time, money and effort into analyzing all of the available content within the systems. The Social Networks is an ecosystem open for analysis with a wealth of crude data just waiting to be stored. All of the major Social Web sites: Facebook, LinkedIn, Google Plus, and any website can be accessed and stored for further analysis. It is important to mention that while the data is readily available some rules need to be followed. [3] – [4]

OPPORTUNITIES FOR THE DATA

A company that is researching opinions from the general public could develop and deploy such a system that will gather the information that is of interest about its previous or current products and services. An analyst can create a system that will provide him all of the necessary data he needs in order to provide feedback on the topics of interest. Could we see Twitter data, or tweets as they are referred as, like a historical document?

Historical documentation has always been created using new methods of information sharing:

stories, books, audio recordings, video recordings, magazines, and digital content to name a few. In the past we have been able to go to the library and search all of these types of media distribution, today we can do the same by searching the Internet. Tomorrow all of the information that is being generated by Social Networks will be of interest as newspaper has been for sometime. [5]

Like newspapers a media that expresses a thought, a way of thinking, so do tweets. These tweets can reflect what a person is feeling because of an experience they have just lived, these tweets can also influence they way others feel about the message and create a heated discussion between different users. In essence the tweets will provide a state of mind of all the captured data. The Library of Congress of the USA started a project in 2010 where they acquired all of the Twitter data from 2006 – 2010 [6] from the Twitter Company.

TWITTER HARVESTER

The first phase of a suggested system that will be able to gather all if not most of the available data from Social Networks can be called a harvester or gatherer tool. The Twitter Harvester is a tool designed and developed to gather and store all of the tweets that are generated by users of the system. Twitter Company provides an API that is necessary to use in order to store all of the available data within its system. [7] The tweets that the API provides are from users that do not have a private setting on their account, as well as it limits the amount of tweets returned by the API to 3,200. This type of limitation makes it necessary to start gathering any data from Twitter as soon as possible. Once a user has tweeted more than this amount limit it will be hard to acquire that data.

This harvester is an imperative portion of the bigger system, a system that will be able to analyze each tweet that is generated, a system that could display a graph of influence between users, a system that can mine text; basically a system with so many options each analyst will find something of interest.

Different technologies were used to develop this harvester, such as JSON, Python, CouchDB and Redis. [8] The book Mining the Social Web, contains lots of data as well as good examples that are basic introducing concepts and pieces of code that showcase a simple process to gather one user data. This book has been the guideline used to start the implementation of this system. But one can certainly replicate these processes in a much larger user base with some modifications to the examples. This book and the Twitter API provide all the necessary details in order to quickly implement such a system, and improve it to fit the desired needs.

This harvester once it is deployed will work by itself, going through the lists of users for all of their tweets, updating the users lists to make the harvest more efficient. The process to acquire the tweets is still in development to find the most efficient steps, the Twitter API comes with some limitations this to ease the burden of their own system while providing anyone interested in developing and deploying a Twitter application. For example, one of the limitations encountered was the amount of calls to the Twitter APIs that can be done, this limitation created the necessity of reviewing how may calls had been made or to look at messages from the API stating that the limit has been reached. Thus creating some validation points in order to have a system that never stops until it has reached the end.

The first phase of the harvester started with a limited set of users, around 15 users were used to make a feasible code work. As previously mentioned, the limitation of amount of calls to the Twitter API made it necessary to start with a handful of users, but once this process matured the users list increased to up to 300 users. After a few weeks with a list of 300 users, some issues started to show. Efficiency started to be the topic of the code that had been developed, it was seen that sometimes users did not write a single tweet between days, weeks or months. Many have been the twitter accounts found that the users never write a single tweet; so another script was created to analyze simply all of the gathered data. This new script generated different lists that help in the execution of the harvester. Three lists were created to separated these first 300 users in high, medium, low twitter usage. Once the code started to execute with better results, another script was created to increase the amount of users to up to 50,000 users.

Currently the harvester that has been developed and deployed is executed on demand, passing a parameter indicating which user list to use. Within a week of having this code running, new problems started to show up. Errors about suspended account and private accounts, thus creating two new lists to take into consideration when gathering the data. Since the access to users data is limited in Twitter, if a user has a security feature activated and it is not following back, the user created for this tool would need to have those private users to grant it permission to access their data.

