An end-to-end framework for event detection in social media

Nguyen Thanh Tam
École Polytechnique Fédérale de Lausanne
tam.nguyenthanh@epfl.ch

Daniel Gatica-Perez
Idiap Research Institute
gatica@idiap.ch

ABSTRACT

With the growth of social media, information sharing on micro-blogging platforms such as Twitter has exploded. This huge knowledge base can be leveraged to extract useful information such as real-world events. The dynamic nature of this corpus can be exploited to not only detect, but also model and track the evolution of events over time. However, event detection is hindered by the sheer amount of available social media data and its unstructured, free-text representation. In this work, we develop an end-to-end framework that facilitates the event detection process by offering text processing and mining operators for analyzing and grouping raw data across different dimensions. The analytic findings focus on a case study in Twitter data. Our framework can be extended to other textual corpora as well as publicly available for further performance evaluation and functionality integration.

Keywords

event detection; Twitter; topic modeling

1. INTRODUCTION

Social media has a huge user base sharing all kinds of information at a very high rate. Information shared on social media range from personal information like what they are eating, to local events like festival celebrations, to events having worldwide impact like forest fires. Since the users of social media are spread all over the world, people usually share information about events almost instantaneously. This fact makes the study of social media data very important in order to model the evolution of important events in real-world.

Event detection in social media has many applications. First, it can be used to predict future outcomes such as using the peaks of tweet sentiment to predict stock price [5]. Second, it can be used to understand a known event by performing the topic modeling [9]. Last but not least, it can be used for early warning of emergency situations [15]. However, along with the diversity and richness of information pervading the Twitter ecosystem comes an equally huge amount of uninteresting, insignificant and noisy information. With this motivation, we wish to explore the problem of detection and tracking of events using social media.

While many event detection techniques have been developed for time series in general and for social media in particular, there is little work on the evaluation of their performance altogether. The main reason is the unstructured nature of human text in social media, which contains different types and levels of textual elements such as term, topic, sentiment, and entity. Moreover, events might not be known before-hand and can be interpreted in different ways (e.g. #bursty keywords, #followers, anomalies in spatial-temporal dimensions). The former calls for automatic text-mining approaches, whereas the latter implies a need for semi-automatic or automatic processing to formulate and extract the events.

In this paper, we develop an end-to-end and unified framework for event detection in social media. We implement the state-of-the-art techniques in text mining as well as event detection. Potential users (e.g. social researchers) can utilize our framework to study and analyze their hypotheses in social media. Moreover, new techniques or new datasets can be plugged into the framework for comprehensive performance evaluations. The remainder of the paper is organized as follows. Section 2 describes the dataset we are using to showcase the framework. Section 3 presents the analytical architecture of our framework pipeline. Section 4 shows some highlighted analyses in the dataset. And Section 5 concludes the paper.

2. DATA

The data contains tweets from Twitter Streaming API with bounding box located in Guanajuato, a tourist city in central Mexico, collected as part of the SenseCityVity project [14, 16]. The descriptive statistics for the dataset are given in Table 1. The geographic distribution of geo-tagged tweets is given in Figure 1. The time series of tweet counts are depicted in fig. 2 over 58 days with average 5773 tweets per day.

Table 1: Dataset characteristics

Statistics	Quantity
#Tweets	334836
Language	es (78.52%), en (7.7%), others (13.78%)
Period	09/04/2014 – 05/06/2014

ACM ISBN .
DOI:

