***Scenarios & Sentiment Analysis***

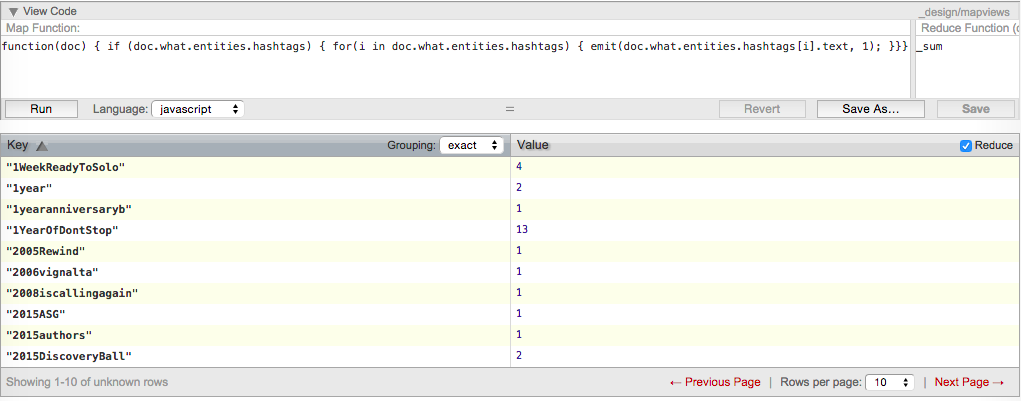
Scenario analysis in this project consisted by some with sentiment analysis and some without it. A good scenario analysis could be used to infer people’s attitude towards a specific topic or event or even a trend in certain areas of industry and so on. Therefore, several interesting scenarios have been analysed through three following processes.

***Map/Reduce:***

Map/Reduce is used to generate views needed by scenarios and sentiment analysis. Map/Reduce is a programming paradigm and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster. [1] In CouchDB, Map is implemented as a JavaScript function that maps view keys to values, and returns a list of key value pairs. Reduce is implemented to reduce the list to a single value. The result of Map/Reduce is a stored B+ tree named view.

To do the analysis, the first step would be using Map/Reduce capabilities provided by CouchDB to aggregate tweets that are associated with a specific scenario. Some of the Map/Reduce functions could simply aim at searching whether a given term or string is appeared in the tweets (e.g. the following scenario 2, topic of C2E2). Others could be very complex, such as to find out the most hot topics users discussing about or the top 10 users with most followers (seeing in scenario 1 - hot topics and scenario 5 – followers distribution respectively).

Take the finding most hot topics as an example. Map function could be implemented as to search whether a tweet contains any hashtags. If it does, the function will emit a list of key/value pairs as <hashtag\_name, 1> for every hashtag. The result list then passed to the Reduce function. In there, the count of hashtags that have the same name will be summed up. However, since CouchDB only supports sorting by key, a list like below will be returned which is not what is expected. Therefore, another function is required as a filter to sort the list by the sum-up values, instead of the keys.



The scenario of 1, 3, 4, 5 and 6 mainly use the tweets retrieved form Map/Reduce directly, while scenario 2 and 7 needs more processes on tweets. That is pro-processing and sentiment analysis.

***Pro-Processing:***

Pro-processing on tweets particularly means to parse the ‘text’ in tweets. Since natural language is significant complicated for a machine to understand and process, certain necessary pre-processing on ‘text’ are indispensable. In this project, this function is implemented in ‘TextParser.py’. The ‘TextParser.py’ is responsible for the following tasks.

1. Delete meaningless words appearing in ‘text’. The meaningless word means that this word is not useful and helpful to determine the sentiment of a ‘text’. In this project, these words are named as ‘stop words’. Stop words are various. For instance, the majority of them could be a pronoun (e.g. ‘she’, ‘they’…), a preposition (e.g. ‘about’, ‘to‘…). Some of them could be an adverb, such as ‘how’, ‘ever’ and so on. In this project, all the stop words are collected manually and stored in a data file named ‘stop\_words.txt’.
2. A special scenario is also been considered that some people would like to repeat certain characters in a string to express their emphasizing. (e.g. instead of using ‘bad’, people might use ‘baddddddddd’ to address their feelings.) In this case, deleting those repeated characters is necessary.
3. Deleting mentioned tweeters(@), hashtags(#) and URLs in ‘text’. Since the mentioned tweeters and hashtags show only very limited help in sentiment analysis, they are deleted in the implementation. Cause basically, they are just nouns of topics or names which could used to detect what this tweet is talking about. However, the most general way to obtain topic relevant tweets is by searching whether a certain term or string of this topic appears in the ‘text’ using Map/Reduce. Thus, these tweets are already associated with the topic, which makes the fields of mentioned tweeters and hashtags are no longer useful for sentiment analysis.
4. After the processes above, a new ‘text’ is generated and ready to be tagged its sentiment.

