Streaming Algorithms for Event Detection and Tracking

Report submitted as a partial fulfillment of the course

CS F266 - Study Oriented Project

Prepared by

Abhishek V Joshi

2015A7PS0116P

Under the guidance of

Prof. Navneet Goyal &

Prof. Poonam Goyal

Department of Computer Science and Information Systems

ADAPT Lab, BITS Pilani

First Semester (2017 – 2018), Semester Report

Acknowledgement

I would like to express my gratitude to **Prof. Navneet Goyal** for giving me the opportunity to work on the Project “Streaming Algorithms for Event Detection and Tracking” at ADAPT Lab, BITS Pilani. Special thanks to **Prof. Poonam Goyal** for providing her valuable inputs and insights for my work. I thank **Prerna Kaushik Ma’am** for helping me out in every possible way and identifying the key features of my work. I would also like to acknowledge my fellow lab-mate, Shivankit Gaind for helping me learn and motivating me to work.

Table of Contents

Abstract

The aim of the project is to use information rich streams of data from multiple social media platforms for detection and tracking of real life events. Twitter being the fastest growing social networking service that supports micro-blogging became the obvious choice to start with. My major focus was on efficient detection, tracking and De-multiplexing similar events without the use of geo-tags.

Introduction

Twitter allows its users to share short messages of no more than 140 characters, commonly known as tweets, from their accounts. It is estimated that over 500 million tweets are posted every day [1]. These tweets contain information about everything ranging from the routine activities of a user to globally and locally emerging events. With the evolution of state of the art stream mining algorithms and the growth of computational resources in the recent past, it has become feasible to extract information from these high speed data streams.

Several approaches have been successfully proposed to use the massive amount of data that the Internet provides. Some examples show that researchers have developed methods for sentiment analysis of a user from their posts, Sarcasm detection using behavioral modeling, real time event detection, etc. The obvious choice to test and deploy the approaches was Twitter due to its unique features and large amounts of publicly available data.

Event detection, in our context, refers to the identification of significant entities which describe some real life occurrence. Tracking of events is nothing but monitoring the change in these entities, in other words changes in the description of some event, with time. De-multiplexing is the process of separating identities of similar events that may be occurring at different parts of the world.

To put in crude terms, all the data will have to be converted into some kind of a numerical representation that will aid us in achieving our goals effectively and efficiently. Each kind of event will have a corresponding numerical representation, let’s call them signals. When new events start emerging, spikes in the signals of that particular event will be observed. While evolving events will tend to shift the signals from one form to another in a smooth and gradual manner, it is natural to understand that dying events will have their signals flattened out over time.

The process of Retrospective event detection will already know the kind of signals it should observe for detecting a certain kind of event. While new event detection will have to look for sudden absurd spikes in the distribution of the incoming signals.

The process of representing events is similar to Topic modeling in terms of the semantics. Hence many ideas for event detection have been borrowed from topic modeling and natural language processing. Topic modeling has been very widely studied, for traditional kind of media sources like newspapers, documents and academic papers [2][3][4]. Such approaches simply assume that all the data be available in the memory and can be used multiple times during the process.

Traditional methods for topic detection, albeit pose new challenges for social media platforms like Twitter due to the following factors:

- Limited and short size of the tweets (particular to Twitter).
- High velocity of incoming data.
- Tremendously high data volumes.
- Extremely dynamically evolving content.
- Lots of noise.
- Multi-modality of the data.

These challenges have been discussed below in a little more detail:

1) It is difficult to extract useful information out small natural entities, unless they are packages of highly complex, organized data and we know the exact algorithm to extract it. However tweets are simple human expressions written in a concise form thus making it a difficult job to extract any useful information out of a single one of it. While one can easily argue that using more number of tweets will achieve better results, one also will have to keep in mind the unstructured distribution of information across incoming tweets. Hence rendering the traditional methods of topic modeling potentially useless. While some researchers consider the limited size of a tweet to be a drawback for apply mining techniques for pattern extraction and information retrieval, others argue that this limitation if used correctly allows us to generate better feature sets to do the data mining job.

