A Report

On

EVENT DETECTION AND TRACKING

Submitted by

SHIVANKIT GAIND

(2015A7PS0076P)

For the partial fulfillment of the course

CS F266 - Study Oriented Project

Under the guidance of

Prof. Poonam Goyal

and

Prof. Navneet Goyal

First Semester (2017 – 2018)

Acknowledgement

I would like to take this opportunity to express my sincere gratitude and deepest regards to my teacher and guide **Prof. Poonam Goyal** for giving me an opportunity to work with her as an undergraduate research assistant for an ongoing research project "**Online Streaming Algorithms For Social Media Analytics**" under the sub-domain of "**Event Detection And Tracking From Multiple Social Media Streams**" in **Advanced Data Analytics and Parallel Technologies Lab**.

Her ideas and guidance helped me traverse alternate routes whenever I faced major roadblocks. Without her insight and brain-storming, the project would not have reached its full potential.

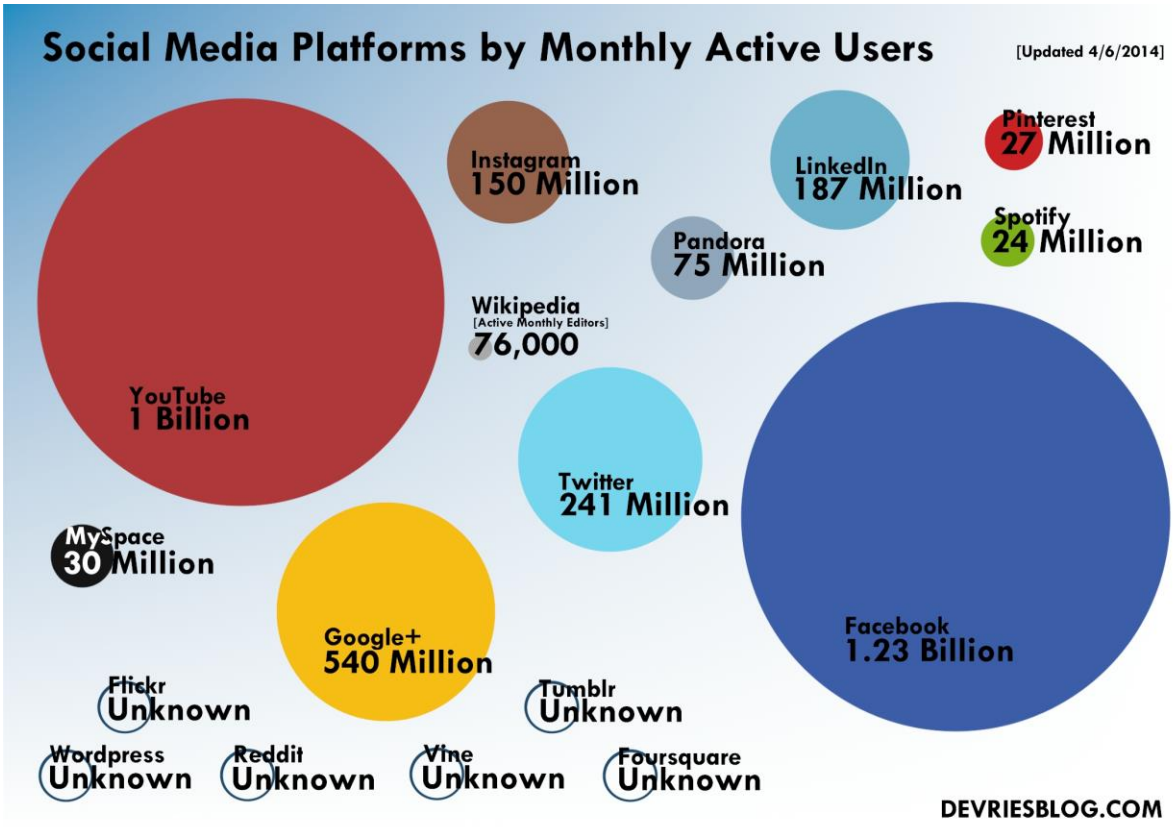
The acknowledgements are incomplete without mentioning my research supervisor, **Mrs. Prerna Kaushik**, who has constantly been working with me on the same project. I would like to thank her for her expert advice, support and encouragement throughout this project.

Table of Contents

Motivation behind the Project

The Web 2.0 era brought a lot of revolutionary changes in the way World Wide Web content is generated and utilized. Social media and online Social Networks are nowadays the most widely used services along with search engines. In this fast-growing Digital World, Social Media has become a part and parcel of everyone's life.

Here is an image showing the number of monthly active users on various social media platforms.



Data generated from Web 2.0 activity are of great value since they reflect aspects of real-world societies. Moreover, data are easily accessible since they can be collected through web-crawlers or public APIs. These two qualities constitute the main motivation for researchers studying online social networks.