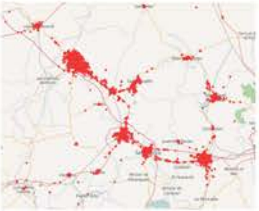


Figure 1: Geographic distribution of dataset

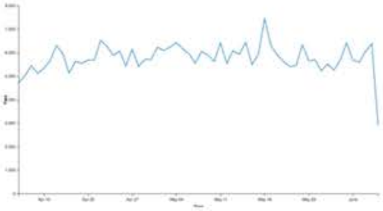


Figure 2: Raw tweet counts for each day

3. METHOD

Figure 3 depicts the complete pipeline of our event detection system. The input of our framework is Twitter data or any textual social media data. The first and foremost step is *Data Preparation*, which pre-processes Twitter data. This step is of paramount importance because Twitter data is extremely noisy and contains a lot of irrelevant and redundant information that have are either not informative and interesting or have no bearing on event extraction process. The processed Twitter data is then fed into the event detection pipeline. One of the most important information of tweets for the purpose of event detection is *Syntactical Analysis*, including analyzing the frequency, co-occurrence, and clustering of hashtags and terms. The terms or hashtags are of prime importance to the *Semantic Analysis* component, which segregates the tweets into topic clusters where each cluster could potentially represent a general or specific topic in real-world. The component further provides entity recognition and sentiment analysis to group the tweets across different dimensions for further analyses. This is followed by timeline based or location based segmentation of tweets in each semantical cluster. To pin-point specific event instances, the tweets are grouped by the analyzed semantical elements (topic, entity, sentiment). Finally event instances are detected by automatic techniques (e.g. moving average for time-based or heatmap density for location-based) and formulated by combining the semantical element and timeline/location from the pipeline. The structure of the pipeline implies an inherent dependency in the event detection process, where the output of each component is the input of the next component. In the following, we discuss each component specifically, involving technical details.

Data Preparation. The pre-processing step involves cleaning the data to get rid of noisy elements. In particular, two operators are provided.

- *Tokenization:* divides the text into tokens, in particular emoticons, HTML tags, user mentions, hashtags, numbers, quotes, words, etc. Regular expressions for tokenization are provided in the appendix.

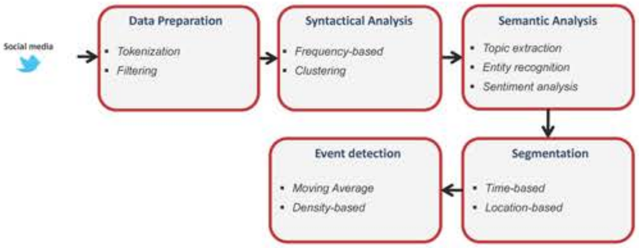


Figure 3: Event detection pipeline

- *Filtering:* filters or retains informative tokens. For example, stop words that occur very frequently such as *a, the, that, etc.* are removed using Python NLTK English corpus because they are not informative. We also take into account punctuation and special phrases ('rt', 'via'). While some simple terms represent a clear topic, more often than not they don't give us a deep explanation of what the text is about; hence, we further provide the extraction of n-grams (i.e. phrase of n tokens) using Python NLTK library.

It is worth noting that the performance of data preparation step depends on the NLP tools and language-specific dictionaries. Obtaining best cleaned results require further benchmarking of those tools, reusing practical guidelines in the NLP literature, or manually removing unexpected terms.

Syntactical Analysis. Assuming we have collected a list of tweets, the first exploratory analysis that we can perform is a simple word count. In this way, we can observe what are the terms most commonly used in the data set. In order to keep track of the frequencies while we are processing the tweets, we can use collections .Counter() which internally is a dictionary (term: count) with some useful methods like most_common(). Similarly, we can compute the frequencies of hashtags and report the top-frequent hashtags in the data.

Sometimes we are interested in the terms that occur together. This is mainly because the context gives us a better insight about the meaning of a term, supporting applications such as word disambiguation or semantic similarity. Technically, to capture the co-occurrences, we build a co-occurrence matrix M_{ij} such that each element contains the number of times the term i has been seen in the same tweet as the term j. This implementation is pretty straightforward, but depending on the size of dataset and on the use of the matrix, one might want to look into tools like *scipy.sparse* for building a sparse matrix. We could also look for a specific term and extract its most frequent co-occurrences without building the co-occurrence matrix by filtering the tweets contain the given term in the *Data Preparation* step and then extract the most frequent terms of the filtered tweets.

Semantic Analysis. The first type of semantic analysis is segregating tweets into topics. Each topic consists of a set of tweets that are related to each other. Each such cluster is intended to correspond to a real-world topic. The intuition is to cluster the tweets into broad sets of related tweets. Each set could potentially contain numerous instances of events that might or might not be directly connected, but could be grouped superficially under the purview of their topic. This analysis further facilitates the event evolution and tracking process. To this end, one could employ topic modeling techniques like LDA [4] which infer topic distributions in documents in an unsupervised manner.