2) The high volume and velocity of the streaming data is a challenge for computationally heavy algorithms. To detect events in real time, the techniques must ideally be of one pass type, wherein either the data can be accessed only once, or it stays in the memory for a very brief period of time. This challenge is also addressed by emergence of high performance computing and parallel technologies.

3) Due to highly dynamic nature of the data streams, conventional text summarization techniques cannot be employed.

4) Users post lots of tweets that contain irrelevant information subject to our task, sometimes even leading to an adversarial dataset. For example, at the time of NASA's curiosity Mars rover mission launch, Bobak Ferdowski's, the astronaut's, hairstyle became more popular than the main event on social media platforms [6]. This lead to absurd results by event detection algorithms. Users also tend to share memes, sarcastic comments and other types of potentially useless content through their accounts.

5) The conventional topic modeling methods assume the data to be of the type text only. But with social media, one can encounter multi-modal data involving images, gif's, videos and text.

In our project we employ a method presented by Wang et. al [5]. in their paper which proposes an event detection method that de-multiplexes events without the use of location information in an entirely unsupervised NLP-free fashion. It processes only the text data from the tweets and uses a novel sliding time window based approach to monitor the evolution of events.

Literature Survey

Farzinder et.al.[6] very comprehensively covered the ideas presented by various researchers in a survey paper. They have categorized the event detection process into two types: New event detection and retrospective event detection. They have also identified two categories of events, namely specified and unspecified. This categorization is done based upon the semantic knowledge that is available to us before the event detection task. It also extensively covers Twitter as a source of information, highlighting the features that makes it the most looked at social media platform for topic modeling task. It also provides an overview of traditional topic detection techniques involving document-pivot and feature-pivot techniques.

In a paper by Hongyun Cai et. al.[7] named “Indexing Evolving Events from Tweet Streams”, a Multi-layered inverted list data structure was proposed. The paper uses a Single pass incremental clustering algorithm, to make clusters of the incoming data such that each cluster corresponds to an event. To classify a new tweet into one of the clusters, a nearest neighbor type of query is required, for which the MIL data structure is used. The retrieval of the nearest neighbor is performed using an algorithm hybrid of Depth first search and Apriori pruning. Evolution of events was captured using the dynamics of the clusters, i.e. the merge and split operations for any cluster.

Another technique proposed by Mariam A et.al.[8] named “A rule dynamics approach to event detection in Twitter” uses the hash-tags of tweets to produce rules. This method disregards all the words apart from the hash-tags. It generates rules in two consecutive time windows using the frequent item sets (hash-tags). A rule matching algorithm is used to filter out only the relevant rules from the two time windows. These rules are then classified into one of the following classes : Unexpected consequent rule, Unexpected conditional rule, Emerging rules and New rules. After classification, these rules are mapped to real life events using a trend analysis algorithm. Refer the figure below.