Information posted on these social media platforms have been covers everything from daily life stories to the latest local and global news and events, be it in the form of text, images or videos. Monitoring and analyzing this rich and continuous user-generated content can yield unprecedentedly valuable information, enabling users and organizations to acquire actionable knowledge.

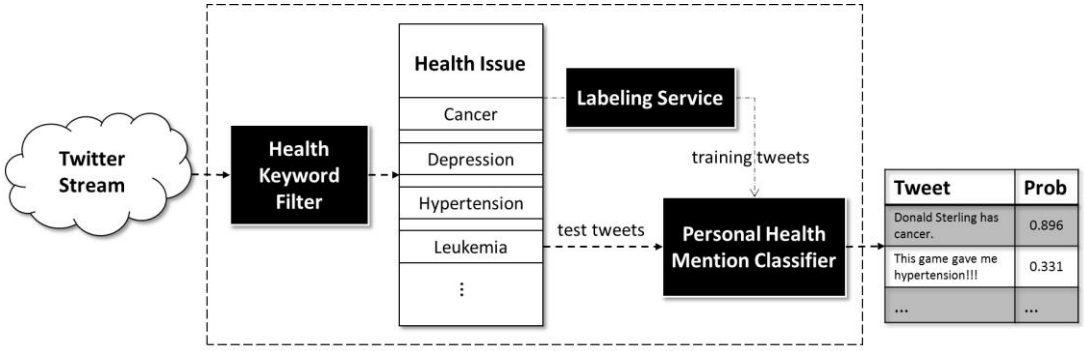
Event detection is a research area that attracted attention during the last years due to the widespread availability of social media data. The problem of event detection has been examined in multiple social media sources like **Twitter, Flickr, YouTube, Instagram and Facebook**. The task comprises many challenges including the processing of large volumes of data and high levels of noise.

This project aims at doing a literature survey of various existing techniques in the domain of event detection and tracking, implementing few of them and innovating new techniques for improving performance. Before diving into the field, let's have a look at various applications of event detection on social media and how these applications have brought a revolution in the ways people are informed about the happenings all around the world.

Applications of Event Detection

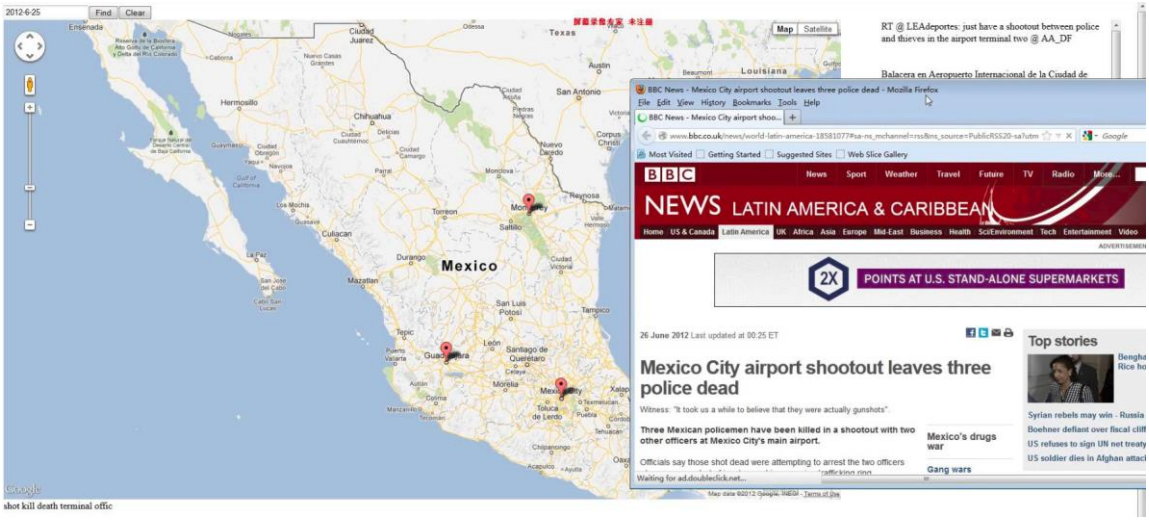
This section includes a set of interesting applications of event detection systems and methods. The applications range from generic global events to celebrity specific incidents.

- Twitter was used to identify tweets that are about **health issues**. A study investigated what types of links the users consult for publishing health related information. A similar application was developed where authors collected tweets about Influenzas and identify flu outbreaks. Their results are similar to Google-trends based flu outbreak detection especially in the early stages of the outbreak. It is easy to see the potential social impact of such applications.



- Techniques are being designed for **identifying earthquake incidents** with Twitter users as sensors. Efforts are made to detect the location and the trajectory of the phenomenon. There are systems which monitor Twitter and emails citizens whenever an earthquake is detected. The response time of the system is proved to be quite fast, similar to the Japan Meteorological Agency. Not only that, there are systems to detect flood events in Germany providing visual information on the map.

