

**Event Detection and User Interest Discovering in**

**Social Media Data Streams**

Prepared as partial fulfilment for the assignment in course

**CS F469**

**Information Retrieval**

Taken by Instructor Incharge

**Prof. Poonam Goyal**

**Akankshya Mishra 2016A7PS0026P**

**Narapareddy Bhavana 2016A7PS0034P**

**Karabee Batta 2016A7PS0052P**

**PROBLEM STATEMENT**

Event detection, i.e. identification of the occurrence of an event, in a time period t, i.e. in real time using social media data streams such as Twitter.

**BACKGROUND OF THE PROBLEM**

**I DESCRIPTION OF SELECTED APPLICATION DOMAIN**

Events can be defined as real-world occurrences that unfold over space and time (Allan et al. 1998; Troncy et al. 2010; Xie et al. 2008; Yang et al. 1998). Event detection involves identification of such occurrences by analysis of certain data.

Due to the expansive growth of social media posts in the recent past, several modes of research have been explored using their data streams. Event detection in such data streams is a useful concept as it may provide quick, new information and lead to increased effectiveness in transparency, disaster help, and many other domains.

The data used for event detection is often in the form of a stream(especially considering social media like Twitter), or in an unstructured format, such that the event information has to be efficiently extracted out of this pool. This is typically achieved using document/text classification and similarity analysis to form event occurrence clusters.

**II MOTIVATION OF THE PROBLEM**

Event detection from streams of news stories has been explored and studied. However, using social media data streams, such as from Twitter, poses a challenge due to the ill-structured, restricted text that must be evaluated. In addition to these, there are several meaningless and polluted texts that are irrelevant to detection of events. Even within a small time period, the volume of data generated is too large to be manually examined for meaningful information, such as the occurrence of an event.

Thus, to facilitate efficient retrieval of event occurrences in such data streams, algorithms must be developed.

**III TECHNICAL ISSUES INCLUDED**

Event detection algorithm is implemented using authority value and minimum distance values.

**RESEARCH GAP:**

We still are in need of a better evolution model, one that keeps track of past related events and by using them as training sets can identify future events related to it. It can also help us indirectly to keep track of the changing interests of the influential users, Keeping track of one can help us detect the other.

Also the current methods for clustering and filtering apply to tweets that are in English, non english words are very likely to get filtered out ; mainly due to the fact that English has the largest corpus on the twitter dataset.1

**LITERATURE SURVEY:**

Micro-blogging is a broadcast medium that allows users to exchange small digital content such as short texts, links, images, or videos. Twitter is the most popular and fastest growing micro-blogging platform in use today, where the text posts have to adhere to a 280 word limit. According to alexa rankings, it is the eighth most popular site in the word.Its 336 million users contribute to its daily broadcast of over 500 million tweets. 1 Tweets can be seen as a dynamic source of information enabling individuals, corporations, and government organizations to stay informed of “what is happening now.”

It currently provides different ways for users to converse and interact by referencing each other in posted messages in a well-defined markup vocabulary.Placing the “@” symbol before a user name (also called a handle, “@username”)creates a *mention* or a *reply* link to the referenced user’s account. A mention is used anywhere in the message for signaling that the mentioned user is also registered on Twitter. A reply is a special mention from one user in response to another user’s message starting with the replied-to @username (Honeycutt and Herring 2009). Mentions are displayed in the referenced user accounts to keep track of messages mentioning their names. Twitter also allows users to forward or *retweet* someone else’s tweet to their followers. It is commonly carried out by using the RT prefix before the user name that originated the message, “RT @username.” Retweeting is a common practice on Twitter to share useful or interesting information while giving credit to the original user (Boyd et al. 2010). Topics on Twitter can be categorized by a *hashtag*, which is any keyword preceded by a hash sign “#” (e.g.#nlptechnologies). Hashtags were developed as a means to create groupings on Twitter. Twitter users can use hashtag to indicate the subject of their messages, to collate tweets from different users on a shared subject, and to regularly track specific events in real time. Twitter provides an application programming interface (API),2  which allows developers to programmatically access the public data streams as well as many features of the service. For instance, Twitter streaming API provides filtering by location, keywords, author, and others. The availability of Twitter data has motivated significant research work in various disciplines and led to numerous applications and tools.