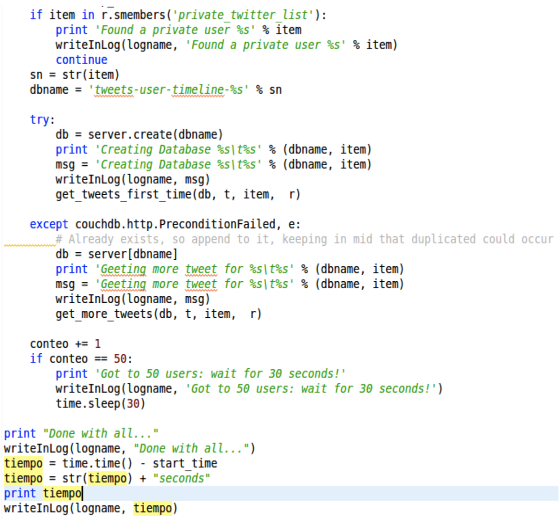
A change in the early development of the harvester was needed, when twitter screen names re changed. Twitter lets the users change their screen name whenever they desired, creating another problem when validating the lists of users. The Twitter API was consulted to find a solution to this problem, and the solution was to start using a user id created by Twitter. This change improved greatly the system in making the executions successful, easing our system in only looking at numbers and not names that could change frequently.

TECHNOLOGIES USED

As previously mentioned, the book "Mining the Social Web" was of great influence in deciding what technologies were to be used. It demonstrated how easy, simple and quick we could get an environment setup and running code that would accomplish the different ideas presented. Python, Redis, CouchDB and JSON were used because of the availability; being open source projects this facilitated the installation of the server where this project has been running. Python is the programming language were the harvester was written in [9], Redi [10] and CouchDB [11] are the databases used where the user lists and tweets are stored, finally JSON is used since the Twitter APIs return all of the data using this type of data-interchange format [12].

Python

Python is the programming language the Twitter Harvester is written in, a scripting language was decided to be used for its ease when it comes to the compilation process. Also, it has available modules that integrate with the other technologies selected and are simple to include in the environment. Familiarity with this programming language made it more easy to implement such system, this helped with timing of deployment since the sooner the system is running the more data it will acquire. Figure 1 below shows part of the code used in the harvester.

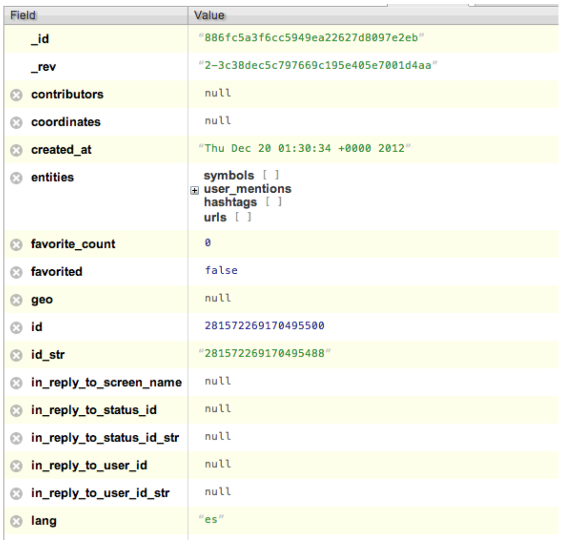


```
    if item in r.smembers('private_twitter_list'):
        print 'Found a private user %s' % item
        writeInLog(logname, 'Found a private user %s' % item)
        continue
    sn = str(item)
    dbname = 'tweets-user-timeline-%s' % sn

    try:
        db = server.create(dbname)
        print 'Creating Database %s\t%s' % (dbname, item)
        msg = 'Creating Database %s\t%s' % (dbname, item)
        writeInLog(logname, msg)
        get_tweets_first_time(db, t, item,  r)

    except couchdb.http.PreconditionFailed, e:
        # Already exists, so append to it, keeping in mid that duplicated could occur
        db = server[dbname]
        print 'Geeting more tweet for %s\t%s' % (dbname, item)
        msg = 'Geeting more tweet for %s\t%s' % (dbname, item)
        writeInLog(logname, msg)
        get_more_tweets(db, t, item,  r)

    conteo += 1
    if conteo == 50:
        print 'Got to 50 users: wait for 30 seconds!'
        writeInLog(logname, 'Got to 50 users: wait for 30 seconds!')
        time.sleep(30)

print "Done with all..."
writeInLog(logname, "Done with all...")
tiempo = time.time() - start_time
tiempo = str(tiempo) + "seconds"
print tiempo
writeInLog(logname, tiempo)
```

