Summarization of Football Event using Tweets

Sandeep Kasa
T.V Ravi Teja

Outline

- Introduction
- Problem
- Project Objectives
- Procedure
- Output and Evaluation
- Plan for future work
- Conclusions
- References

Introduction

- For Events that have “structure” or are long-running, users are likely to want a summary of all occurrences so far .
- In response to searches for ongoing events, today’s major search engines simply find tweets that match the query terms, and present the most recent ones .
- For structured events like Football , it is better to use more sophisticated techniques to summarize the relevant tweets.
- We formalize the problem of summarizing Football event tweets and work on a solution.

Problem

- Considering the game of Football , just returning the most recent tweets about the game is problematic for two reasons:
  - The most recent tweets could be repeating the same information about the event (say, the most recent “assist”)
  - Most users would be interested in a summary of the occurrences in the game so far.

Our Goal : To extract a few relevant tweets that best describe the chain of Interesting occurrences in that match .

Project Objectives

- Scan Twitter Feeds using hash-tags about various Football matches during their occurrences and build a tweet corpus.
- The important moments within an event are detected by searching for extreme changes in update volume(spikes) on a per minute basis.
- Several noise reduction algorithms are used to eliminate spam and off-topic status updates.
- Extraction of relevant tweets from a set of tweets(segment) is done by associating with each tweet a vector of the TF-log(IDF) .
- Sentences from relevant tweets from each cluster are ranked using the phrase graph and the top ‘n’ sentences are returned which form the summary.

Procedure

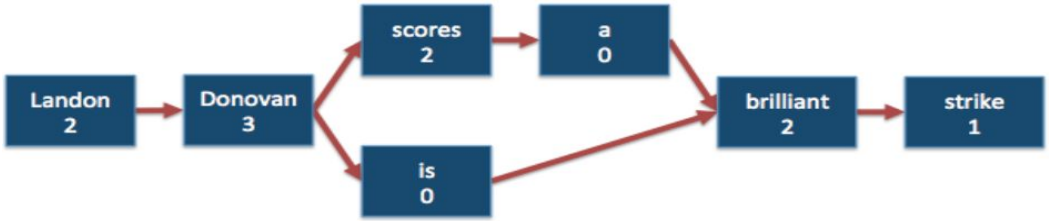
- Identified a significant need of the problem, and understood it as fully as possible in the context in which it occurs.
- Collected the tweets for three to four football matches by using the streaming API of twitter with the help of hash-tags.
- Noise elimination is done on the streamed datasets to remove the irrelevant tweets.
- Absolute number of tweets on every streamed dataset(with noise and without noise) is plotted over the duration of the match.
- Built a new algorithm that is not limited by the availability of multiple similar events.

Procedure

- Sudden increases, or “spikes,” (from generated plots) in the volume of tweets in the stream suggest that something important just happened because many people found the need to comment on it.
- Computes a threshold for the entire event from basic statistics of the set of all slopes for that event from the plot. After trying several formulas , “3*median” as threshold is closely matched to the spikes.
- After identifying all slopes that exceed the threshold, we generate a list of “spikes” (<start time> , <peak time> , <end time>) that correspond to the important moments in the events.
- Extraction of relevant tweets from a set of tweets is done by associating with each tweet a vector of the TF-log(IDF) . We select those tweets which are closest to all other tweets from the event.

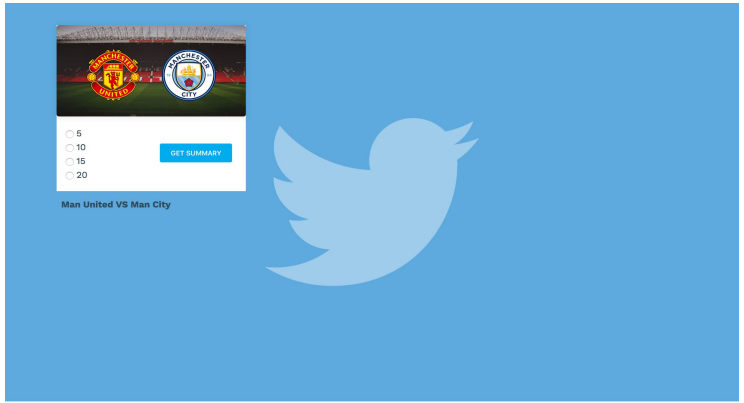
Procedure

- Find the N sentences contained in our set of tweets that best summarize the set. Our approach is to construct a phrase graph from the longest sentence in each status (tweet) , weight the graph according to frequency of words and a few other heuristics, and then to score the longest sentence using the phrase graph.