- The TEDAS system targets **Crime and Disaster incidents** by identifying where and when they happened. A map visualization of tweets is available. Flickr and YouTube are also utilized where the goal is to detect content related to an emergency. The above systems help the authorities in detecting real-time incidents as well as in extracting useful information after the event.



- In the area of **sports analytics**, an application named EvenTweet system could detect the start time and the location of football matches for UEFA 2012. The system was able to detect National Football League events of the 2010–2011 season. The events include touchdowns, interceptions and goals.

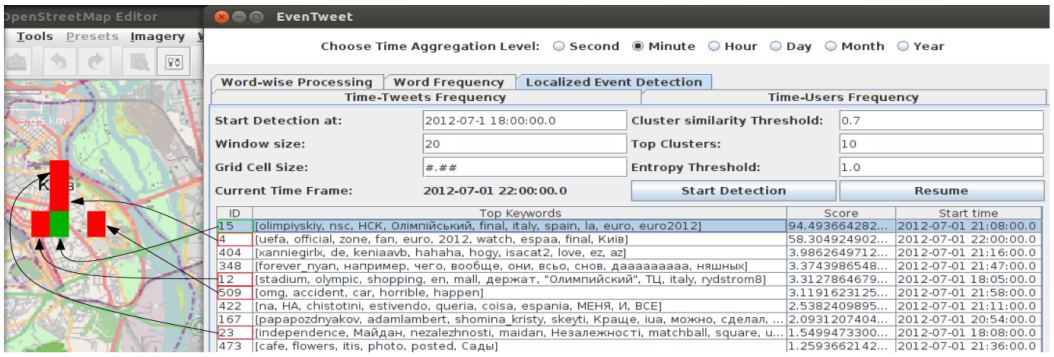


Figure 2: EvenTweet's interface to visualize detected localized events. A user can enter values for the different parameters and select a timestamp at which the real-time detection starts. Localized events, along with other associated information, are listed in descending order by their scores and updated once a new time frame elapses. Events selected by the user will be shown as a green cell on the map.

- An Android application is developed that finds local events given a specific geographic area. The application is able to provide **summaries** to the users. The Jasmine system detects local events for the user according to the desired size of event and the number of users attending it. It presents summaries and some important tweets per event in order to provide with a short description. This group of applications could support mobile users looking for **"happenings" nearby**.

Research Challenges and Requirements

There are numerous research challenges inherent in event detection. In this section we discuss the ones that differentiate this task from other well-known problems. Hence, we justify why off-the shelf data and text mining approaches are not suitable for tackling event detection.

Volume and Velocity

Data from social media come in great volume and velocity. Therefore, algorithms should be online and scalable in memory and computational resources. High data volume makes batch processing computationally infeasible. So, most of the related work aim in building online systems capable of processing high rate streams such as the Twitter Sample stream (1 %) or even the Firehose stream (100 %).

Real-Time Event Detection

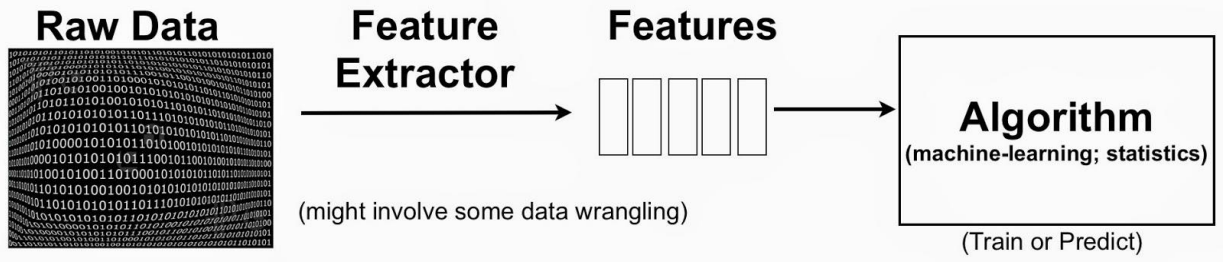
Events should be identified as soon as possible, especially when the approach is intended to be used in critical applications like emergency response. In this case, methods for event detection should be evaluated not only in terms of Precision and Recall but also in terms of how fast they can identify a specific type of event.

Feature Engineering

Selecting the most suitable features to utilize in supervised or unsupervised learning components is not a trivial task. Textual representations such as Term-Document matrices are not sufficient. As many researchers have observed, there are specific characteristics that appear in event related messages. These features could be content-based attributes such as TF-IDF scores, number of tags and emoticons or structural features like the number of followers (Twitter) or friends (Facebook). Hence, it is obvious that the utilization of the correct feature-set is very crucial for the event detection process.