Figure 1
Python Code of the Harvester

CouchDB

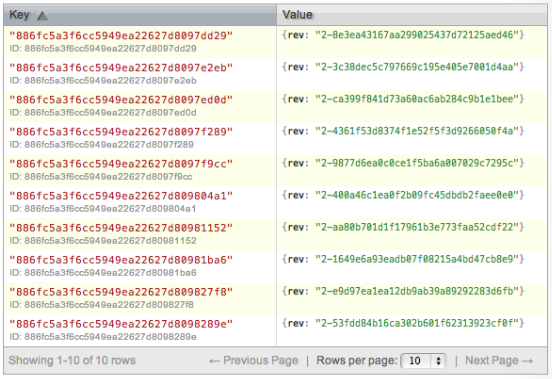
CouchDB is the database used for storing all of the users tweets, some of the advantages of using this database system are: it is a document based database system, no need of database schema to be defined, any change in the data being stored will be processed. Installation is fairly simple and easy; also integration with Python was quick and easy. When the harvester is gathering the data from the users it simply passes all of the tweets as they are received from the Twitter API and stored, each tweet is stored as an individual document. Figure 2 below shows an example of how tweets are stored. Each user has its own database, so in essence this database system is managing more the 50,000 users with thousands of tweets or documents being stored in each of the databases. Figure 3 below shows a list of the tweets stored from a user.



Field	Value
_id	"886fc5a3f6cc5949ea22627d8097e2eb"
_rev	"2-3c38dec5c797669c195e405e7001d4aa"
contributors	null
coordinates	null
created_at	"Thu Dec 20 01:30:34 +0000 2012"
entities	symbols [ ] ⊞ user_mentions hashtags [ ] urls [ ]
favorite_count	0
favorited	false
geo	null
id	281572269170495500
id_str	"281572269170495488"
in_reply_to_screen_name	null
in_reply_to_status_id	null
in_reply_to_status_id_str	null
in_reply_to_user_id	null
in_reply_to_user_id_str	null
lang	"es"

Figure 2
A Tweet Stored in CouchDB for a User

After managing to get more than the 50,000 databases in one server environment, it has been noticed that searching or reviewing the list of databases or even an individual database with thousands of documents makes the performance degrade.

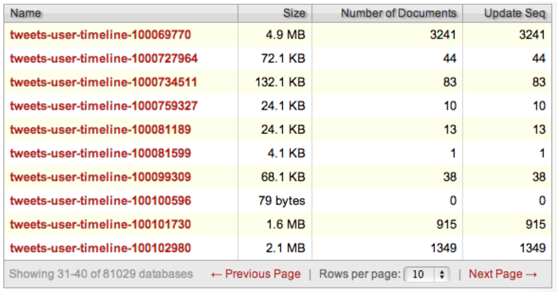


Key ▲	Value
"886fc5a3f6cc5949ea22627d8097dd29" ID: 886fc5a3f6cc5949ea22627d8097dd29	{rev: "2-8e3ea43167aa299025437d72125aed46"}
"886fc5a3f6cc5949ea22627d8097e2eb" ID: 886fc5a3f6cc5949ea22627d8097e2eb	{rev: "2-3c38dec5c797669c195e405e7001d4aa"}
"886fc5a3f6cc5949ea22627d8097ed0d" ID: 886fc5a3f6cc5949ea22627d8097ed0d	{rev: "2-ca399f841d73a60ac6ab284c9b1e1bee"}
"886fc5a3f6cc5949ea22627d8097f289" ID: 886fc5a3f6cc5949ea22627d8097f289	{rev: "2-4361f53d8374f1e52f5f3d9266050f4a"}
"886fc5a3f6cc5949ea22627d8097f9cc" ID: 886fc5a3f6cc5949ea22627d8097f9cc	{rev: "2-9877d6ea0c0ce1f5ba6a007029c7295c"}
"886fc5a3f6cc5949ea22627d809804a1" ID: 886fc5a3f6cc5949ea22627d809804a1	{rev: "2-400a46c1ea0f2b09fc45dbdb2faee0e0"}
"886fc5a3f6cc5949ea22627d80981152" ID: 886fc5a3f6cc5949ea22627d80981152	{rev: "2-aa80b701d1f17961b3e773faa52cdf22"}
"886fc5a3f6cc5949ea22627d80981ba6" ID: 886fc5a3f6cc5949ea22627d80981ba6	{rev: "2-1649e6a93eadb07f08215a4bd47cb8e9"}
"886fc5a3f6cc5949ea22627d809827f8" ID: 886fc5a3f6cc5949ea22627d809827f8	{rev: "2-e9d97ea1ea12db9ab39a89292283d6fb"}
"886fc5a3f6cc5949ea22627d8098289e" ID: 886fc5a3f6cc5949ea22627d8098289e	{rev: "2-53fdd84b16ca302b601f62313923cf0f"}