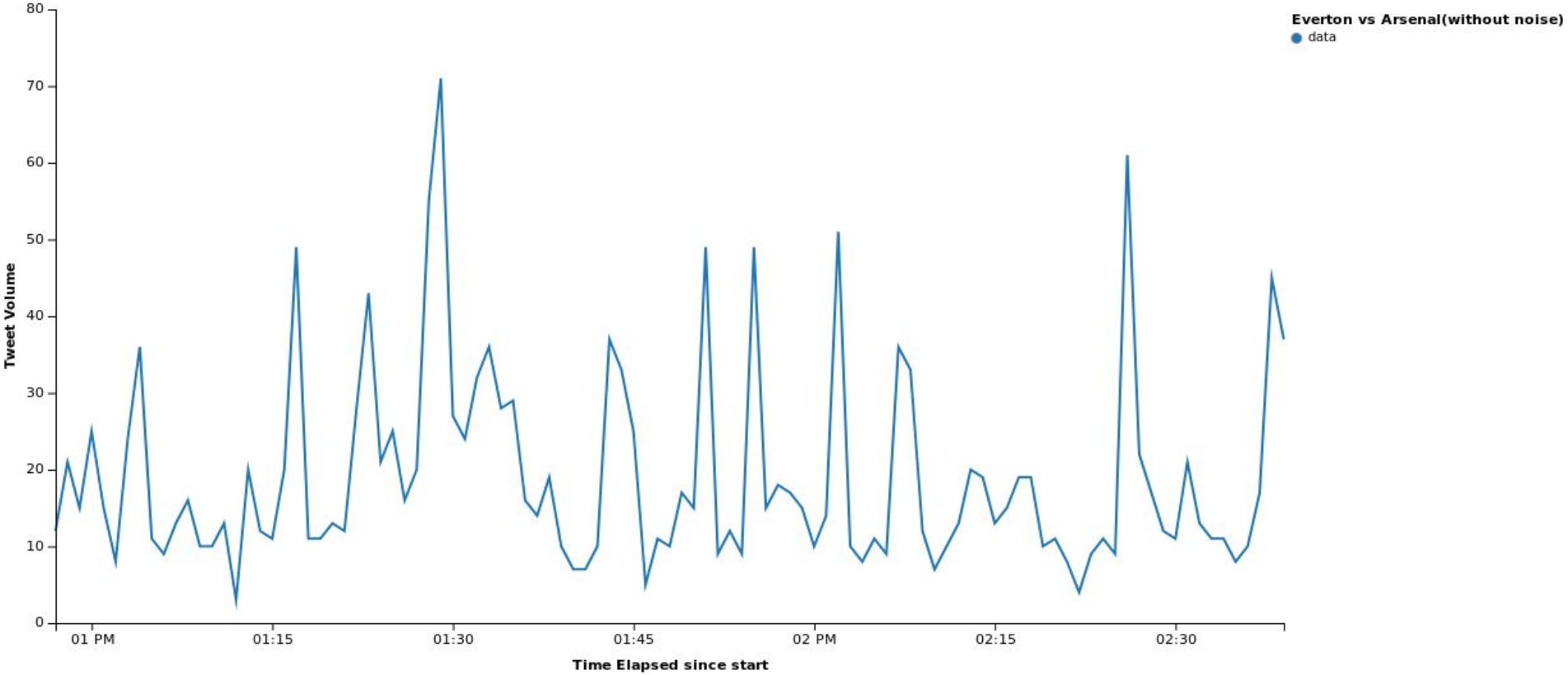
Procedure

- Created a user interface for the offline model which displays various football matches .
- Summary is obtained by selecting the desired match.