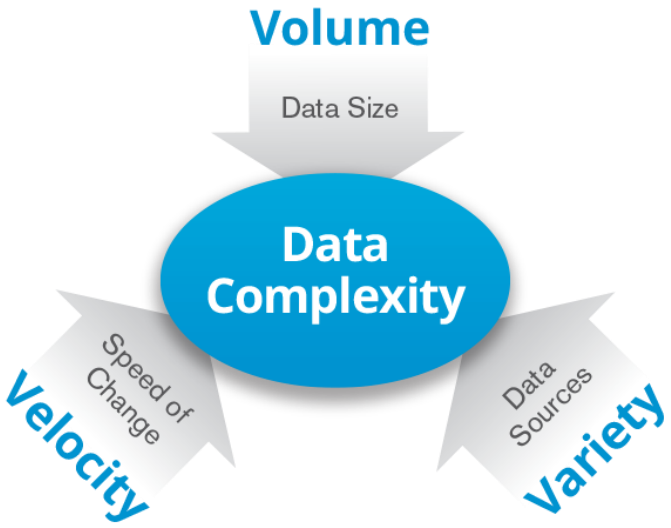
The first step towards overcoming these challenges would be gaining a deep understanding of the already existing techniques and solutions to the above problems in the same domain. Hence, the next section gives a summary of the

literature survey performed by me which is followed by the details of the research paper implemented as one of the major tasks assigned this semester.



Noise and Veracity

It is only natural that user generated information is characterized by noise. Social media are filled with spam messages, advertisements, bot accounts that publish large volumes of messages, hoaxes, as well as internet memes. Another obstacle is that textual information in social media is very limited. Users usually publish very short messages a fact that makes off-the-shelf Text Mining and NLP methods unsuitable.



Literature Survey

A major part of the project involves Literature Survey. Analyzing the tremendous amount of work done in this particular domain helped me to:

- gain an impression about the important aspects of the event detection
- identify data sources that other researchers have used;
- identify and become familiar with the style of writing that is used -
- identify the relationship between concepts;
- identify ideas for further consideration;
- see how I can prevent myself from repeating any errors that have been identified in previous work;
- creating my own reading and critiquing strategy.

Following is the summary of various features and techniques being exploited in the domain of event detection and tracking:

TABLE 1. Taxonomy of Event Detection Techniques in Twitter.

References	Type of event		Detection method		Detection task		Application
	Specified	Unspecified	Supervised	Unsupervised	NED	RED	
Sankaranarayanan et al. (2009)		x	x	x	x		Breaking-news detection
Phuvipadawat and Murata (2010)		x		x	x		Breaking-news detection
Petrović et al. (2010)		x		x	x		General (unknown) event detection
Becker et al. (2011a)		x	x	x	x		General (unknown) event detection
Long et al. (2011)		x		x	x		General (unknown) event detection
Weng and Lee (2011)		x		x	x		General (unknown) event detection
Cordeiro (2012)		x		x	x		General (unknown) event detection
Popescu and Pennacchiotti (2010)	x		x		x		Controversial news events about celebrities
Popescu et al. (2011)	x		x		x		Controversial news events about celebrities
Benson et al. (2011)	x		x			x	Musical event detection
Lee and Sumiya (2010)	x			x	x		Geosocial event monitoring
Sakaki et al. (2010)	x		x		x		Natural disaster events monitoring
Becker et al. (2011)	x		x			x	Query-based event retrieval
Massoudi et al. (2011)	x			x		x	Query-based event retrieval
Metzler et al. (2012)	x			x		x	Query-based structured event retrieval
Gu et al. (2011)	x			x		x	Query-based structured event retrieval

References	Detection techniques	General features	Twitter-specific features
Sankaranarayanan et al. (2009)	Naive Bayes classifier and online clustering	Term vector	Hashtags and timestamps
Phuvipadawat and Murata (2010)	Online clustering	Term vector, proper nouns (conventional NER)	Hashtags, #followers, #retweets and timestamps
Petrović et al. (2010)	Online clustering (based on locality sensitive hashing)	#tweets, #users and entropy of messages	–
Becker et al. (2011a)	Online clustering and support vector machine classifier	Term vector	Hashtags, multi-word hashtags with special capitalization, retweets, replies and mentions.
Long et al. (2011)	Hierarchical divisive clustering	Word frequency and entropy	Probability of word occurring in hashtags
Weng and Lee (2011)	Discrete wavelet analysis and graph partitioning	Individual words	–
Cordeiro (2012)	Continuous wavelet analysis and latent Dirichlet allocation	–	Hashtag occurrences
Popescu and Pennacchiotti (2010)	Gradient boosted decision trees	Correlation of target events (or entities) with the Web and traditional news media	Proportion of nouns, verbs, questions, bad words, etc.; #tweet, #retweets, #replies, #tweets per user, hashtags; proportion of tweets and hashtags involving buzziness, sentiment, controversy