Applying LDA on Twitter data masqueraded as documents gives us logical clusters of related tweets with each clustering representing a topic. The core idea is that LDA considers each tweet as a bag of words and take as input the frequency statistics (e.g. tf-idf) from the previous step. The output is the clusters of co-occurring words. Traditional LDA, however, has some limitations:

- *Sensitive to parameters:* we have to specify the number of topics we want to output from LDA. In order to better model the real-world data robustly, one could apply non-parametric approaches, such as hierarchical LDA [8], which also organizes the topics according to a hierarchy (useful for grouping general or specific topics).
- *Sensitive to short text:* LDA works well on structured documents like news-wire data, articles, and blogs. Due to the noisy, sparse, extremely short, and unstructured nature of micro-blogs, standard LDA fails to infer topic distributions. Several variants have been proposed in literature for the topic inferences on micro-blogs. One could employ Twitter-specific version of LDA [11, 22] for topic based clustering of tweets.

Beside topic extraction, we can also perform further semantic analysis such as entity extraction and sentiment analysis:

- *Entity Extraction:* the idea is to find the important words, places, persons in the text. Note that we can use the @-mention to refer to a specific Twitter user. One could apply the Twitter Name Entity Recognition (NER) tool [13]. As noted in [13], the reason behind using a NER specifically trained for twitter is that the tweets are generally noisy and that traditional NER tools do not work well for micro-texts. Especially to further increase the accuracy of location entity, one can also pipeline NER with a dictionary based location detection mechanism which uses publicly available geo dictionary.

- *Sentiment Analysis:* refers to the use of text analytics approaches applied to the set of problems related to identifying and extracting subjective material in textual data. One of the most basic techniques is counting the frequency of each term with a pre-defined collection of positive words and negative words [18]. More complicated methods (e.g. classification, Python NLTK) can be found in the literature [10].

Segmentation & Event Detection. Before proceeding to event detection, we segment each topic cluster based on timeline, creating sub-clusters within each topic corresponding to non-overlapping window lengths of N days. Timeline based segmentation enables modeling of event evolution and temporal tracking. Events can be identified as peaks in timeline. Automatic techniques for event detection include wavelet-based, moving average, box-and-whisker [20].

Another segmentation approach is location-based, which build a gridded topic counts by grouping unique tweets of given topic in an area using Twitter coordinates. Events can be identified as high-density areas [7, 12].

4. RESULTS

Since the data analysis is not the main focus of our work, we only report highlighted results in the English tweets of the Guanajuato dataset.

Syntactical Analysis. We report the top frequent and co-occurrent elements of the dataset:

- *Top frequent terms:* are depicted in Figure 4, including 'others', 'love', 'follow', 'de', 'lol', 'GTO', etc. We removed stop words, hashtags, links, emoticons, etc. but English 1-gram words.

- *Top frequent hashtags:* are illustrated in Figure 5, including '#MexicoNeedsWWAT', '#friends', '#love', '#WWAT', '#boxing', '#selfie', etc.

- *Top co-occurrent hashtags:* the top-3 co-occurrent pairs of hashtags are listed below with their number of co-occurrences in the data. Note that this is counted for English tweets only. Moreover, we do not report the co-occurrent terms since the meaning of co-occurrent terms is similar to topic modeling.

```
('#nolazydays', '#norestday', 28)
('#HondurasNeedsWWAT', '#WWAT', 23)
('#mexico', '#sanmigueldeallende', 15)
('#CentroAmericaNeedsWWAT',
                  '#HondurasNeedsWWAT2014', 13)
('#Billboards2014', '#ElPulsoBillboard', 13)
('#mcc', '#nolazydays', 13)
```

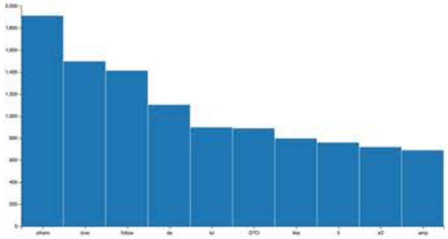


Figure 4: Top frequent terms

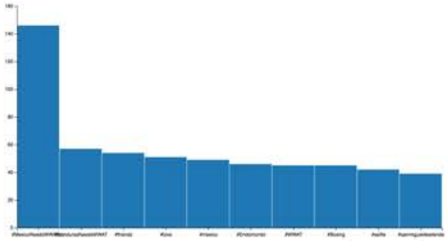


Figure 5: Top frequent hashtags

Semantic Analysis. We report the top-10 topics detected from the English tweets. We use the PyLDAvis [17] (a Python port of LDAvis) to visualize the results of topic modeling as well as allow users to configure parameters and explore further interesting findings. The interface in Figure 7 has two views. The left-hand side view presents the detected topics and their marginal frequency over the document collection. The distance between topics is the Jensen-Shannon divergence with multidimensional downscale to 2 [1]. In other words, the distance represents the amount (weighted by frequency) of overlapping terms between two topics; non-overlapping terms are not counted (or counted as zero similarity).