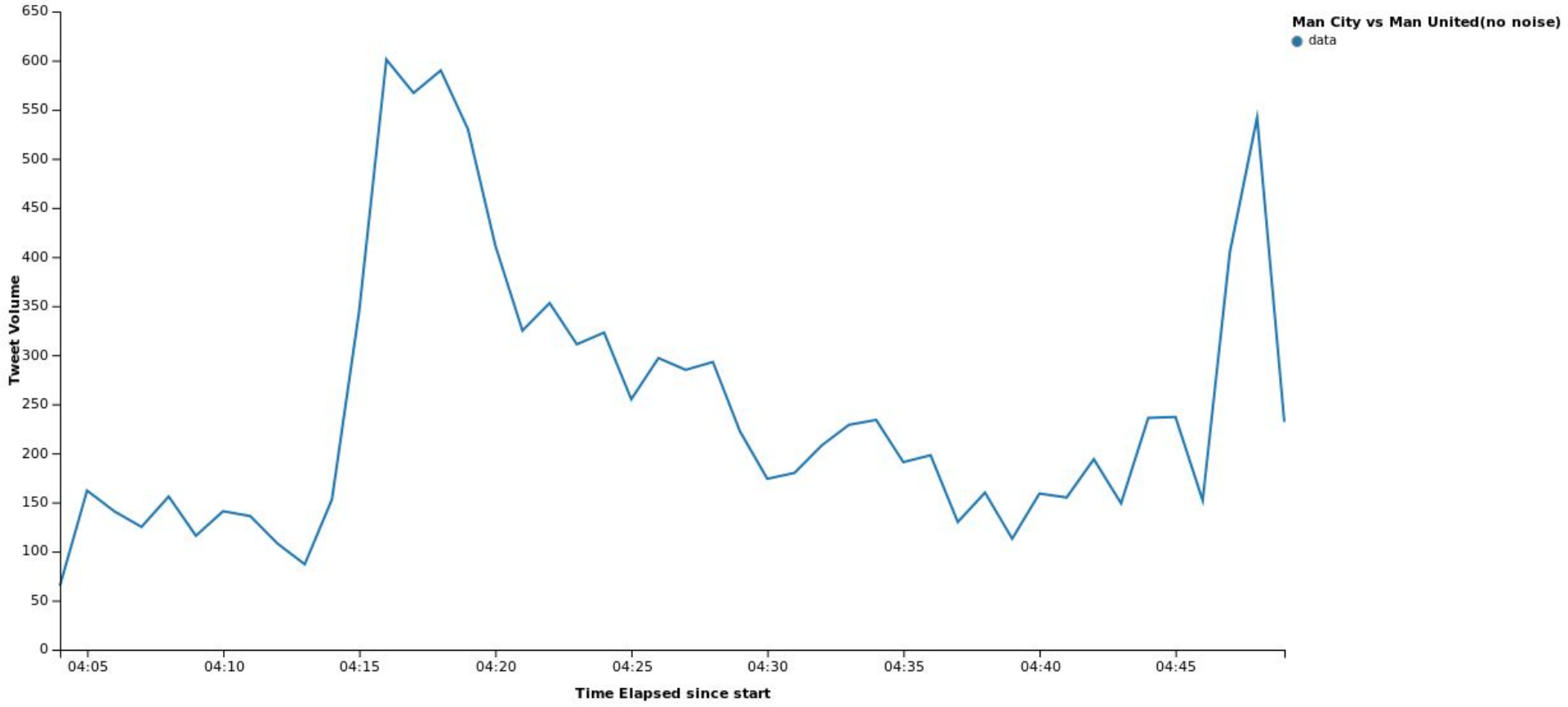


- Studied few Twitter problems in the real world and tried to relate various approaches to solve them.

Output and Evaluation



Output and Evaluation



Output and Evaluation

Time Intervals of detected sub events in Arsenal vs Everton match with the threshold calculated to 15 (using the heuristic mentioned)

```
sandy@SandyPC:~/BTP/Code$ python event.py

VALUE OF THE THRESHOLD :  15
 -----

                        Peak Time    Start Time   End Time

Sub-event 00 -   2016-03-19 13:03:00    13:02:00    13:04:00
Sub-event 01 -   2016-03-19 13:13:00    13:12:00    13:14:00
Sub-event 02 -   2016-03-19 13:17:00    13:16:00    13:18:00
Sub-event 03 -   2016-03-19 13:22:00    13:21:00    13:23:00
Sub-event 04 -   2016-03-19 13:23:00    13:22:00    13:24:00
Sub-event 05 -   2016-03-19 13:28:00    13:27:00    13:29:00
Sub-event 06 -   2016-03-19 13:29:00    13:28:00    13:30:00
Sub-event 07 -   2016-03-19 13:43:00    13:42:00    13:44:00
Sub-event 08 -   2016-03-19 13:51:00    13:50:00    13:52:00
Sub-event 09 -   2016-03-19 13:55:00    13:54:00    13:56:00
Sub-event 10 -   2016-03-19 14:02:00    14:01:00    14:03:00
Sub-event 11 -   2016-03-19 14:07:00    14:06:00    14:08:00
Sub-event 12 -   2016-03-19 14:26:00    14:25:00    14:27:00
Sub-event 13 -   2016-03-19 14:38:00    14:37:00    14:39:00
sandy@SandyPC:~/BTP/Code$
```

