Event Detection in Social Media
A Short Survey

Presented by
-Satarupa Guha
-Ayushi Dalmia
Event Detection

• The underlying assumption is that some related words would show an increase in the usage when an event is happening.

• An event is therefore conventionally represented by a number of keywords showing burst in appearance count.

• Goals
  • Identify if an event of interest has occurred
  • Characterize the event – location, time, type
  • Detect as early as possible
  • Detect as accurately as possible
Applications of real time event detection

• Early detection of disease outbreaks
• Crime hot-spot detection
• Environmental Monitoring
• Online Marketing
• Opinion Tracking
Event Detection from Social Media – Twitter

• Dynamic and Diverse
• Continuously changing
• Large amount of user generated content – 500 million tweets are generated everyday
• Short and self-contained nature
Timeline

• Event Detection and Tracking in Social Streams
  • TwitterMonitor: Trend Detection over Twitter Stream
  • En Blogue – Emergent Topic Detection in Web 2.0 Streams
  • Streaming First Story Detection with application to Twitter
  • Bieber no more: First Story Detection using Twitter and Wikipedia
  • Extraction and Compilation of Events and Sub-events from Twitter
  • Twevent: a segment based event detection from Tweets
  • Open Domain Event Extraction from Twitter
  • Structured Event Retrieval over Microblog Archives
  • Discover Breaking Events with Popular Hashtags in Twitter
  • Social event detection with interaction graph modeling.
Event Detection and Tracking in Social Streams

Sayyadi, Hassan, Matthew Hurst, and Alexey Maykov

ICWSM. 2009.
Event Detection and Tracking in Social Stream

- **Idea**

  - **Episode**: A sub-event, say rally at XYZ place
  - **Saga**: A larger event, say presidential elections
  - Present a keyword graph method that detects and describe events in social media

Event Detection and Tracking in Social Streams – Approach

Represent the event using keywords

- Build a key graph

Community detection on key graph

Event Detection and Tracking in Social Streams

– Approach

Represent the document by a set of keywords that maximally distinguish the document

- Document summarised by a set of terms does not imply that some other episode cannot be described by some subset of those terms

Represent the document by a set of named entities

- A document summarized by the terms 'Hillary' and 'Clinton' could not be used to discover related documents that only mention 'Hillary' or only mention 'Clinton'.

Build a KeyGraph

- Represent each keyword as a node (preprocess and filter out nodes with low term frequency, document frequency and IDF). Create an edge if they occur together (again filter out noisy edges)

Event Detection and Tracking in Social Streams – Approach

• In order to support overlapping keywords – duplicate the nodes with high conditional probability on the edges.
• Delete the edges with high betweenness centrality
• The connected components represent a document of the event
• Document clustering is employed and documents with high variance are considered events – each event will talk about broad number of smaller category
• Use document overlap to merge similar events.

• A very basic approach
• Does not include Twitter specific features
• Cannot deal with spam events
• Not scalable
  • Approximate algorithms will be employed to find out the betweenness centrality for large graphs
Timeline

- Event Detection and Tracking in Social Streams
- **TwitterMonitor**: Trend Detection over Twitter Stream
- En Blogue – Emergent Topic Detection in Web 2.0 Streams
- Streaming First Story Detection with application to Twitter
- Bieber no more: First Story Detection using Twitter and Wikipedia
- Extraction and Compilation of Events and Sub-events from Twitter
- Twevent: a segment based event detection from Tweets
- Open Domain Event Extraction from Twitter
- Structured Event Retrieval over Microblog Archives
- Discover Breaking Events with Popular Hashtags in Twitter
- Social event detection with interaction graph modeling.
TwitterMonitor: Trend Detection over Twitter Stream

Michael Mathioudakis, Nick Koudas

SIGMOD 2010
Twitter Monitor

• Trend Detection in real-time
• Analysis of trend

Architecture of the system
Trend Detection

Bursty Keywords Detection

Group Bursty Keywords
Trend Detection: Bursty Keywords Detection

Idea: A keyword is identified as bursty when it is encountered at an unusually high rate in the stream. Whenever a keyword exhibits bursty behavior, TwitterMonitor considers this an indication that a new topic has emerged.

