www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 4 April, 2014 Page No. 5379-5383

Twitance
Twitter Recommendation and Analytic Tool

Ashish Singh, Rohit Khatana, Jitendra Kumar, Aravind S, Gautam Rege, Geeta Patil

Student in Department of Information Technology, Army Institute of Technology, Pune, India
Student in Department of Information Technology, Army Institute of Technology, Pune, India
Student in Department of Information Technology, Army Institute of Technology, Pune, India
Student in Department of Information Technology, Army Institute of Technology, Pune, India
Co-Founder & Managing Director, Josh Software Pvt Ltd., Shivaji Nagar, Pune, India
Asst Professor, Department of Information Technology, Army Institute of Technology, Pune, India

***Abstract* — Twitance is a recommendation and analytic tool for Twitter which would help Twitter users to find people on their particular interest so as to better engage with the audience. It works as a search engine of people. A website on which user can search for his topic related Twitter users and then afterwards use them as mentions in his tweet to increase his audience. Later on the user will be provided with the reach of his tweet. A graphical representation of who all have re-tweeted his tweet, made his tweet favorite, replied to his tweet and who among those re-tweeters have greater influence. He also gets to know who are the people who have similar interests to him.**

Keywords - ROR, Elasticsearch, Tire, Mongo-id, Snowball, Ripple Graph etc.

I.INTRODUCTION

Usually every Twitter User desires that his tweet reaches to the maximum audience and receive an ample amount of response over that. Also, people like finding others who have similar interests to theirs.

Twitance is a people search engine which provides its users with the particular users related to their search query. They could improvise their search results by removing the tags which do not match to their query, as particular query itself can have different meanings associated with it i.e. let us suppose the user searches for query over "apple" the results would be shown both for the "Apple" Company and the fruit "apple". This works with the help of Elasticsearch and Twitter API. The tweets along with the information are downloaded via the Twitter API and the data is stored and analyzed through the Ruby interface for elasticsearch i.e. Tire. Further saved searches are cached for later to get faster results. After this the user will be provided with the analytics of the tweet. With a pictorial representation the user will be shown who all were responsible for his tweet influence. People with large number of followers who had re-tweeted his tweet would become his greatest influencers and have bigger circles. Thus this will be shown using a ripple graph.

II.RELATED WORK

Till date there are many applications based on Twitter like Tweetreach, which gives you an idea about the re-tweets and list of contributors in extending your tweet reach. Another app named Klout measures the size of your social media network, it measures your influence using data points from Twitter such as following count, followers, re-tweets. Klout score are supplemented with true reach, amplification and network impact.

Tweet Cloud is a services used to generate a cloud of most used words within your tweets. It counts only the words and shows most used word accordingly how many times it has been used. It contains a time-line span by which we can generate cloud of only those tweets which lies between the given time-span.

Vocus was quite close to what we are planning to do but they are not doing a depth wise analysis of tweets. They have gender wise and location wise demographics. Their public beta is still not out and they are not just for twitter.

ManageFlitter provides its registered users information regarding its un-follows, people search that too to those who are registered with that app. It provides you with the Power Post, analytics, un-follow, follow and people search as its features. Its only drawback being that it provides information of only the registered users to that app.

Tweet-effect shows the tweet timeline with your gain/lose followers timeline to determine which tweet makes you lose/gain followers. It analyzes the latest 200 tweets and highlights tweets with your losing or gaining followers. Various other applications i.e. Twitanalyzer, Microplaza, Twist, Twitturly etc. are there but the ideology of Twitance is different from these.

III. OVERVIEW

Our product allows a user to search for people by entering their queries and finding relevant results to their interests. If they are not satisfied with the results they can further refine those results to get a better outcome.

A. TWITANCE

Twitance provides people search without any registration and provides information of all the people on Twitter. It is not affecting the privacy of any twitter user, based on the public resources it is providing its results to the users. Twitance is a Ruby On Rails application. The user will be able to access site without any authentication to a search bar where he can enter his query and on based on that query user will be receiving tags along with a list of top users related to that topic.
It also focuses on machine learning that if the user is satisfied with the search results, those results are stored and after-wards it can be accessed faster for the user with the same query.

B. RUBY ON RAILS

The site is based on Ruby on Rails is a framework for building websites that can make it more affordable to create and maintain your site while simultaneously offering improved performance and faster development times. The five big reasons why Ruby on Rails would be:

-Significant Cost Savings:
Ruby on Rails is essentially a free development toolkit, which runs on a free operating system (Linux) and works with multiple databases and web servers (most of which are free). By using a cost-free platform, we are able to significantly reduce costs without sacrificing any speed, security or performance.

-Rapid Development:
Ruby on Rails is a rapid application development tool which allows us to model out website feature quickly. We can go from modeling and estimating to actual development very rapidly.