This leads us to a specific use of tweet analysis, i.e event detection. Event detection includes finding important happenings on a global or a large regional scale.

However we mostly are interested in events that are happening on a global scale, per say global events. But there exist multiple trivial and local events. Typically an event is triggered by the use of “#”, called a hashtag. The more is the number of people using the hashtag , the more popular the event becomes. Traditional mass media channels take some time to relay the information to people, but with forums like twitter, new gets spread almost immediately.

We need a method in check that ensures:

1. General and trivial information gets filtered out
2. We are looking at events of mass interest and impact (e.g. global events)

There are various techniques for event detection that are in use .

1. Document pivot techniques:

An event consists of three phases; data pre-processing, data representation and data organisation or clustering. Data pre-processing deals with filtering out stop-words and other commonly used phrases. Traditional data representation for event detection involves the *term vectors* or *bag of words* whose entries are nonzero if the corresponding terms appear in the document.3  Each term in the vector is typically weighted using the classical term frequency–inverse document frequency (*tf-idf* ) approach. Since the temporal order and the semantic features of the text are avoided by vectors, a model relying purely on such techniques will not be able to capture a relationship between events.

An entity vector must answer the 4 W's 4  i.e what,where,when and why Similarity between events is measured by Euclidean distance, Jaccard’s coefficient and cosine similarity.More recently, other similarity measures have been proposed such as the Hellinger distance and clustering index.

Event detection can be further divided into retrospective event detection(RED) and new event detection(NED).RED focuses on discovering previously unidentified events using established historical facts and figures. NED focuses on discovering new events from live streams in real time. Both use clustering based algorithms. These include the traditional k-means and its variants, k -median and k-means++.

New event detection has been characterized as a *query-free* retrieval tasks because the event information is not known a priori and hence cannot be expressed as an explicit query. In contrast to RED, NED must provide decisions (new or old events)as documents arrive. Hence it typically uses greedy algorithms. These techniques are called document pivot techniques because they cluster documents based on their textual similarity. This cannot be the only criterion for selection.

1.2 Feature pivot techniques:5

Due ti the recent increase in interest of bursty event detection techniques in media, we need something more reliable than the basic textual similarity. Feature pivot techniques model an event in text streams as a bursty activity with certain features rising in frequency over concentrated periods of time, the underlying assumption being that some words would have increased usage over some time intervals. In his seminal work, it was proposed in infinite-state automaton to model the arrival times of documents in a streams to identify bursts that have high intensity over limited durations of time. The states of the probabilistic automaton correspond to the frequencies

of individual words, while the state transitions capture the burst, which correspond to a significant change in word frequency. Here word appearance can be modelled as a binomial distribution, identifying bursty words as a heuristic based threshold and grouped similar events. An incremental suffix tree data structure was applied to reduce the time and space constraints required for

online detection.

Further events can be specified or unspecified. NED focuses on unsupervised events, because no prior information is available about the event, hence it has to rely on the temporal stream of tweets.hence it also has to learn to discriminate from issues of mass importance from trivial events using scalable and efficient algorithms. Usually a naïve Bayes classifier is used for separation and an online clustering algorithm. Messages that are similar to each other are then grouped together to form a news story. Similarity between messages are based on tf-idf with an increased weight for proper noun terms, hashtags, and usernames. Results have shown that ranking according to the number of users is better than ranking according to the number of tweets and considering entropy of the message reduces the amount of spam messages in output. Each message is represented as a tf-idf weight vector of its textual content, and cosine similarity is used to compute the distance from a message to

cluster centroids. In addition to traditional preprocessing steps such as stop-word elimination and stemming, the weight of hashtag terms are doubled because they are considered a strong indication of the message content. The authors combined temporal, social, topical, and Twitter-centric features. These features have to be constantly updated with time because so do the trending events.