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

Figure 3
User Database Displaying all the Tweets Available

So the use of CouchDB as an intermediary storage or first step storage is an acceptable step, upon further reading in the Twitter website it was later found out that they use MySQL as their database system of choice. Such change in design for this harvester or this Social Networking analytical tool will need to be considered.



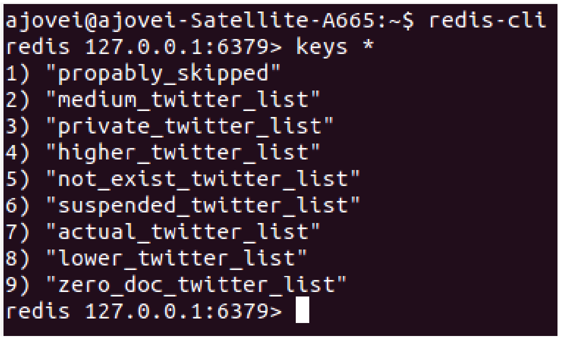
Name	Size	Number of Documents	Update Seq
tweets-user-timeline-100069770	4.9 MB	3241	3241
tweets-user-timeline-1000727964	72.1 KB	44	44
tweets-user-timeline-1000734511	132.1 KB	83	83
tweets-user-timeline-1000759327	24.1 KB	10	10
tweets-user-timeline-100081189	24.1 KB	13	13
tweets-user-timeline-100081599	4.1 KB	1	1
tweets-user-timeline-100099309	68.1 KB	38	38
tweets-user-timeline-100100596	79 bytes	0	0
tweets-user-timeline-100101730	1.6 MB	915	915
tweets-user-timeline-100102980	2.1 MB	1349	1349

Showing 31-40 of 81029 databases ← Previous Page | Rows per page: 10 | Next Page →

Figure 4
CouchDB List of Databases Available

Redis

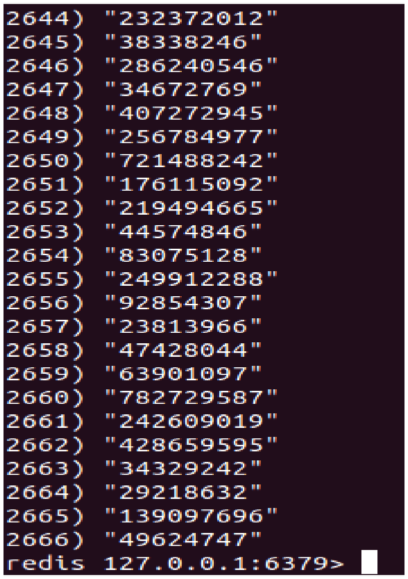
This version of NoSQL system was used because of the benefits it provides when managing sets, providing atomic operations after a quick installation and not much configuration nor design. This technology easily provides us the means of managing huge amounts of data, where applying basic set algebra makes all the difference. Also it provides it provides an auto comparison, this helps in eliminating duplication when used. Figure 3 shows all of the sets created to improve execution of the harvester.



```
ajovei@ajovei-Satellite-A665:~$ redis-cli
redis 127.0.0.1:6379> keys *
1) "propably_skipped"
2) "medium_twitter_list"
3) "private_twitter_list"
4) "higher_twitter_list"
5) "not_exist_twitter_list"
6) "suspended_twitter_list"
7) "actual_twitter_list"
8) "lower_twitter_list"
9) "zero_doc_twitter_list"
redis 127.0.0.1:6379> 
```

