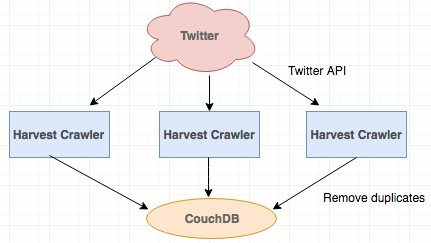
2. Data crawler:

Since we need a large tweets dataset for the analysis part in section 7 and 8, two different kinds of data crawlers are chosen in this report, which are REST and STREAMING APIs, provided by twitter. We implemented both of them into different VM (virtual machine) to persistently get tweets from twitter and store these tweets into CouchDB on each VM. After about 15 days crawling data, we finally obtained about 8.1G tweets data on our CouchDB. However, with the help of twitter id, it’s easy for us to recognize the duplicates and add a unique one into CouchDB application. The architecture for data harvest is as below:

**

*Figure. 1 Architecture for data harvest*

2.1 REST API

2.2 STREAMING API

Comparing to Rest API, Streaming API can give developers a lower latency access to twitter’s global stream of tweet data.

Also, Streaming API has a persistent HTTP connection from client side to server side so that it’s a real-time delivery and can ensure the completeness of tweets as it can gain more tweets than Rest API. For example, to Rest API, it works as request and response way, meaning that server reply back to client as response. Streaming API is different in respect of response, since server would send response to client whenever a tweet is available, so this is a type of socket program where continue stream of responses will come to client from server and the message will appear into the stream immediately with minimal overhead.

Besides, unlike Rest API, there is not rate limited with Streaming API, but a small fraction of the total volume of tweets would be returned.

7. Basic Scenario Analysis

The scenario for the project is to analysis the tweets in Melbourne with a list of “Fast Food” keywords. We will do sentiment analysis for the basic scenario and find a correlation between “fast food” and death in an extra extension scenario.

7.1 Map/Reduce

Map/Reduce is a programming paradigm for processing large datasets with a distributed and parallel algorithm on a cluster. In CouchDB, the Map function can maps keys to values, and generate a list of <key, value> pairs. The Reduce function is to reduce a list of same key to a single value. And the result for Map/Reduce is a stored B+ tree, which is named “view”.

The basic analysis in this project, we use the in-built Map function in CouchDB to create a view by searching a list of “Fast Food” keywords in tweet text. If it does, Map function will emit a list <key, value> pairs as figure2, key is tweet id and value is the tweet text.

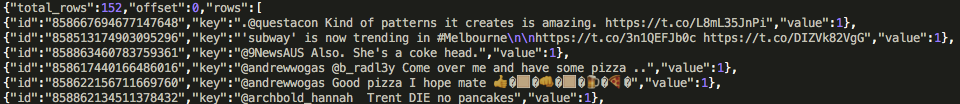
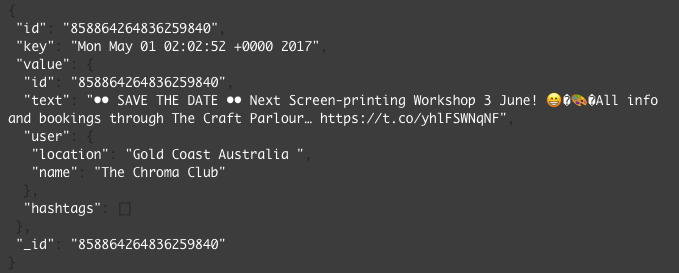


Figure.2 Example of Map <key,value> pair

Besides, to key or value part, we can put multiple information. It’s very helpful for us to create a view, since if we have multiple limitations in key part, we can get a more specific view and if we have multiple values in value part, we can combine these values for analyzing.

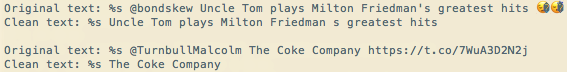


7.2 Sentiment Analysis

The first basic analysis in this report is to analyze Melbourne people’s sentiment with fast food, using the capabilities of Map/Reduce provided by CouchDB to aggregate tweets with a list of fast food keywords.

7.2.1 Pro-Processing

Before implementing sentiment analysis, we choose to pre-rocessing the tweet text first. Regular expressions were used in this stage to remove all the links(e.g. http://), hashtags(e.g. #Australia), emoticons(e.g. #d506) and all non English alphabet.



7.2.2 Use third library for sentiment analysis