Output and Evaluation

Time Intervals of detected sub events in Manu vs ManCity match with the threshold calculated to 72 (using the heuristic mentioned)

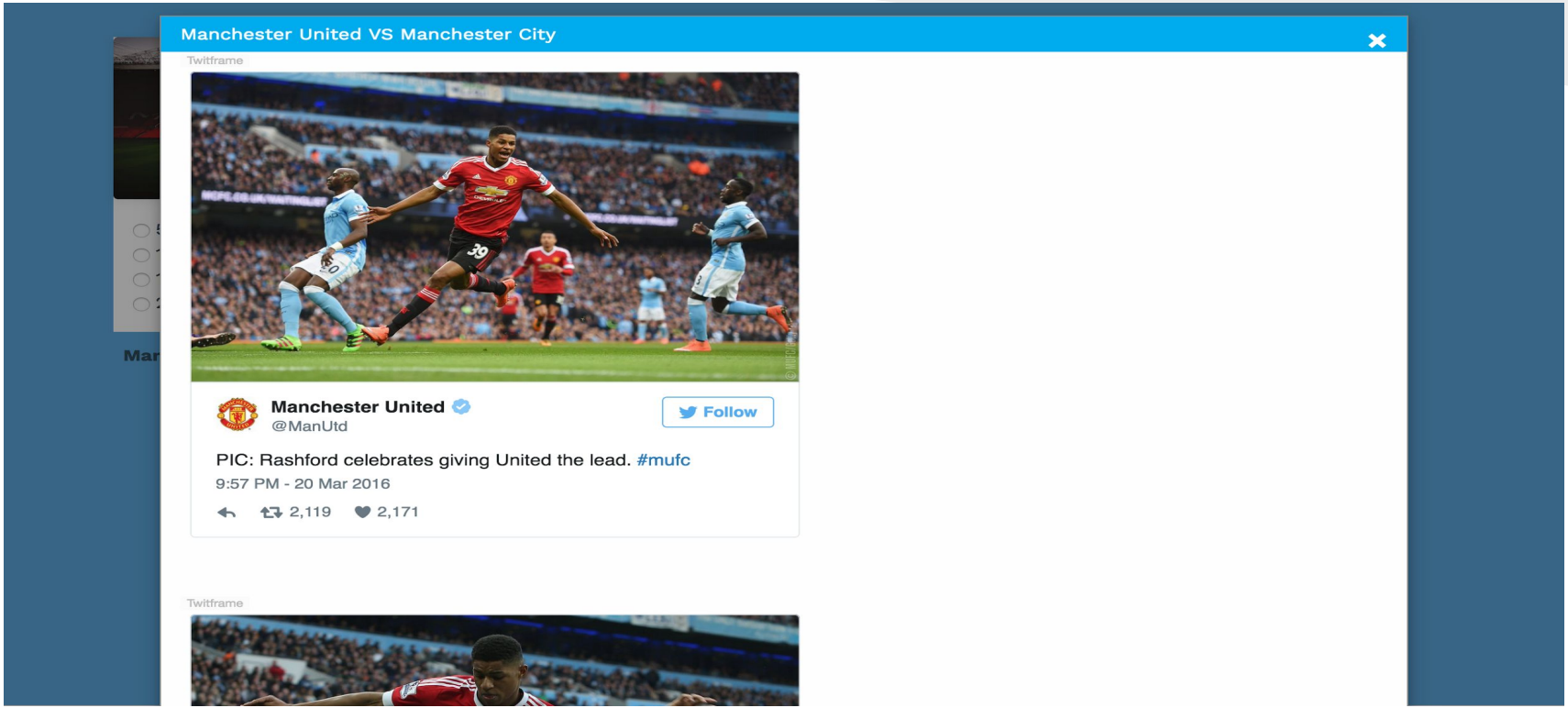
```
sandy@SandyPC:~/BTP/Code$ python event.py

VALUE OF THE THRESHOLD :  72
 -----

                        Peak Time    Start Time   End Time

Sub-event 00 -   2016-03-20 16:05:00    16:04:00    16:06:00
Sub-event 01 -   2016-03-20 16:15:00    16:14:00    16:16:00
Sub-event 02 -   2016-03-20 16:44:00    16:43:00    16:45:00
Sub-event 03 -   2016-03-20 16:47:00    16:46:00    16:48:00
Sub-event 04 -   2016-03-20 16:48:00    16:47:00    16:49:00
sandy@SandyPC:~/BTP/Code$
```