Table 2: Example of detected topics

Topic	Top relevant terms
2	follow,like,good,make,celaya,come,night,starbucks,way,ve,gto
4	lt,time,san,miguel,allende,im,got,gto
6	leon,gto,la,el,centro,feel,thanks,club,commercial,new

Table 2 illustrates the detected topics that are similar to each other (placed closed to each other in the view). In this case, the reason these topics (2,4,6) are similar to each other because it contains the same word 'gto'. Note that we only extract 1-gram words; further investigations required if necessary.

Time-based Event Detection. After having the extracted topics or hashtags, we can start grouping the tweets along these dimensions and perform event detection on the time series. In the first case study, we simply track the time series of a top-frequent hashtag, namely '#WWAT', which is a concert tour officially kicked off on 25 April 2014 in Colombia. Figure 6 presents the result, which implies an event starting on 25 April, reaching its peak on 26 April and falls off on 27 April, matching the real event. If we only retain the tweets in this period and extract the topics, we have Table 3.

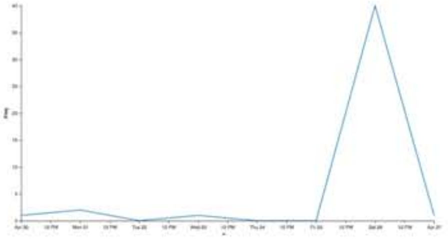


Figure 6: Event detection based on timeline of a hashtag

Table 3: Detected topics for WWAT event

Topic	Top relevant terms
1	love,follow,make,happy,hey,mexico,kisses
8	leon,need,gto,dates,tour,oh,tonight,guanajuato
3	san,miguel,allende,bar,good,el,gto

5. CONCLUSIONS

In this paper, we have developed an end-to-end and unified framework that helps social researchers to detect and analyze events in social media. As the source codes used in the benchmark are publicly available [2], we expect that the preliminary findings presented in this paper will be refined and improved by the research community, in particular when more data become available, more experiments are performed, and more techniques are integrated into the framework in the future.

Although the framework is just a protype and the findings are limited on English tweets only, our techniques open up several future directions of research. First, one can develop an online version of our framework to incorporate streaming data, as the online versions of topic modeling and event detection are available [19, 3]. Second, one can build on top of our framework for complex event processing, which matches, aggregates, formulates, and abstracts small events into a complex event for deeper understanding [6, 21].

6. REFERENCES

[1] https://cran.r-project.org/web/packages/LDAvis/vignettes/details.pdf, 2016.

[2] https://github.com/tamlhp/csm, 2016.

[3] H. Abdelhaq, C. Sengstock, and M. Gertz. Eventweet: Online localized event detection from twitter. *Proceedings of the VLDB Endowment*, 6(12):1326–1329, 2013.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[5] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[6] M. A. Cameron, R. Power, B. Robinson, and J. Yin. Emergency situation awareness from twitter for crisis management. In *Proceedings of the 21st International Conference on World Wide Web*, pages 695–698. ACM, 2012.

[7] D. Cheng, P. Schretlen, N. Kronenfeld, N. Bozowsky, and W. Wright. Tile based visual analytics for twitter big data exploratory analysis. In *Big Data, 2013 IEEE International Conference on*, pages 2–4. IEEE, 2013.

[8] D. Griffiths and M. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems*, 16:17, 2004.

[9] G. Ifrim, B. Shi, and I. Brigadir. Event detection in twitter using aggressive filtering and hierarchical tweet clustering. In *Second Workshop on Social News on the Web (SNOW), Seoul, Korea, 8 April 2014*. ACM, 2014.

[10] B. Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[11] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 889–892. ACM, 2013.

[12] F. Morstatter, S. Kumar, H. Liu, and R. Maciejewski. Understanding twitter data with tweetxplorer. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1482–1485. ACM, 2013.