***Sentiment Tagging:***

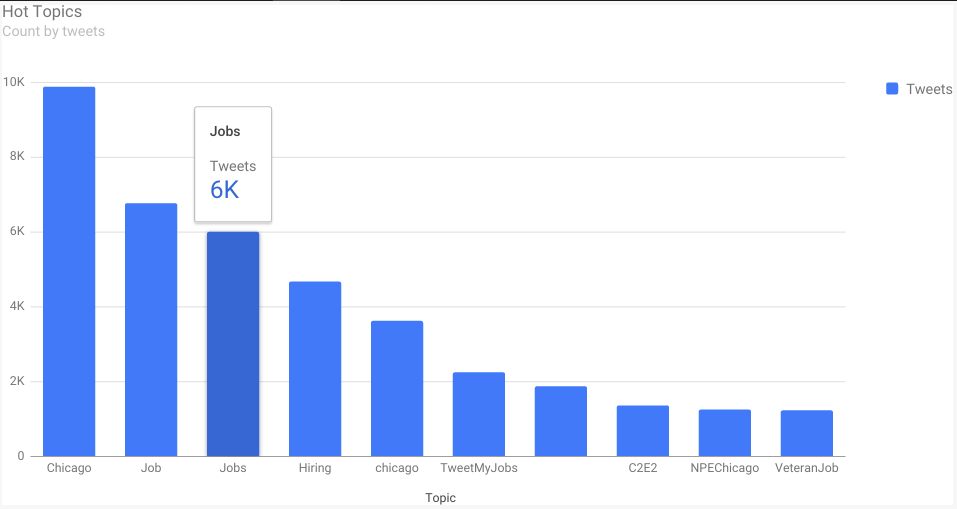
The last step is tagging the sentiment field based on analysing ‘text’ of each tweet. TextBlob is used in this project as the tool to tag each tweet’s sentiment. TextBlob is a Python library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as noun phrase extraction, sentiment analysis etc. [2] Both the build-in classifiers ‘PatternAnalyzer’ and ‘NaiveBayesAnalyzer’ are experimented in this project. Because of ‘PatternAnalyzer’ showing more accuracy, it is chosen as the final classifier to do sentiment analysis.

Using this analysis method, tweets retrieved from the database will be add another filed named ‘sentiment’, with the values of its sentiment classification (positive/negative/neutral) and sentiment score (where [-1,0) represents negative, 0 represents neutral, and (0, 1] represents positive).

Source codes of pro-processing and sentiment tagging are under the directory of ‘sentiment’.