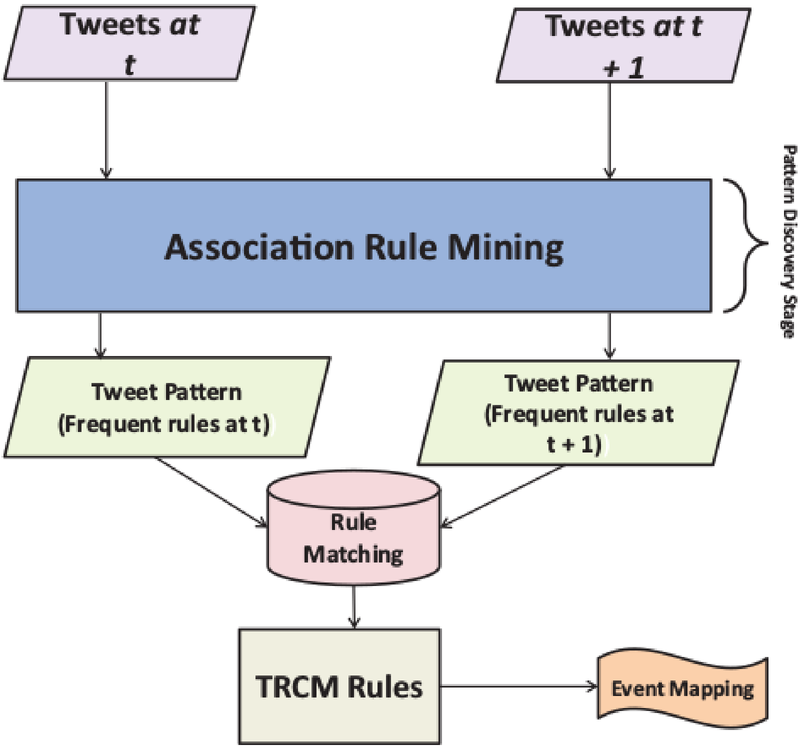


Fig. 1. TRCM process.

Katragadda et. al.[9] have proposed a graph based model to detect an event as soon as it is onset. It generates a co-occurrence graph in real time, prunes unnecessary edges and vertices using thresholds given as user parameters, makes clusters out of them and validates the relevant events out of the formed clusters.

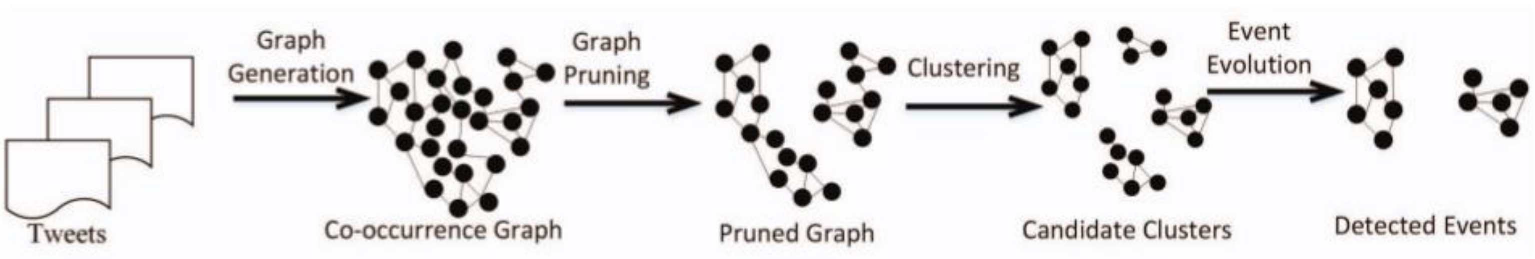


Fig. 1. Workflow for Event Detection at Onset

The same set of authors have also proposed a way to combine data streams from multiple social media platforms at various stages of graph generation to yield better results.[10]