Another method for event detection is based on clustering of discrete wavelet signals built from individual twitter generated words.6  In contrast with Fourier transforms, which have been proposed for event detection from traditional media , wavelet transformations are localized in both time and frequency domain and hence able to identify the time and the duration of a bursty event within the signal. Wavelets convert the signals from the time domain to time-scale domain, where the scale can be considered as the inverse of frequency.7  Signal construction is based on time-dependent variant of document frequency–inverse document frequency (DF-IDF), where DF counts the number of tweets (document) containing a specific word, while IDF accommodates word frequency up to the current time step. A sliding window is then applied to capture the change over time using the H-measure (normalized wavelet entropy). Trivial words are filtered out on the basis of (a threshold set on) signals cross-correlation, which measure similarity between two signals as function of a time lag. The remaining words are then clustered to form events with a modularity-based graph partitioning technique, which splits the graph into subgraphs each corresponding to an event. Finally, significant events are detected on the basis of the number of words and the cross-correlation among the words related to an event. A continuous wavelet transformation based on hashtag occurrences combined with topic model inference using latent Dirichlet allocation(LDA) 6  has been proposed. Specified event detection is like a planned social event. It aims to detect events that somehow went undetected in the past.

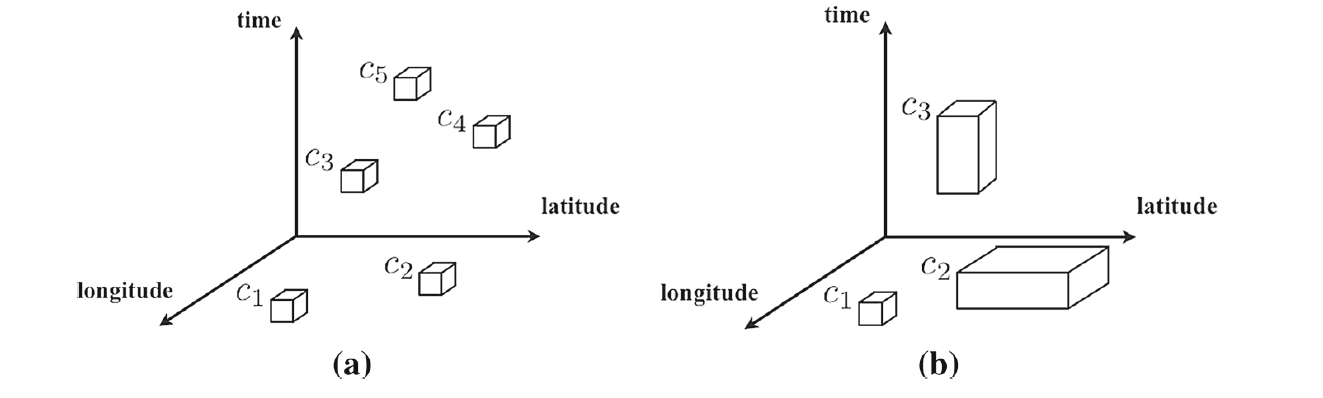
Factor graph model: 8

It simultaneously analyses individual tweets , clusters according to an event type and induces a canonical value for each event property. The motivation is to infer a comprehensive list of musical events from Twitter (based on artist–venue pairs) to complete an existing list (e.g., city event calendar table) by discovering new musical events mentioned by Twitter users that are difficult to find in other media sources. At the message level, this approach relies on a conditional random field (CRF) to extract the artist name and location of the event. The input features to CRF model include word shape; a set of regular expressions for common emoticons, time references, and venue

types; a bag of words for artist names extracted from external source . The factor graph model is then employed to capture the interaction between all components and provide the final decision. identify Twitter messages for an event, they begin with simple and precise query strategies derived from the event description and its associated aspects (e.g., combining time and venue). An annotator is then asked to label the results returned by each strategy or over 50 events that provide high-precision tweets. Quality indicators are required to retrieve individual tweets and check for their relevance /authority over the context. There is also a recency factor involved, regarding the temporal aspect of the tweet. 9

A different technique called Locality sensitive hashing can be implemented which uses bucket hashing to form the initial set of clusters and then uses tries to advance the formation and detection of an event.10

Another Multimodal tweet detection11 assumes a model in which each word in a tweet that uses the i-th hashtag is independently generated from the multinomial distribution with a single trial (i.e., categorical distribution and the geolocation is done using a von-Mises-fisher representation. this is not optimal as it ignores the temporal aspect of the tweet.



(a)shows events of similar scales localised in both space and time and (b) unlocalised events9

**REFERENCES:**

1.https://www.alexa.com/siteinfo/twitter.com

2.A survey of techniques for event detection in twitter, Farzindar Atefeh and Weil Khreich

3..Automatic targeted-domain spatiotemporal event detection in twitter, Ting Hua et al.