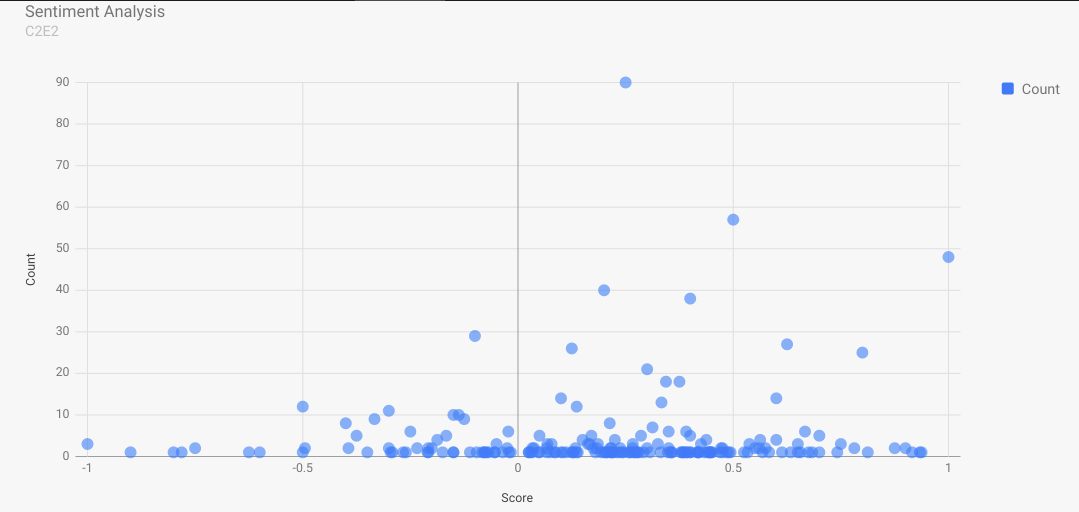
*Scenario Analysis 1: Hot Topic*

The hot topics are statistical hashtags in tweets that users mentioned most frequently. Figure 1 illustrates the most popular 10 topics in Chicago. It shows that three topics (They are Job, Jobs, and Hiring respectively) are highly associated with finding jobs among the top five ones. This phenomenon could probably reflect a fact that the employment market in Chicago is still weak and certain amount of people are experiencing the difficulties to find a job.



*Scenario Analysis 2: Topic of C2E2*

C2E2 is an example of a specific topic chosen from the hot topics above. C2E2 is the abbreviation of Chicago Comic & Entertainment Expo. Figure 2 shows the sentiment distribution of tweeters regarding C2E2. Overall, it can be seen that people in Chicago hold more positive attitudes towards C2E2 than negative one, which expresses that they love Comic and the Comic-relevant things.

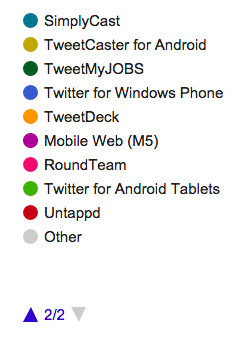
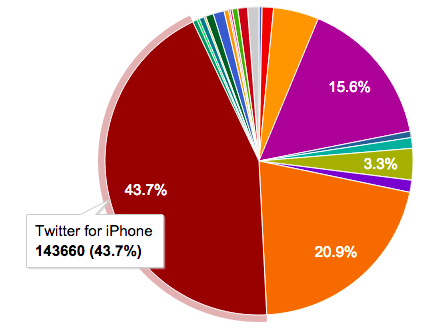


Based on the overall sentiment results, a further analysis is conducted to see what specific opinions hold by users to C2E2. This function is implemented in ‘text\_analysis.py’ and the result is shown below. The result presents that lots of users have feelings such as ‘thanks’, ‘love’, ‘great’, ‘best’ to C2E2.

[(u'c2e2', 694), (u'chicago', 219), (u'mccormick', 189), (u'comic', 114), (u'table', 90), (u'entertainment', 67), (u'expo', 63), (u'panel', 60), (u'thanks', 59), (u'time', 54), (u'booth', 51), (u'love', 44), (u'move', 44), (u'great', 43), (u'best', 41), (u'convention', 38), (u'trooper', 35), (u'storm', 35), (u'clone', 34), (u'chicks', 34)]

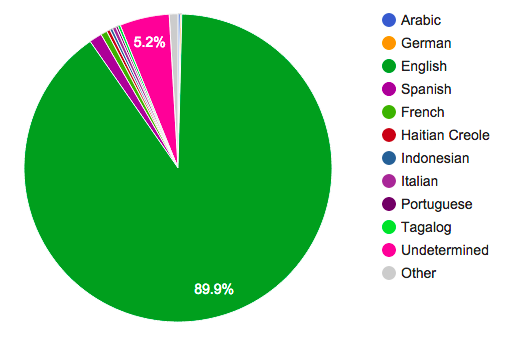
*Scenarios Analysis 3: Device Distribution*

As the development of industry technology, devices used to access Internet are more and more various. In aspect of product popularity or other analysis, it is significant to detect what devices are used most by people. Figure 2 demonstrates the details of the distribution of devices people used to post tweets. As shown in the figure, iPhone is the most popular device with 43.7% users using it, following by android device that owns 20.9% users. An interesting phenomenon that only 3.3% Twitter users post tweets from the Twitter Web Client can be found from this figure. Compared with the percentages of either Twitter for iPhone or Twitter for Android, this number could be an evidence that mobile device is the overwhelming tendency in the future’s IT industry development.