[13] A. Ritter, S. Clark, O. Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics, 2011.

[14] S. Ruiz-Correa, D. Santani, and D. Gatica-Perez. The young and the city: Crowdsourcing urban awareness in

a developing country. In *Proceedings of the First International Conference on IoT in Urban Space*, pages 74–79. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

[15] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.

[16] D. Santani, S. Ruiz-Correa, and D. Gatica-Perez. Looking at cities in mexico with crowds. In *Proceedings of the 2015 Annual Symposium on Computing for Development*, pages 127–135. ACM, 2015.

[17] C. Sievert and K. E. Shirley. Ldavis: A method for visualizing and interpreting topics. In *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, pages 63–70, 2014.

[18] P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.

[19] Y. Wang, E. Agichtein, and M. Benzi. Tm-lda: efficient online modeling of latent topic transitions in social media. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 123–131. ACM, 2012.

[20] J. Weng and B.-S. Lee. Event detection in twitter. *ICWSM*, 11:401–408, 2011.

[21] E. Wu, Y. Diao, and S. Rizvi. High-performance complex event processing over streams. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 407–418. ACM, 2006.

[22] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *European Conference on Information Retrieval*, pages 338–349. Springer, 2011.

APPENDIX

Regular Expressions for Tokenization.

```
#Emoticons: eyes [:=;], nose [oO\-]?,
            mouth [D\)\]\(\]/\\OpP]
#HTML tags: <[^>]+>
#@-mentions: (?:@[\w_]+)
#hashtags: (?:\#+[\w_]+[\w\'_\-]*[\w_]+)
#numbers: (?:(?:\d+,?)+(?:\.?\d+)?)
#words with - and ': (?:[a-z][a-z'\-_]+[a-z])
#other words: (?:[\w_]+)
#anything else: (?:\S)
```

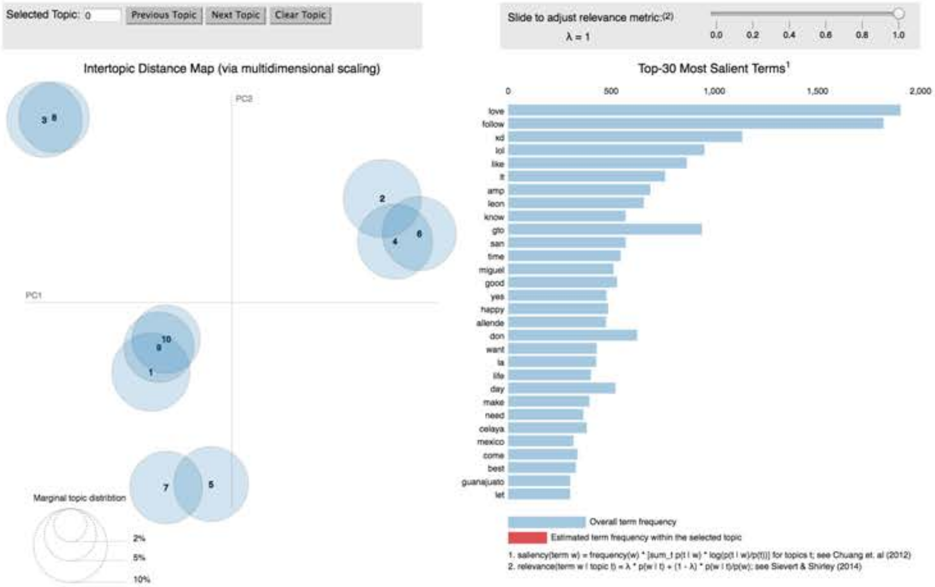
Development Environment.

```
Python=2.7
pyLDAvis=1.3.0
sklearn=0.17.1
pandas=0.18.0
nltk=3.2
```

Demos.

```
python -m SimpleHTTPServer 8889

http://localhost:8889/test.en.term_freq.html
http://localhost:8889/test.en.count.time_chart.html
http://localhost:8889/test.en.topic.time_chart.html
http://localhost:8889/test.en.term_freq.html
http://localhost:8889/test.count.time_chart.html
 http://localhost:8889/test.en.geo.html
 http://localhost:8889/topics.html
```



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 7: Interactive interface of topic analysis