4.Cluster-discovery of Twitter messages for event detection and trending, Shakira Banu Kaleel ad Abdolreza Abhari.

5.Event Detection in Twitter Microblogging,Nikalaos D. Doulamis et al.

6.Event detection, tracking, and visualization in Twitter:a mention-anomaly-based approach, Adrien Guille, Cecile Favre

7.Real-Time Traffic Event Detection From Social Media, Di Wang et al.

8.Multimodal Event Detection in Twitter Hashtag Networks Yasin ,Yılmaz and Alfred O. Hero

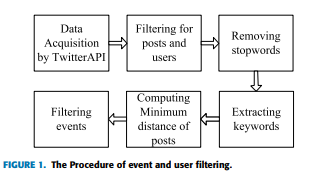
9.Multiscale event detection in social media, Xiaowen Dong et al

10.Real-Time Novel Event Detection from Social Media , Quanzhi Li et al.

11.Event detection over twitter social media streams, Xiangmin Zhou and Lei Chen

**SYSTEM DESCRIPTION**

**EVENT FILTERING:**



**I DATA ACQUISITION**

Tweepy is used along with Twitter Streaming API to capture the live stream of tweets.

After authorising the account, the stream is accessed with the filter() function, giving the language desired as English, and a list of stopwords to filter out tweets, as almost every tweet which is posted will contain one/two stop words.

**II FILTERING THE POSTS AND THE USERS**

Filtering is done using another python script, which processes the raw data into JSON format and selects the terms needed for the functionality: namely time of creation, id, retweet count, screen\_name, text, and user id.

Appropriate processing is done to ensure that the original of the retweeted tweet is taken into account for the event clustering process.

**III REMOVING STOPWORDS**

A dictionary of stopwords is used to filter the relevant words only. For this: Natural Language Toolkit (NLTK) is used. It is an open source Python library for Natural Language Processing. It includes a dictionary of stopwords which is used in the remove\_stopwords() function.

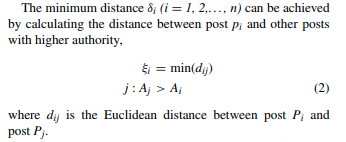
**IV EXTRACTING KEYWORDS**

Text preprocessing of the tweet text has been divided into multiple functions: removal of noise, word tokenization, normalization and stemming.

Normalization comprises of removing non-ASCII characters, converting all to lowercase, removing punctuations, and removing stopwords. The finalized preprocess() function is used to return a list of stemmed keywords.

**V COMPUTING MINIMUM DISTANCE**

Minimum distance between posts is computed according to the method specified in the paper:



**VI FILTERING EVENTS**

Filtering events is done using Algorithm 1 from the paper

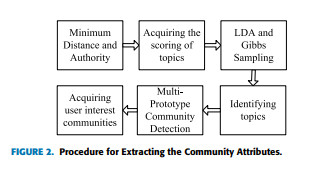
First, selection of top K event topics is done from the plot of Authority Matrix versus Minimum Distance matrix.

Then, the similarity matrix and the event-post matrix (dimensions K x N, where N is total number of posts) are computed.

An iterative algorithm then follows: until the event clusters do not change, or maximum iterations have been exceeded, the event-post and the similarity matrix values are recomputed.

The output is the event clusters.

**COMMUNITY ATTRIBUTES EXTRACTION:**



**I MINIMUM DISTANCE AND AUTHORITY**

Authority score of a tweet is calculated using a modified HITS(hyperlink-induced topic search) algorithm.