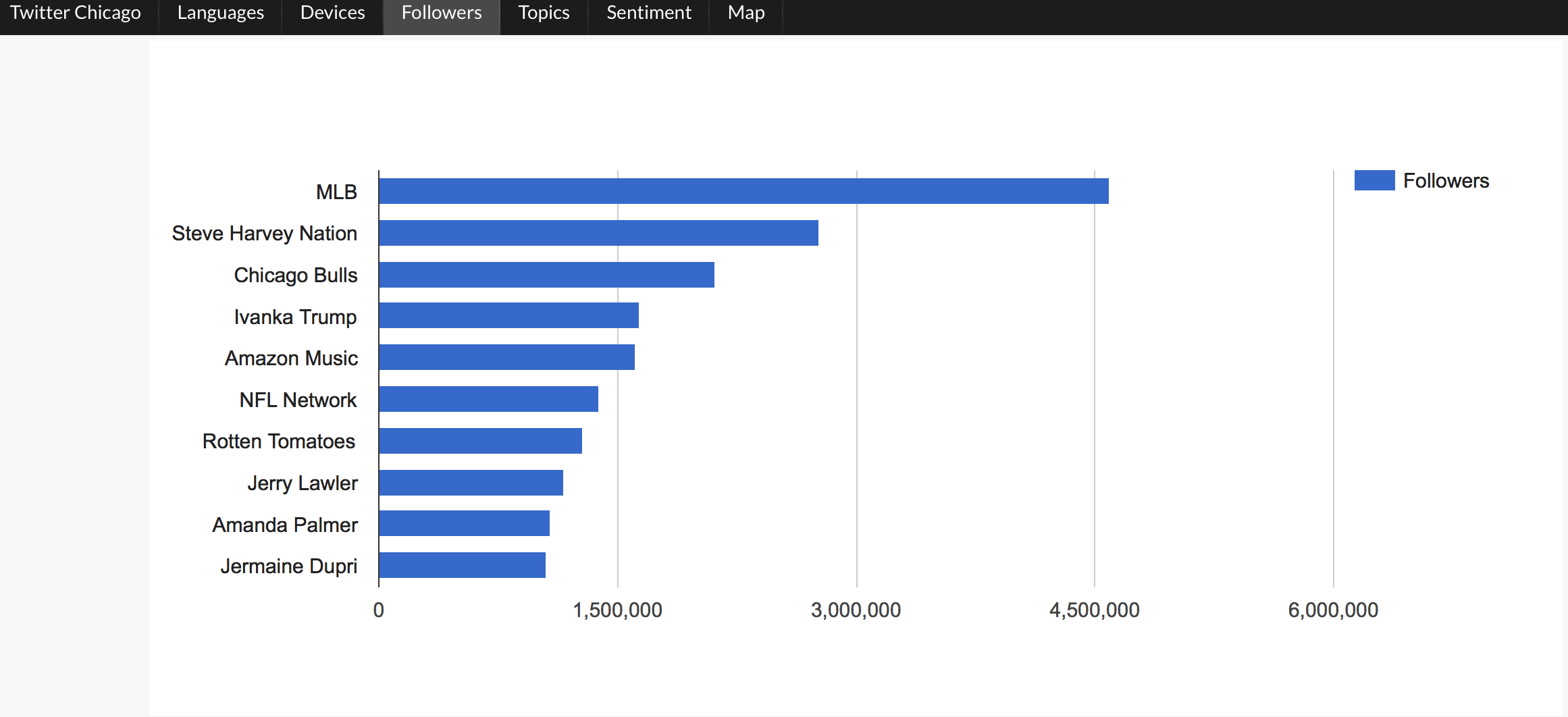
*Scenario Analysis 4: Language Distribution*

Since Chicago is a famous metropolis, people from around the globe bring their cultures here and throw them into this big pot. Thus, an analysis based on the language distribution is also conducted to see what kinds of language are daily used by people in Chicago. Figure 2 shows the language distribution of those tweets. Instead of English, the most widely used language is Italian (1.3%), then followed by French (0.7%), which means lots of Italian and French are living in Chicago. However, there are 5.2% users cannot be determined using which language. The reason could lies in that these users probably are Chinese, Japanese, Korean or other foreigners that their languages are dramatically different with English and therefore are difficult to be detected and recognized.