References	Detection techniques	General features	Twitter-specific features
Popescu et al. (2011)	Gradient boosted decision trees	Part-of-Speech tagging and regular expressions (in addition to the features used by Popescu et al. (2011))	Relative positional information, length of snapshot, category, language (in addition to the features used by Popescu et al. (2011))
Benson et al. (2011)	Factor graph model and conditional random fields	Term vectors for artist names (extracted from Wikipedia) and for city venue names.	Word shape, patterns for emoticons, time references, venue types
Lee and Sumiya (2010)	Statistical modeling of normal crowd behavior	–	#Tweet, #Crowd, #MovingCrowd based on geotags
Sakaki et al. (2010)	Support vector machine classifier	–	#Words, #keywords and the words surrounding users query
Becker et al. (2011)	Recursive query construction	Term frequency and co-location	Hashtags and URL
Massoudi et al. (2011)	Generative language modeling		Emoticons, post length, shouting, hyperlinks, capitalization, recency, #reposts and #followers
Metzler et al. (2012)	Temporal query expansion technique		Burstiness score based on the frequency of query term occurrence
Gu et al. (2011)	Event modeling (ETree)	Term vector and n-gram models	Replies to tweets

Event Types: In the literature we come across the following types of events:

- **Planned**: Events with a predefined time and location (e.g. a concert).

- **Unplanned**: Events that are not planned and could happen suddenly (e.g. a strike, an earthquake).
- **Breaking News**: Events connected to breaking news that are discussed in conventional news media (e.g. the result of the elections in Greece discussed by the global press).
- **Local**: Events limited to a specific geographical location. The event impacts only this area (e.g. a minor car accident).
- **Entity Related**: Events about an entity (i.e. a new video clip of a popular singer).

The table below summarizes the range of the different event types in terms of space and time. It also reports in which media these events are more probable to be observed in.

Event type	Time duration restrictions	Geographical distribution	Observable in
Planned	High	Medium	Social media, news media, event portals
Unplanned	Low	High	News media
Breaking news	High	Low	News media
Global	Low	Low	News media, online sources
Local	High	High	Local media, online sources
Entity	High	Low	News media, blogs

Now, after the literature survey, as a part of a practical implementation experience, I was assigned the task of implementing a recent research paper based on event detection from two Social Media Streams – Twitter and Instagram. The coming section gives the details of the research paper whose implementation has been completed this semester in JAVA (for better performance evaluation).

Social Fusion: Integrating Twitter and Instagram for Event Monitoring

Introduction

This paper develops an algorithm to identify and geo-locate real world events that may be present as social activity signals in two different social networks. Specifically, the focus is on the content shared by users on Twitter and Instagram in order to design a system capable of fusing data across multiple networks.

The two networks have complementary advantages. Twitter data are more prolific leading to detection of more events, but it is also more noisy, generating more false positives. In contrast, Instagram data feeds are sparser, but events detected based on Instagram data tend to include fewer false positives.

Hence, fusing the two together, can offer a solution that features the benefits of both; the results have a much smaller fraction of false positives compared to using Twitter alone, and have more events detected, compared to Instagram.

Finding Potentially Related Posts

- To find which Instagram posts are potentially related to which Twitter posts we need a logical distance metric between an Instagram post and a Twitter posts.
- A convenience metric is the location referred to in the post. However, most tweets do not mention location.
- Thus, keywords also need to be considered. Instagram posts contain image tags (we call hash-tags).
- Therefore, there is a need to identify whether words contained in a tweet are related to these hash-tags or not.
- In this paper, a “quick and dirty” approach is chosen that rely on string matching, but does not consider semantics.

String Matching Algorithm

- To reduce the noise, we first do some pre-processing on tweet text by removing the English stop words, special characters (non alphanumeric), and web links.
- We also do not consider the query keyword as it will be present in all the Instagram/Twitter posts by default.
- It is also important to note that the hash-tags are sometimes composed of multiple words, merged together.
- In order to overcome this issue, we use the processed tweet text and remove all the white spaces to form a single string.
- Next, we determine the number of hash-tags from the Instagram post that are present as substring within the modified tweet string. This metric known as tag similarity is defined as below:

$$tag\_sim = \frac{\#\ of\ tags\ present\ as\ substring\ in\ tweet\ string}{\#\ of\ tags}$$

Maximum Likelihood Estimation Algorithm

- We emphasize that these are potentially relevant tweets but we do not yet know, based on the above distance metric alone, if they are truly relevant and not (i.e., only accidentally similar).
- A contribution of the work is to offer a maximum likelihood estimate of actual relevance.
- The maximum likelihood estimation algorithm leads to the discovery of three separate quantities: (i) whether an Instagram location is an actual event location or not, (ii) for a given Instagram event location, what are the significant tags and the corresponding relevant tweets (tweet event clusters) corroborating the observation, and (iii) what is the exact geo-coordinate (location) where the event happened.

- An unsupervised method is proposed in which we assume that we have no prior knowledge of the significance of the Instagram tags as well as no prior knowledge of the relevance of the retrieved tweets using the above similarity metric.

Final Algorithm

- Given an Instagram cluster containing a set of hash-tags and location information we first retrieve the tweets based on the tag similarity metric.
- We then initialize the value of the parameters to some random values.
- The algorithm then performs the E-steps and M-steps iteratively until θ converges.
- Specifically, at every E-step we try to determine the probability value of a tweet Tj being relevant as assign it to R(t,j).
- Based on this probability value we next perform M-step where we identify the optimal value of all the parameters as described in our derivation.
- After the convergence we get a ranked list of tweets based on the R(t,j) values. Alternatively we can also assign a binary value to the tweets based on the condition R = 1 if $R(t,j) \geq 0.5$ or R = 0 otherwise.