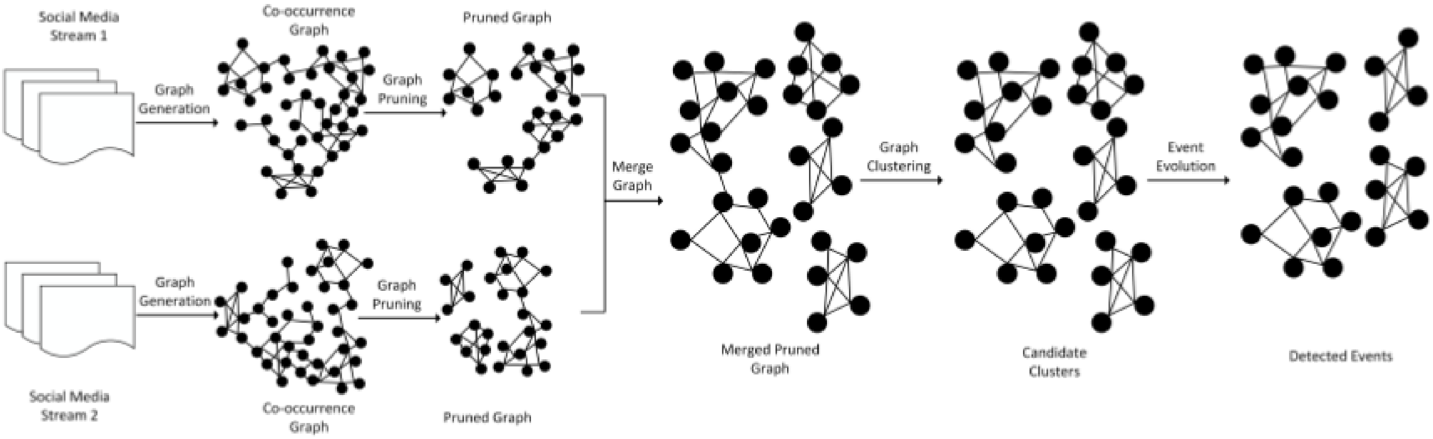
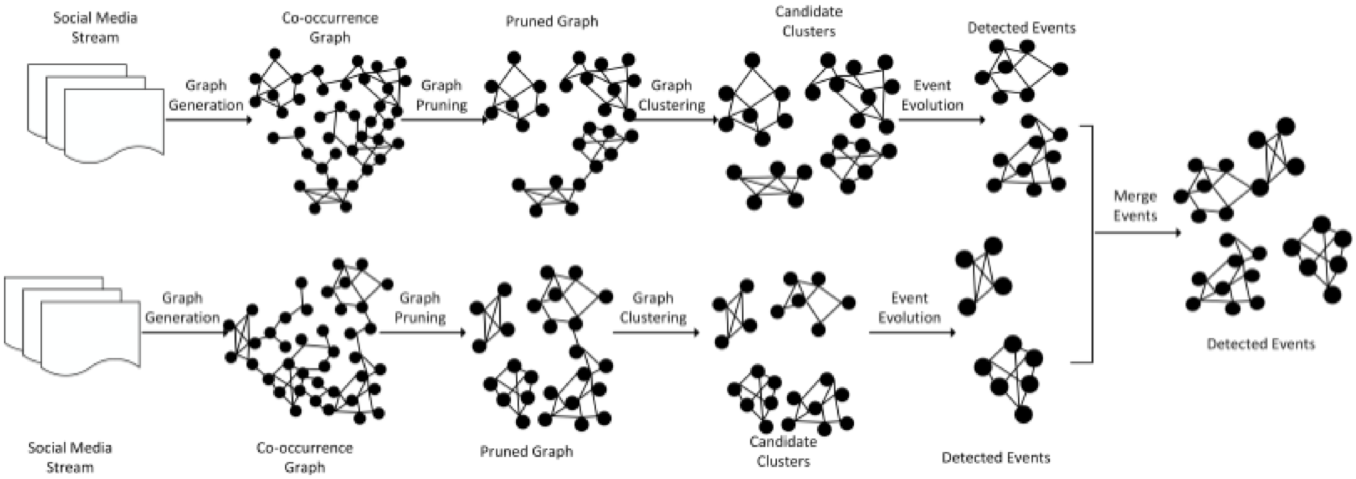


Figure 3: Workflow to Detect Events at Onset from a Multiple Data Streams: Graph Pruning



StoryLine

StoryLine[5] is a novel social sensing (back-end) service that exploits real-time content posted on social media to detect, demultiplex, and track instances of physical events of interest to the user.

The De-multiplexing works on the sparsity argument given by the author. The English alphabet consists of around 10,000 words. With the huge volume of data that comes in, the tweets are able to very densely populate the space of words. Hence distributions of two different but similar events will have a very high tendency of overlapping with one another. The sparsity argument says that this overlap will be greatly reduced if the vector space would have been a cross product of English words with English words. Hence the events will naturally be De-multiplexed. Hence this approach works on keyword pairs instead of only keywords.

While the same event instance is characterized by similar word pairs from different tweets, the probability that the same word pairs characterizes two separate event instances reduces by a substantial amount.

Whenever a new event appears, the discriminative keyword pairs describing that event will occur disproportionately in the current time window as compared to the previous time window. This is another way of saying that the signals corresponding to that event will suffer a spike. Such pairs offer more information gain as they do not normally co-occur.