Figure 5
Redis List of Sets Available

Figure 4 below shows the elements of one of the sets showed above. These are generated depending on their activity; low, medium, high are targeted by their amount of tweets, they correspond to 1250 or less, between 1250 and 2500, and 2500 or more respectively. Private and Suspended list are generated when the harvester is running and either of these are found.



```
2644) "232372012"
2645) "38338246"
2646) "286240546"
2647) "34672769"
2648) "407272945"
2649) "256784977"
2650) "721488242"
2651) "176115092"
2652) "219494665"
2653) "44574846"
2654) "83075128"
2655) "249912288"
2656) "92854307"
2657) "23813966"
2658) "47428044"
2659) "63901097"
2660) "782729587"
2661) "242609019"
2662) "428659595"
2663) "34329242"
2664) "29218632"
2665) "139097696"
2666) "49624747"
redis 127.0.0.1:6379>
```

Figure 6

Portion of the higher_twitter_list

CONCLUSION

This project is the start of a much bigger system, one that will eventually spread to access and store data from all of the Social Networking sites that are available and will become available. The need for such a system is of importance to many different industries that are creating and providing a place with access to this data is of mutual benefit. As previously mentioned, the infrastructure currently needed to house only the Twitter data is easy to attain and create. But constant improvements and upgrades will be needed every other year, in order to be capable of providing the analytical tools and keep an increasing warehouse stable.

I was able to create such a tool that connects to Twitter and gathers all of the tweets for a defined user list, that later improved the capabilities of increasing the user list. Other scripts were developed to help with the ranking of user accounts to help create smaller lists to gather their tweets at a faster pace. This system will keep improving its methods of gathering the users data, analyzing new parameters that could be added in the data being gathered. Constant change in the data has to be monitored in order to keep the system healthy and up to date. Currently the tweets are being stored using CouchDB, this storage will be an intermediary storage system since the access to all the data that is currently available takes too much time, and a final storage unit will be developed in order to maximize the efficiency desired.

Technologies will be reviewed from time to time to help grow the system; innovative technologies such as CouchDB and Redis were implemented thanks to their ease of use, configuration, and implementation. These new technologies I learned in order to develop this system and I have started using them in other personal / professional projects. The next step to follow with the development of this system is to find the right technologies to start providing analysis of the available data. Or to improve the server implementation in order to have data accessible faster, a more robust deployment needs to be planned out switching the execution from on-demand to automation.

The seed has been planted in creating a system that will be able to mine social networks, the first phase will always have be creating a harvester. Without data what is there to mine, using a harvester we can have access to a universe of data that otherwise take too much time sitting in front of a computer reading and writing notes. Later different concepts of analysis can be implemented and studied.

REFERENCES

[1] "The fastest, simplest way to stay close to everything you care about.", Retrieved on October 14, 2013, https://twitter.com/about

[2] "#TwitterRevolution.", Retrieved on October 14, 2013, http://www.cnbc.com/id/100792286

[3] "Terms of Service", Retrieved on October 15, 2013, https://twitter.com/tos

[4] "Developer Rules of the Road", Retrieved on October 15, 2013, https://dev.twitter.com/terms/api-terms

[5] "Late to the Feast: Newspapers as Historical Sources", Retrieved on October 15, 2013, http://www.historians.org/perspectives/issues/1993/9310/9310ARC.cfm

[6] "How Tweet It Is!: Library Acquires Entire Twitter Archive", Retrieved on October 15, 2013, http://blogs.loc.gov/loc/2010/04/how-tweet-it-is-library-acquires-entire-twitter-

archive/?doing_wp_cron=1381897725.0348920822143554687500

[7] "Developer Website for Twitter API", Retrieved on October 16, 2013, https://dev.twitter.com/

[8] Russell, Matthew, A., "Preface",Mining the Social Web Vol. No. #, date, XV - XVII.

[9] "Python Programming Language – Official Website ", Retrieved on October 17, 2013, http://www.python.org/

[10] "Redis Official Website ", Retrieved on October 17, 2013, http://redis.io/

[11] "CouchDB Official Website ", Retrieved on October 17, 2013, http://couchdb.apache.org/

[12] "Introducing JSON", Retrieved on October 17, 2013, http://www.json.org/