Implementation

All the modules required to implement the event detection and tracking procedure described in the paper "**Social Fusion: Integrating Twitter and Instagram for Event Monitoring"** have been coded.

As of now, sample tweets for testing the modules are collected from **Twitter4J** (an open-sourced, mavenized and Google App Engine safe Java library for the Twitter API which is released under the Apache License 2.0) whereas the sample Instagram posts are obtained from **Instagram4J (**a Java driver/wrapper for the Instagram API).

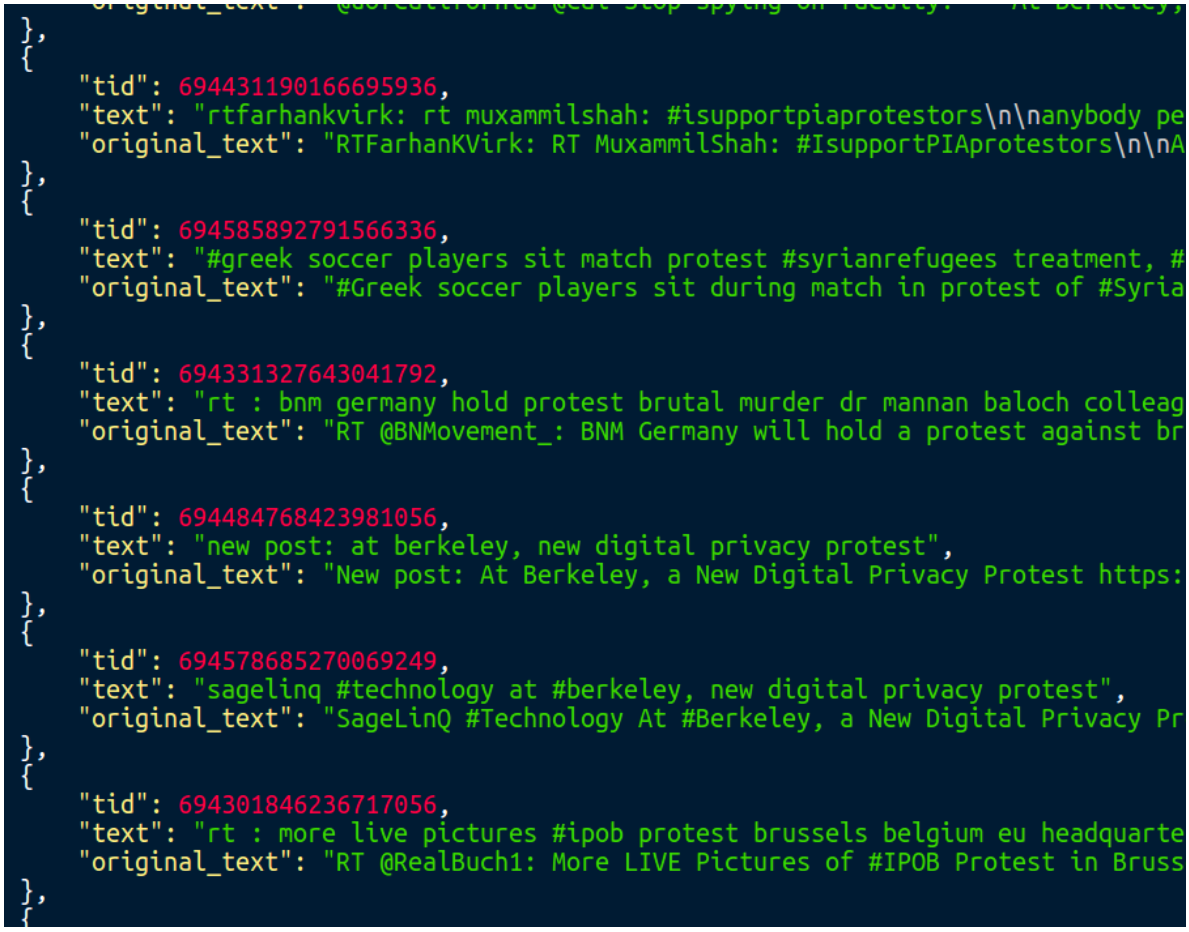
The entire code is written in Java for better performance evaluation.