To keep a track of events on the timeline, a sliding window approach is used, and all the discriminative keyword pairs from the previous window are inherited in the new window. To smoothen out the distribution of tweets across two consecutive time windows, they are made to overlap each other by a certain amount.

Each generated keyword pair corresponds to a separate bin to begin with, such that every tweet that contains a particular keyword pair is put in the corresponding bin. Various bins are consolidated if their contents are similar. The content is regarded as the statistical distribution of words in the tweets of a particular bin. If two bins are more similar to each other than a set threshold value, then they are consolidated to form a single bin.

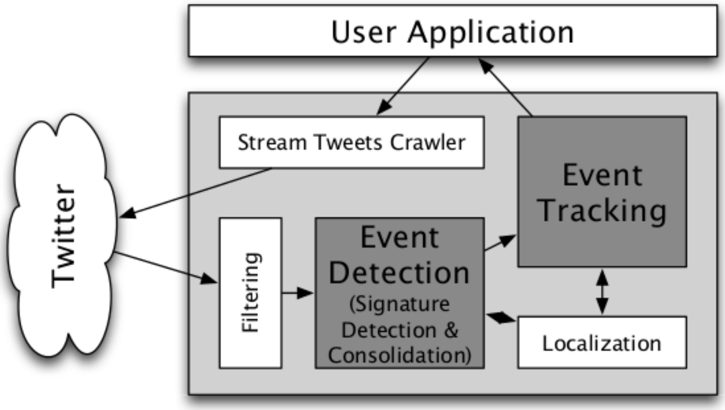
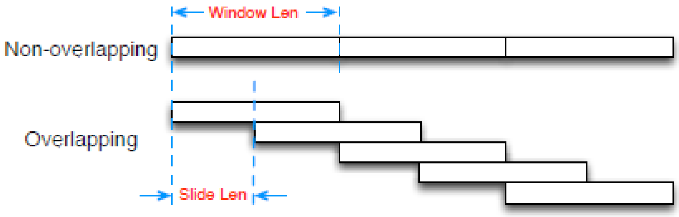


Figure 3: Event Tracking System Architecture

The bins from the k^{th} window are consolidated first; then these are consolidated with the bins of the $k\text{-}1^{th}$ window. To capture the evolution of a particular event, an overlapping sliding window approach is used. It helps smooth out the changes in the lexical frequency distribution of fast developing events over time.



The evaluation of the consolidation algorithm is done using the error rate metric. It is defined as ratio of the number of incorrectly grouped 2-keyword signature pairs to the total number of 2-keyword signature pairs. A 2-keyword signature pair is said to be incorrectly grouped if two signatures corresponding to the same event are put into different groups or if two signatures corresponding to different events are put into the same group. Jaccard distance similarity metric showed the best results in consolidation of bins.

The evaluation of the de-multiplexing is done using the precision of the detected events; that is the ratio of the number of true events detected to the total number of events detected by the algorithm.

Implementation

The flow of work in the implementation is as follows:

1) To simulate the sliding window approach with real time event detection, the tweets are first discretized in buckets and stored on the disk depending upon the time at which they appeared.

2) The tweets are first cleaned of any URLs, punctuations and special characters, such that only words/phrases remain in its content. Then a stop-word corporus (provided by Prerna ma'am) is used to clean the tweets of any stop-words occurring on Twitter. Tweets are stored along with their IDs and timestamps in JSON object format on the disk.

3) Multiple buckets are read into the memory from the disk, to generate one time window. This time window will slide over the buckets in an overlapping manner.

4) Each tweet's content is sorted, then keyword pairs are generated. Each keyword pair corresponds to a bin. Each bin contains all the keywords of the tweets in which that particular keyword appeared.

5) Two bins are consolidated if its content similarity exceeds a certain threshold. This consolidation is applied in one time window until no more of the bins can be consolidated.