-"The Ruby Way":
The way Ruby on Rails was created, there is consistency in structure and methodology when writing code. The Model-View-Controller architecture that Ruby on Rails uses makes it a lot easier to manage the code between developers. This means than an individual developer's "coding style" doesn't get in the way of writing the code, so passing off code from one developer to another involves a much shorter learning curve.

-Collaboration:
The Ruby development community is extremely active and responsive. People are constantly developing code for talking with other APIs; as such, our developers have a much larger and more diverse toolkit to lean on.

-Future Demand and Adoption:
Since website are moving further and further away from being static hubs of information and are becoming much more dynamic and interactive, lots of the newer web services that have launched recently run Ruby on Rails.eg: Hulu, Groupon, LivingSocial, SoundCloud, Twitter, YellowPages, Shopify, SlideShare, GitHub etc.

C. ELASTICSEARCH

Tweets are analyzed with the help of elasticsearch which is used by Tire gem (client for the Elasticsearch search engine/ data-base). It provides Ruby-like API for fluent communication with the Elasticsearch server and blends with Active Model calls for convenient usage in Rails applications. Elasticsearch consists of Index, Analyzers and Data. Index in this is similar to a virtual namespace that points to all the shards the data lives in these shards. Shards are nothing but different databases we get after dividing our original database horizontally. A standard analyzer consists of a standard tokenizer, lowercase filter, English stemmer and stop-words filter. In our application we are using the snowball analyzer, which is similar to standard analyzer except it having the snowball filter. Snowball follows a particular algorithm which causes the following conversions.[17] Words ending with :

- SSES becomes SS
- IES becomes I
- SS remains as SS

And few more conversions based on the vowel and consonants in those words. This includes the functionality of filtering and faceting which helps in improving the search results processing.

Searching in Elastic search includes two steps:
1. Matching all documents that meet the given criteria.
2. Giving every document its own score on the basis of scoring algorithm then sort them in descending order.

Searching will be done using the usual REST full API (web requests). All elasticsearch queries boil down to the task of restricting the result set, scoring, then sorting. The main algorithm governing the scoring of documents is TFIDF (implemented in lucene). Lucene's general strategy is to first exclude all documents with no matches for search terms, then rank the documents that do match.
A document's score will be higher when:

- When the matched term is rare, which is to say that it is found in fewer documents than other terms.
- The term appears in the document at a greater frequency than other terms within the document.
- If multiple terms are in the query and the document contains more of the query's terms than other document contains more of the query's terms than other documents.
- The filed or document had a boost factor specified at index-time or query time.[16]

The search API provided by the search endpoint can be used for both /index as well as /index/type paths. The search endpoint works with both the GET and POST HTTP methods.
The Twitter API consists of two types that includes Streaming API and the Search API.

Twitter Search API is only for minimum hitting the twitter servers. They are made for only getting tweets from a particular account or finding a set of tweets with specific keyword. If we want to hit Twitter's server at extreme velocity and are hitting its rate limits then we should use Streaming API. [12]

If we need to make a data mining project or analytic project and we need highly intensive data from Twitter then this Streaming API can be helpful.

D. TWITTER SEARCH API

Twitter search API which is used to download the tweets is a part of Twitter's v1.1 REST API. It allows queries against the

indices of recent or popular Tweets. It is important to know that the Search API is focused on relevance and not completeness means that some Tweets and users may be missing from search results. If you want to match for completeness then you should consider using a Streaming API instead. [15] Twitter offers advanced search functionality which includes:

- Boolean:
  In this we have the AND, OR and EXACT PHRASE opts. By default AND is treated between terms, OR results in either of the terms present. EXACT matches the whole turn inputed.

- Subtractive:
  We can specify words which we do not want to see in our search results. Tweets containing such words will not appear in search results. We can add as many words we want. For example, if the query is shahrukh khan –deepika –BigB –Priyanka, then the tweets which contain deepika, BigB and Priyanka as words in them will be exempted among the tweets which we receive for the search.
- Sentiment based:
  When we search on twitter we get results containing both the positive and negative sentiments. Users can specify what sentiments they are expecting in the tweets.[10]

- By Time search:
  It is very useful in searching when we are only interested in results in time frame not older than say yesterday.

A few handy ways to refine search in Twitter includes:
- Suffix the search term with re-tweet to get the results which do contain re-tweets.

- If you want to see only those tweets that have links suffix it with "filter: links"

- For tweets from a certain user, suffix the search term with "from: username"
- We can also search for sentiment categorized tweets by adding suffix as :(, :), ? for negative, positive and question.