Output and Evaluation

- A subset of tweets from the Arsenal vs Everton tweet corpus is taken as the input and the algo for finding relevant ‘n’ tweets is applied. The ‘n’ taken in this case is 5.
- Output tweets on this subset:

```
sandy@SandyPC:~/BTP/Code$ python tfidf.py ./Datasets/arsenalvseverton.json
The Score of each tweet

[0.0, 0.0, 0.1369386631187064, 0.2191843675107441, 0.24383464928723386, 0.25972057345242006, 0.26373510315172727, 0.2913054205771068, 0.34273195303143
1, 0.35453342017034795, 0.36011613729364406, 0.45505161831430124, 0.4771141907250938, 0.4809250089583208, 0.5698057805491303, 0.6353355088969441, 0.73
38470798077473, 1.0126792266109623, 1.0179865973001867, 1.2234807397355747]


RT @guardian_sport: Goal! Everton 0-1 Arsenal (Welbeck) https://t.co/ywS2YwNmxI #efc #afc

RT @BBCSport: GOAL!   What a goal this is. Great passing and Danny Welbeck skips past the keeper.   #EFC 0-1 #AFC.  https://t.co/tkzoFklSrn

RT @Football__Tweet: GOAL: Everton 0 - 1 Arsenal. Classic Gunners goal. Danny Welbeck rounds the keeper and opens the scoring. Class! #EFC

RT @BBCMOTD: We make it 13 passes in the build-up to the #AFC goal.   #EFC sliced open. 0-1.  https://t.co/ZHQZH1luug  #EFCvAFC https://t.c

RT @Football__Tweet: WOODWORK: Everton hit the beans on toast early doors. #EFC #AFC
```

Output and Evaluation



Output and Evaluation

Manchester United Follow
@ManUtd_ID

FOTO: Rashford merayakan gol usai membawa United unggul. #mufc
9:58 PM - 20 Mar 2016
190 53

Manchester United Follow
@ManUtd_ID

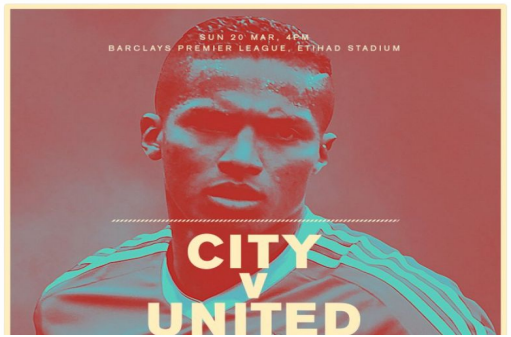
1' - Laga derby Manchester ke-171 telah dimulai. Come on, United. #mufc
9:30 PM - 20 Mar 2016
136 32

Twitframe

Manchester United Follow
@ManUtd

Warm-ups complete. We're 10 minutes away from kick-off. #mufc
9:19 PM - 20 Mar 2016

Twitframe



Some Twitter Problems

- Discovering Similar Users and Grouping them on Twitter
- Localized twitter opinion mining using sentiment analysis
- Click-through prediction for advertising in Twitter timeline

Extensions

- One can examine this heuristic(of selecting the right threshold) to see how it applies to other types of sporting events not in our data set.
- An online version of our algorithm for sub event detection.

Conclusions

- Plotting of the absolute number of tweets over the duration of the match is done . This would help in detecting the intervals of various sub-events over the course of the match.
- Segmenting a timeline of an event with generating the occurences of all the sub events is done.
- Extraction of relevant tweets from a set of tweets in a cluster or segment is done.
- Generated the N sentences(on the basis of scores generated using phrase graph) contained in our set of tweets that will best summarize the set.

References

- Reference Paper 1 ： Event summarization using Tweets
- Reference Paper 2 ： Summarizing Sporting Events Using Twitter
- https://apps.twitter.com/
- http://www.slaw.ca/wp-content/uploads/2011/11/map-of-a-tweet-copy.pdf
- http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/
- https://marcobonzanini.com/2015/03/02/mining-twitter-data-with-python-part-1/