Here is a brief explanation of some important modules in the code (Please refer to the code at Social_Fusion for further understanding):

- **get_cosine**: This module returns the cosine similarity score of two sentences.
- **text_to_vector**: This module returns the term frequency vector of a sentence.
- **get_cosine_tweets**: This module returns tweets matching to a particular tag sentence obtained from instagram tags on the basis of cosine similarity.
- **get_tag_similarity_tweets**: This module returns tweets matching to a particular tag sentence obtained from instagram tags on the basis of tag similarity metric as described in the paper.
- **get_loc_names**: This module returns the location names at different levels of geography (ranging from street name to country name) for a given Latitude-Longitude pair.
- **findCentroid**: This module calculates the centroid for the tweets and then assigns cosine similarity scores to tweets from the centroid.
- **para_est**: This module performs the job of parameter estimation and return back two parameters p1 and p2.
- **perform_EM**: This is the main module performing the EM step.
- **process_clusters**: This module performs the job of processing clusters.

- **IsLocPresent**: This module checks whether the tweet is geotagged or not.
- **store_locations_into_file**: This module stores the geo-locations object in a file to be used later so that repeated calls to API can be avoided every time the code is run.
- **getInstaEvents**: This module returns the JSON array of objects for the instagram events.
- **getTweets**: This module returns the JSON object for the Tweets.

Here is a screenshot of the sample tweet data along with preprocessed tweets.



Here is a screenshot of the sample Instagram events data represented by the image tags and the geolocation.

```
{
   "tags":[
      "Executions",
      "Rouhani",
      "Irans",
      "Stage",
      "protest",
      "Paris",
      "News"
   ],
   "lat":35.715298,
   "long":51.404343
},

{
   "tags":[
      "Digital",
      "protest",
      "Privacy",
      "Berkeley"
   ],
   "lat": 37.871853,
   "long":-122.258423
},
```


Finally, here is a screenshot of a fused bucket, an event whose trace was found both on Instagram and Twitter.


```
{
    "lat": 37.871853,
    "long": -122.258423,
    "tags": [
        "Digital",
        "protest",
        "Privacy",
        "Berkeley"
    ],
    "tweets": [[
        {
            "Modified": "\"at berkeley, new digital privacy protest\" steve lohr via nyt",
            "Original": "\"At Berkeley, a New Digital Privacy Protest\" by STEVE LOHR via NYT https://t.co/q8VBMiUuyI
            "tid": 694485035181850624
        },
        {
            "Modified": "\"at berkeley, new digital privacy protest\" steve lohr via nyt",
            "Original": "\"At Berkeley, a New Digital Privacy Protest\" by STEVE LOHR via NYT https://t.co/oQet47NuiW
            "tid": 694335510869884928
        },
        {
            "Modified": "at berkeley, new digital privacy protest - new york times",
            "Original": "At Berkeley, a New Digital Privacy Protest - New York Times https://t.co/QRSOvA17Tt",
            "tid": 694512786924134402
        },
        {
            "Modified": "new email: at berkeley, new digital privacy protest - from: action.com",
            "Original": "New email: At Berkeley, a New Digital Privacy Protest - from: action@ifttt.com",
            "tid": 694397926622285824
        },
        {
            "Modified": "\"at berkeley, new digital privacy protest\" check via nyt",
            "Original": "\"At Berkeley, a New Digital Privacy Protest\" Check out via NYT https://t.co/DyiWooIMMY The
            "tid": 694397643158585344
        },
```


Results (W.R.T. Original Results)

Recall: 99.8% (Fraction of the tweets that should have been in bucket)

Precision: 100% (No irrelevant tweets are added to a particular bucket)

Exploring APIs

Implementing the above paper helped me getting an idea of:

- What are the various API's offered by different social media platforms.
- How to extract real time data from these API's
- How to preprocess data obtained from these platforms (for example Tweets cleaning)
- What kind of Meta-Data is associated with images on platforms like Instagram
- How to find suitable features from the ones offered by the API's for better performance
- How to find correlation between data obtained from multiple social media platforms
- What are the limitations of data extraction from different API's (for example the number of calls per minute)

Since our project aims on extracting information from **Multiple Social Media Streams**, it would be a wise choice to work with streams that offer different kind of data parallelly (for example – Tweets (text) from Twitter and Images from Instagram) for better social fusion. Instagram has put restrictions on offering its public data and the facility of keyword-based search extraction of image meta-data since 2016.

Hence, I explored different API's from different social media platforms as a part of this project and as a first step towards creating a dataset on which our future research will be carried on after proper labelling.

And the exploration revealed that Flickr API is pretty good in offering various services to the researchers and developers.

Here is a brief description about the same and how to use it as well.

Steps to Use Flickr API

The Flickr API is a powerful way to interact with Flickr accounts. With the API, you can read almost all the data associated with pictures and sets. You can also upload pictures through the API and change/add picture information.

The first thing you need is an **API key**. The API key is a way for Yahoo! to track the activity associated with each API key. It is essentially your user-name for the Flickr API. You can get an API key fairly easily depending on the type of project you are doing.

- If you're producing a **commercial product**, then Yahoo! will want to know about it. You'll need a special API key for that type of project and Yahoo! will have to approve your product.
- However, if you're just experimenting and producing a **non-commercial product**, you can get an API key instantly.

There are essentially **three steps** to working with the API.