So as to use this, the URL needs to be encoded before sending the query result to twitter.URL encoding converts characters into a format that can be transmitted over the Internet. Since URLs often contain characters outside the ASCII set, the URL has to be converted into a valid ASCII format. URL encoding replaces usage ASCII characters with a "%" followed by two hexadecimal digits. URLs cannot contain spaces. URL encoding normally replaces a space with a plus (+) sign or with %20. [7][13]

IV. WORKING AND IMPLEMTATION

The process starts of with the user entering his topic of search, this query is hit on to the Twitter API, which returns a JSON document which consists of the tweets related to our search. This JSON document is provided to the Tire which has a mapping for the database.

```
task :mapping do
          Tire.index 'twitances'  do
    delete
          create :settings => {
      :index => {
        :analysis => {
          :analyzer => {
            :twitance_analyzer => {
              :type => 'snowball',
              :tokenizer => 'snowball',
              :language => 'English',
              :stopwords_path =>
'/stop.txt'            }
          }}}},
      :mappings => {
        :twitance => {
          :properties => {
            :handle=> { :type => 'string'},
            :followers=> {:type      =>
'integer'},

:friends=> {:type => 'integer'},
            :name=> {:type  => 'string'},
            :profile_image_url=> {:type=>
'string'},
            :description=> {:type=>
'string', :analyzer => 'twitance_analyzer'},
            :verified=> {:type=> 'boolean'},
            :tweet=> { :type => 'string',
:analyzer => 'twitance_analyzer'}}}}
  end
end
```

Fig.1. Mapping made for the Tire so as to store the tweets.

Now this database consists of all the tweets along with the information related to that tweet including the username of the tweeter, his/her public info. This data is pushed on to the Search results webpage after the tweets are analyzed through the Snowball, which checks out the tweets and finds the score of each word. Thus a list of most repeated words in those tweets are presented to the user in an ascending order. These words become the relevant tags and along with this a list of users is generated who have used these relevant tags in their tweets. Among the Suggested Twitter users, the user using the words with higher score will top the list and then the next and so on. As a part of machine learning people who are more popular on twitter will be analyzed for their particular topics and these too will be provided to the user. The user can refine his search results by removing the irrelevant tags by clicking on

the crosses provided. As soon as the user starts clicking on these crosses thus removing the tags, the list of Suggested people too gets refined as the people related to those tags removed are terminated.

If the user is able to gain his wants from these results, he may approve them and it can be used for future reference, this making it faster. It is obvious that for a particular search there can be different meanings and it is upon the search engine to maintain to whatever is relevant. Pre-storing the results which were adequate will help in solving this .That is if a new user enters a similar search to those previous ones, then he would be provided a few suggestions over the last searches made similar to his/her current search. This would fasten up the search process and in a way machine learning comes into action.

V. EXPERIMENT AND RESULTS

As for the demo part we searched for "jockey" through our search engine and provided us with the results as shown below. The left side consists of the relevant words related to the search and the search results are shown below which consists of the users who have recently tweeted on that particular topic according to their individual scores. As the word jockey could mean both the horse-rider "jockey" or the garment company named "jockey". Thus the user can remove the tags which do match to his search.

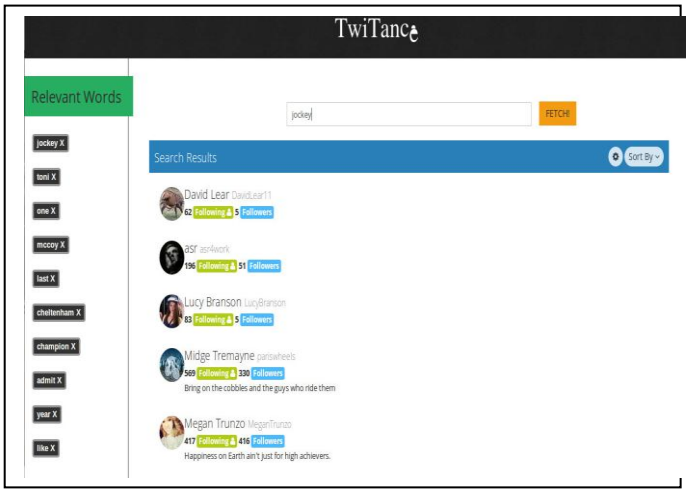

Fig.2. Output after the user enters his query,showing its relevant tags and people related.

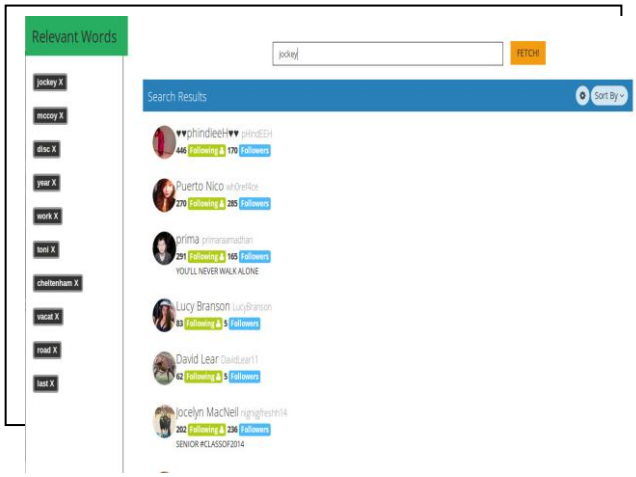

Fig.3. Results refined after removing a few tags from the relevant tags and thus change in the list of the users in its output.