*Scenario Analysis 5: Followers Distribution*

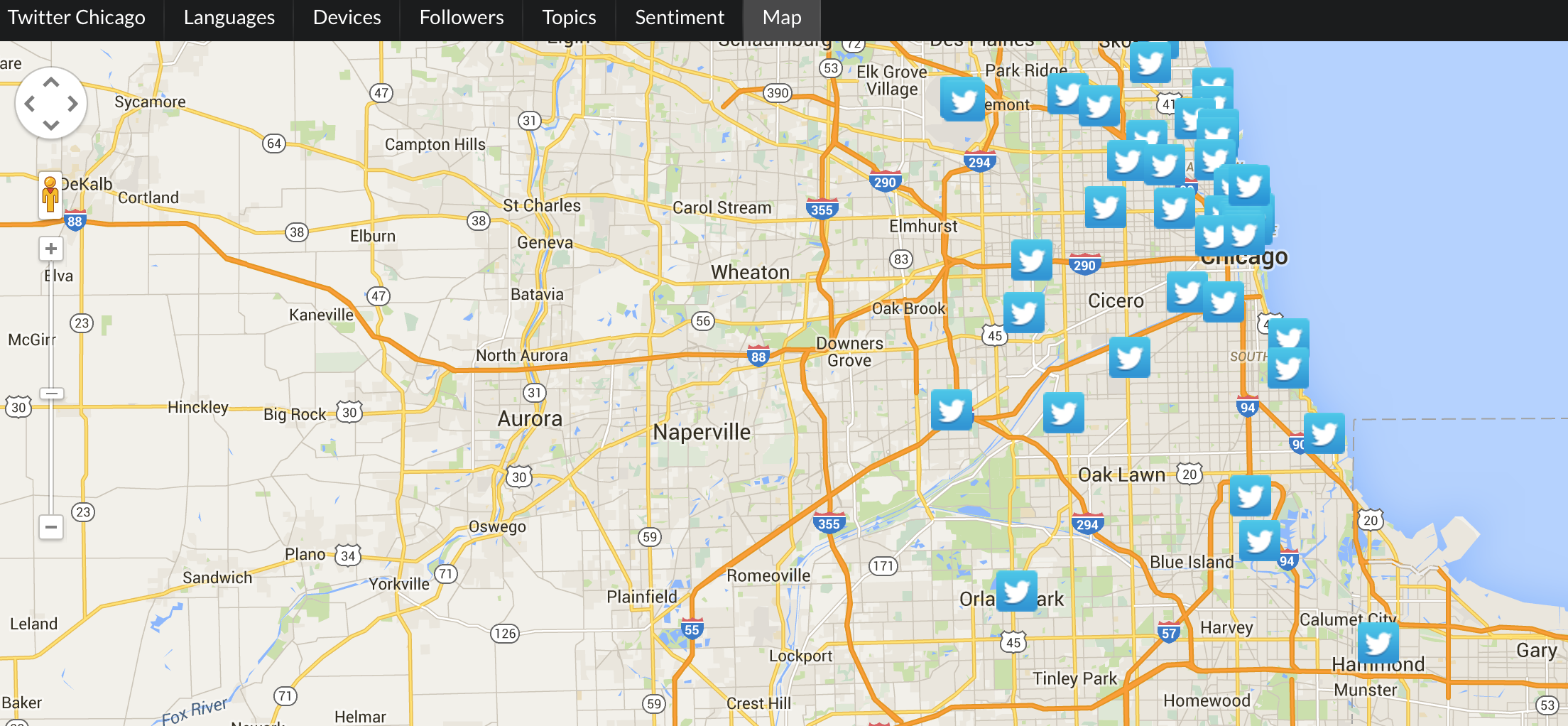
According to this follower distribution, the MLB is the most famous person and has 4,5 million followers. The reason why MLB has heaps of followers is that MLB is a popular sport in US and this MLB account may periodically announce MLB latest news and highlight replays so that these information attract followers. The second person who has most follower is Steve Harvey. He is a comedian, author and celebrity. He usually uploads video and shares something fun with audiences so people love to review what he preforms because of feeling happiness by these videos. The third one is Chicago Bulls which is the famous basketball team so it is a similar features regarding to MLB. The diagram XXX depicts the results of the people who have most followers in Chicago



*Scenario Analysis 6: Latest Tweets*

The majority of tweet data are sent by the main areas in Chicago, representing urban people typical are more enjoyable to utilize social network app. In the urban area, perhaps, most of people take the train to the workplace so that they may have free time to surf the Internet during the transportation period as well as post the information by twitter. Urban lifestyle also is strongly connected with the social network, building the connection with each other. Based on this, this may enable urban people who frequently utilize twitter more than rural people. The diagram illustrates the result of locations of the latest tweet data.

.



[1] http://en.wikipedia.org/wiki/MapReduce

[2] <http://textblob.readthedocs.org/en/dev/>