- First, you need to **send Flickr a request** of the information you would like. This is done by building a special URL (more on this later).
- Second, once Flickr understands your correctly built URL, it will **send the information** that you requested.
- The last step is to **do something with the data** with which Flickr responded. Whatever you're doing with the Flickr API, your interaction will follow these three steps.

Future Work

Literature Survey, API Exploration and Implementing the mentioned research paper constitute the initial phase of the project. All this was an introduction to the vast domain of Information Extraction from Multiple Social Media Platforms.

1. The next step would be to create a dataset using certain features parallelly from multiple social media streams such as Twitter, Flickr or Pinterest. The data would be crawled for a period of one complete month or even more.
2. It will be followed by another phase which includes labelling the self-created dataset on which our future research will be based. This will form the ground truth for evaluating the new techniques we will be working next semester.
3. Hence, what finally follows is the Design Phase, where using the knowledge of the existing solutions, techniques and algorithms in this particular domain, research will be carried out for coming up with new solutions which have better performance in terms of either precision, recall, time taken for event detection/tracking or information representation.

References

- P.Giridhar, T.Abdelzaher, and L.Kaplan. Social fusion: Integrating twitter and Instagram for event monitoring. In UIUC tech report,2017.
- S.Wang ,P.Giridhar, H.Wang, L.Kaplan, T.Pham, A.Yener, and T.Abdelzaher. Storyline: On physical event demultiplexing and tracking in social spaces. In IoTDI, 2017.
- SANKARANARAYANAN, J., H. SAMET, B. E. TEITLER, M. D. LIEBERMAN, and J. SPERLING. 2009. TwitterStand: News in tweets. In Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '09, ACM, New York, NY, pp. 42–51.
- PHUVIPADAWAT, S., and T. MURATA. 2010. Breaking news detection and tracking in Twitter. In IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), Vol. 3, Toronto, ON, pp. 120–123.
- PETROVI´ C, S., M. OSBORNE, and V. LAVRENKO. 2010. Streaming first story detection with application to Twitter. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT '10, pp. 181–189.
- BECKER, H., M. NAAMAN, and L. GRAVANO. 2011a. Beyond trending topics: Real-world event identification on Twitter. In ICWSM, Barcelona, Spain.
- LONG, R.,H.WANG, Y.CHEN, O.JIN, and Y.YU. 2011.Towards effective event detection, tracking and summarization on microblog data. InWeb-Age Information Management, Vol.6897 of Lecture Notes in Computer Science. Edited by H. WANG, S. L I, S. OYAMA, X. H U, and T. QIAN. Springer: Berlin/Heidelberg, pp. 652–663.
- WENG, J., and B.-S. LEE. 2011. Event detection in Twitter. In ICWSM, Barcelona, Spain.
- CORDEIRO, M. 2012. Twitter event detection: Combining wavelet analysis and topic inference summarization. In Doctoral Symposium on Informatics Engineering, DSIE'2012.

- POPESCU, A. M., and M. PENNACCHIOTTI. 2010. Detecting controversial events from Twitter. In Proceedings of the 19th ACM international

Conference on Information and Knowledge Management, CIKM'10, ACM, New York, NY, pp. 1873–1876.

- POPESCU, A. M., M. PENNACCHIOTTI, and D. PARANJPE. 2011. Extracting events and event descriptions from Twitter. In Proceedings of the 20th International Conference Companion on World Wide Web, WWW '11, pp. 105–106.
- BENSON, E., A. HAGHIGHI, and R. BARZILAY. 2011. Event discovery in social media feeds. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Volume 1 of HLT '11, Association for Computational Linguistics, Stroudsburg, PA, pp. 389–398.
- LEE, R., and K. SUMIYA. 2010. Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection. In Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, LBSN '10, ACM, New York, NY, pp. 1–10.
- SAKAKI, T., M. OKAZAKI, and Y. MATSUO. 2010. Earthquake shakes Twitter users: Real-time event detection by social sensors. In Proceedings of the 19th International Conference on World Wide Web, WWW '10, ACM, New York, NY, pp. 851–860.
- BECKER, H., F. CHEN, D. ITER, M. NAAMAN, and L. GRAVANO. 2011. Automatic identification and presentation of Twitter content for planned events. In International AAAI Conference on Weblogs and Social Media, Barcelona, Spain.
- MASSOUDI, K., M. TSAGKIAS, M.DE RIJKE, and W. WEERKAMP. 2011. Incorporating query expansion and quality indicators in searching microblog posts. In Proceedings of the 33rd European Conference on Advances in Information Retrieval, ECIR'11. Springer-Verlag: Berlin, Heidelberg, pp. 362–367.
- METZLER, D., C. CAI, and E. H. HOVY. 2012. Structured event retrieval over microblog archives. In HLTNAACL, pp. 646–655.
- GU, H., X. XIE, Q. L V, Y. RUAN, and L. SHANG. 2011. ETree: Effective and efficient event modeling for realtime online social media networks. In Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference, Vol. 1, pp. 300–307.