In the current scenario the user has removed the tags "toni" and "one". As soon as the user removed these there is a change in the list of recommended people as shown below (Fig.3). The user can further remove un-relevant tags until he is pleased with the results. Then the user will be asked whether he was satisfied with the results and his feedback will be saved. Which help both him/her or the future users who search for similar queries.

The next part includes providing the user with the information regarding the reach of his tweet. This includes the people who re-tweeted, favorited or replied to them. This is shown to the user via a graphical representation that includes all the users from the one who posted the tweet to whom all it spread. The ones who had re-tweeted it and are having large number of followers will be considered as influencers. These influencers might be useful if the user is tweeting regarding the same topic in future. He will be known who have similar interests with him and will be likely increasing his followers/following list.[2]

The influencers are represented in circles with his followers inscribed in his circle. Among those followers if someone re-tweets then that influencer will have a small circle of his own that too inscribed in the bigger circle as shown in figure below (Fig.4).

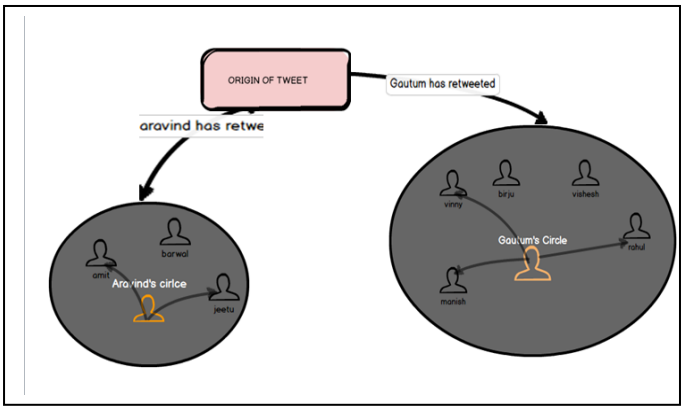

Fig.4. Tweet Analytics prototype, showing the origin of tweet and those who re-tweeted and their circles defining their influence.

The links among the re-tweeters of the tweet will be generated time to time as it is being re-tweeeted. This way the user can easily accumulate his influencers. [1]

ACKNOWLEDGMENT

First of all I would like to express our profound sense of gratitude towards our guides Gautam Rege and Setupathi Ashokan, Josh Software, for their valuable guidance, support and encouragement throughout the period this work was carried out .Their readiness for consultation at all times, their educative comments, their concern and assistance even with practical

things have been invaluable. I would also like to convey our sincerest gratitude and indebtedness to Mrs. Geeta Patil, Department of Information Technology, AIT, who bestowed their effort and guidance at appropriate times without which it would have been very difficult on our part to complete this project. I also thank to Dr. V.P. Gosavi, Principal of Army Institute of Technology for providing me all the necessary facilities to carry out the project. A vote of thanks to all my friends and my parents who stood by me whenever I was in any sort of difficulty.

REFERENCES

[1] Rohan D.W Perara CERDEC Ft Monmouth, NJ "Twitter Analytics: Architecture, Tools and Analysis", 2010 Military Communications Conference.

[2] Ming Hao;Hewlett-Packar Labs, Palo Alto,CA,USA"Visual sentiment analysis on twitter data streams",Visual Analystics Science.

[3] J O'Donovan "Credibility in Context: An Analysis of Feature Distributions in Twitter" ,Third IEEE International Conference on Visual Analytics.

[4] Tim Mezies on "Software Analytics: so what?",IEEE Computer Society.

[5] http://blog.mashape.com/post/48074869493/list-of-40-machine-learning-apis

[6] https://www.mashape.com/duckduckgo/duckduckgo-zero-click-info#!endpoint-Zero-Click-Info

[7] https://www.mashape.com/tdguest/query-classification#!documentation

[8] http://www.idilia.com/developer/

[9] https://www.mashape.com/peerreach/peerreach#!documentation

[10] https://www.mashape.com/intridea/tweetsentiments#!documentation

[11] http://socialmediatoday.com/node/111483

[12] https://dev.twitter.com/docs/platform-objects/users

[13] https://dev.twitter.com/docs/api/1.1

[14] http://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf

[15] https://support.twitter.com/articles/71577-using-advanced-search

[16] http://exploringelasticsearch.com/

[17] https://github.com/karmi/retire