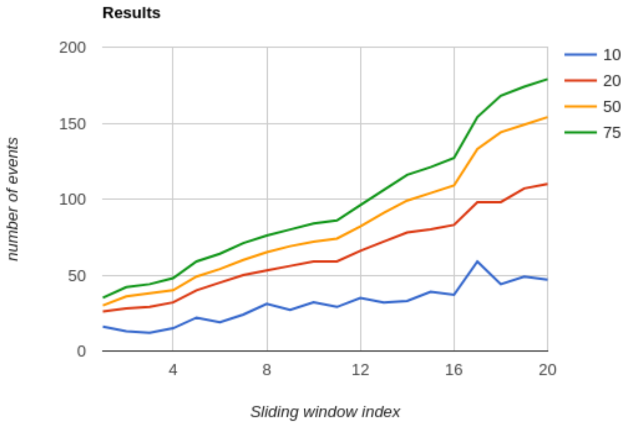
6) The discriminative keyword pairs from the previous time window whose cluster of tweets is monotonically increasing are chosen to add in the next time window as well. Then an inter-window consolidation is used in a similar fashion as that of intra-window.

7) This time window slides until we reach the end, outputting events (the consolidated bins) at each step.

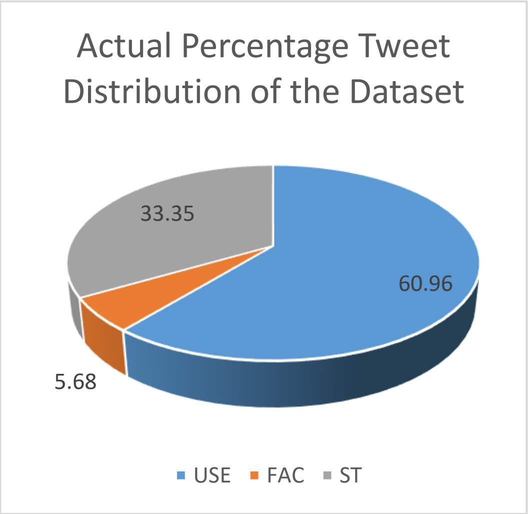
The system has been implemented in Java for the purpose of standardized performance evaluation. A Hash-set data structure has been used wherever a look up operation is required and the ordering of the data does not matter, to speed up the implementation.

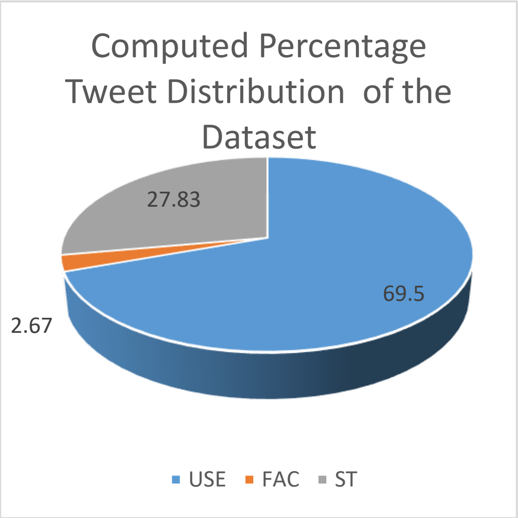
Results

The system is tested on various data sets of comparatively small size for quick evaluation. To test the variation of the number of events detected with respect to the similarity threshold on consolidation, 2017 Wimbledon women's championship dataset was used. The general trend show that the number of events detected increase as the threshold increases, because less number of bucket consolidations. So the trend is consistent with the author's hypothesis. The number of events that are detected also have a positive correlation with the time instants.