Minimum distance is calculated as described in paper:

**II ACQUIRING SCORING OF TOPICS & IDENTIFYING TOPICS**

This is done using LDA function

**IV MULTI-PROTOTYPE COMMUNITY DETECTION**

Multi-prototype community detection is done using Algorithm 2 of the paper

**EVALUATION STRATEGY:**

**1. LDA:**

Latent Dirichlet allocation (LDA) is a [generative statistical model](https://en.wikipedia.org/wiki/Generative_model) that allows sets of observations to be explained by [unobserved](https://en.wikipedia.org/wiki/Latent_variable) groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics. LDA is an example of a [topic model.](https://en.wikipedia.org/wiki/Topic_model)

Withplane notation, which is often used to represent probabilistic graphical model[s](https://en.wikipedia.org/wiki/Probabilistic_graphical_model) (PGMs), the dependencies among the many variables can be captured concisely. The boxes are "plates" representing replicates, which are repeated entities. The outer plate represents documents, while the inner plate represents the repeated word positions in a given document, each of which position is associated with a choice of topic and word. M denotes the number of documents, N the number of words in a document. The variable names are defined as follows:

*α* is the parameter of the Dirichlet prior on the per-document topic distributions,

*β* is the parameter of the Dirichlet prior on the per-topic word distribution,

ፀm is the topic distribution for document *m*,

𝝋k is the word distribution for topic *k*,

𝓩mn is the topic for the *n*-th word in document *m*, and

𝔀mn is the specific word.

**2. HEE:**

Events are first filtered using authority values in the first step, to discover key posts under all hot events.

The HEE model discovers the top k posts of relevance; done from the plot of authority vs post distance.

Then, the similarity matrix and the event-post matrix (dimensions K x N, where N is total number of posts) are computed.

If the authority values for 2 posts are the same , then we check for the minimum distance of posts, which shows us the key events for popular topics.

The proposed HEE method can detect the number of popular events under the different parameter k, and by setting different parameter k, and the proper number of popular events can also be discovered. The output is event clusters.

**EXPERIMENTAL RESULTS:**

After Data Acquisition step, we obtained roughly 60,000 tweets from the streaming API, which after preprocessing were a total of 46,443 distinct tweets made by 40,273 users.

The experiment was done using LDA topic detection model as well as Event detection model and results were thus compared.

The clusters were approximately the same.

The LDA topic model showed 36.134 s to train on the data for topic detection.

The Event detection model showed 22.167 s to detect event clusters.

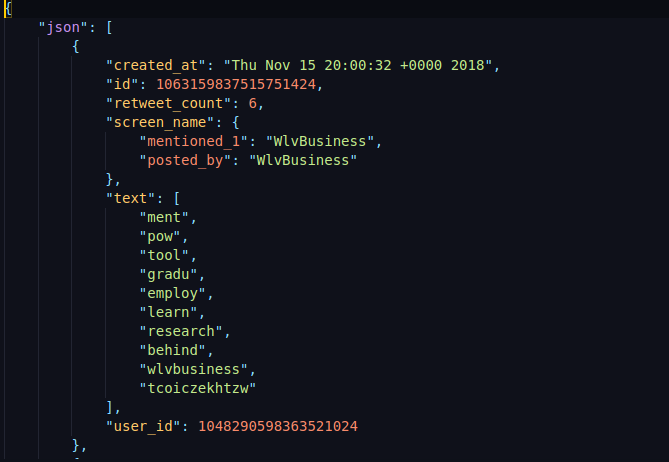
Python and Jupyter scripts used in experimentation include:

1. datastream.py

Streaming data from Twitter API

1. json\_extract.py

Extracting JSON format data as depicted in attached picture



1. preprocessfile.py

Function module to remove stopwords, non-ASCII words : basic text preprocessing

1. topkfiltering.py

Minimum distance and authority, hub score calculation according to modified HITS algorithm

1. cluster.py

Event detection clustering algorithm as described in the paper

1. LDA\_tweet.ipynb

LDA and Gibbs sampling for topic detection and scoring-used for comparative evaluation

1. topK.ipynb

Code for plot of minimum distance and authority, hub score according to modified HITS algorithm as described in the paper to get top K representative posts

**CONCLUSIONS**

Event clusters were generated by both the models (LDA andHEE). We ran both the models on the same dataset of 46,000 tweets. HEE performed better than LDA clocking in at 22.3 seconds. However, this was run on small sample of size 350.

**Improvements suggested:**

1. The paper enforced that every tweet has to be classified into one of the ‘k’ events found. This added noise in the dataset as many of them might not be related to any event but just a usual tweet.
2. The paper hasn’t suggested any threshold in similarity computation.
3. There was further scope for improvement if we manage to optimise the value of k in the searches. This will allow for better event evolution chain formation and will indirectly give scope for finding out potential influential users.