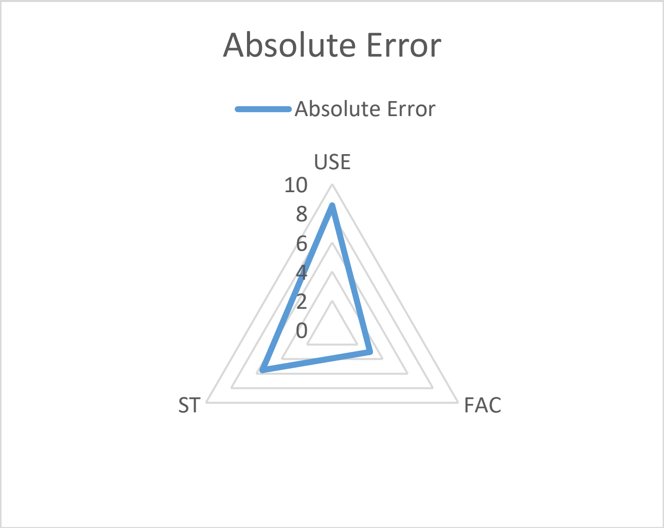


This method is used for detecting events from a general stream in an unsupervised fashion, hence three datasets consisting tweets of Super Tuesday, FACup and US Elections were combined and fed to the algorithm. The bucket which has maximum tweets of any of these previously known events will correspond to that event. A distribution of the buckets corresponding to the given three events is then constructed and compared to the original distribution of the tweets.





The web graph below shows the absolute error measured from the above pie charts.



Critique

The approach presented in the paper is extremely naive and lacks many fundamental aspects that need to be taken care of. The consolidation algorithm is inherently O(n^2) on the number of discriminative keyword pairs that are generated. Thus does not solve the purpose of real time event detection. During a popular event like the Nepal Earthquakes, Tweets are generated at a rate of around 400 tweets per minute pertaining to that event. This generates tens of thousand discriminative keyword pairs for a window of 5 minutes, processing which takes more than 20 minutes on a standard machine.

The paper claims to have detected relevant events in an unsupervised NLP free way, hence lacks the superiority of semantic modeling, generally considered to be of great importance in crowd sensing sentiment analysis and topic detection. The paper also does not specify the meaning of `contents` with respect to the keyword pair bins.

Since the only learning part is in the consolidation algorithm, which itself forms the bottleneck of the whole approach, thus lack of se mantic knowledge and inefficiency are two major drawbacks of the paper.

Future Work

Further work will include the usage of global word-vector models to enrich the semantic knowledge of approaches. This can be used to create two separate vector spaces, in which true events may be detected using cross validating the formed clusters. We will be looking into neural network based approaches for stream mining as well.

References

- [1] https://about.twitter.com/company
- [2] J. Kleinberg, "Bursty and hierarchical structure in streams," *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 373–397, 2003.
- [3] G. Kumaran and J. Allan, "Text classification and named entities for new event detection," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2004, pp. 297–304.
- [4] Q. He, K. Chang, and E.-P. Lim, "Analyzing feature trajectories for event detection," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 207–214.
- [5] Shiguang Wang, Prasanna Giridhar, Hongwei Wang, Lance Kaplan, Tien Pham, Aylin Yener, and Tarek Abdelzaher. 2017. StoryLine: Unsupervised Geo-event Demultiplexing in Social Spaces without Location Information. In Proceedings of .e 2nd ACM/IEEE International Conference on Internet-of-
- .ings Design and Implementation, Pittsburgh, PA USA, April 2017 (IoTDI'17), 11 pages.
- [6] Atefeh, F., & Khreich, W. (2015). A survey of techniques for event detection in twitter. *Computational Intelligence*, *31*(1), 132-164.
- [7] Cai, H., Huang, Z., Srivastava, D., & Zhang, Q. (2015). Indexing evolving events from tweet streams. *IEEE Transactions on Knowledge and Data Engineering*, *27*(11), 3001-3015.
- [8] Adedoyin-Olowe, M., Gaber, M. M., Dancausa, C. M., Stahl, F., & Gomes, J. B. (2016). A rule dynamics approach to event detection in twitter with its application to sports and politics. *Expert Systems with Applications*, *55*, 351-360.
- [9] Katragadda, S., Virani, S., Benton, R., & Raghavan, V. (2016, July). Detection of event onset using twitter. In *Neural Networks (IJCNN), 2016 International Joint Conference on* (pp. 1539-1546). IEEE.
- [10] Katragadda, S., Benton, R., & Raghavan, V. (2017, January). Framework for Real-Time Event Detection using Multiple Social Media Sources. In *Proceedings of the 50th Hawaii International Conference on System Sciences*.