New algorithm: QueueBurst

- One-pass
- Real-time
- Adjustable against spurious bursts
- Adjustable against spam
Trend Detection: Group Bursty Keywords

To group bursty keywords, GroupBurst assesses their co-occurrences in recent tweets

GroupBurst pursues a greedy strategy that produces groups in a small number of steps
Trend Analysis

Identify more keywords associated with a trend

- Employs context extraction algorithms (such as PCA, SVD, etc) over the recent history of the trend and reports the keywords that are most correlated with it.
- Uses Grapevine’s entity extractor to identify frequently mentioned entities in trends.

Identify frequently cited sources and adds them to the trend description

Identify frequent geographical origins of tweets that belong to a trend
Timeline

• Event Detection and Tracking in Social Streams
• TwitterMonitor: Trend Detection over Twitter Stream
• En Blogue – Emergent Topic Detection in Web 2.0 Streams
• Streaming First Story Detection with application to Twitter
• Bieber no more: First Story Detection using Twitter and Wikipedia
• Extraction and Compilation of Events and Sub-events from Twitter
• Twevent: a segment based event detection from Tweets
• Open Domain Event Extraction from Twitter
• Structured Event Retrieval over Microblog Archives
• Discover Breaking Events with Popular Hashtags in Twitter
• Social event detection with interaction graph modeling.
En Blogue – Emergent Topic Detection in Web 2.0 Streams

Foteini Alvanaki, Sebastian Michel, Krithi Ramamritham, Gerhard Weikum

SIGMOD 2011
Motivation

System to notify users of emerging topics

Uses correlations between tags/entities from news, blogs, tweets, etc.

How is it different from TwitterMonitor?

• Instead of looking solely for bursty tags, the system detects shifts in tag correlations as they dynamically arise
Observations:

- Peaks in the popular tags have no influence on the size of overlap.
- Size of overlap grows dramatically, which cannot be explained by looking at the individual frequencies of $t_1$ and $t_2$ alone.
Approach

Seed Tag Selection

• Choose seed tags as popular tags. Use seed tags to generate candidate topics (pairs of tags that contain at least one seed tag)

Correlation Tracking

• For each candidate pair, track their correlations; continuously monitor amount of documents annotated with both tags

Shift Detection

• Define sliding window over correlation values. Whenever window is advances, calculate the correlation changes for all tag pairs. Consider sudden increase (shifts) as indicator for emergent topic
Entity Tagging

Why is it necessary?

Automatic entity tagging to extend the tag space with named entities like people, organizations or places.

How does it work?

When a new documents arrives, take a sliding window of 4 terms and check whether any substring of it match with a Wikipedia article title or a word in an ontology (like YAGO)
Timeline

- Event Detection and Tracking in Social Streams
- TwitterMonitor: Trend Detection over Twitter Stream
- En Blogue – Emergent Topic Detection in Web 2.0 Streams
- **Streaming First Story Detection with application to Twitter**
- Bieber no more: First Story Detection using Twitter and Wikipedia
- Extraction and Compilation of Events and Sub-events from Twitter
- Twevent: a segment based event detection from Tweets
- Open Domain Event Extraction from Twitter
- Structured Event Retrieval over Microblog Archives
- Discover Breaking Events with Popular Hashtags in Twitter
- Social event detection with interaction graph modeling.
Streaming First Story Detection with application to Twitter

Sasa Petrovic, Miles Osborne, Victor Lavrenko

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL
Background

Given a sequence of stories, the goal of FSD is to identify the first story to discuss a particular event.
The traditional approach to FSD, where each new story is compared to all, or a constantly growing subset, of previously seen stories, does not scale to the Twitter streaming setting.

Need:

- **Constant processing time** - Locality Sensitive Hashing
- **Constant space** – keeping number of stories in memory constant
Modified LSH

Why modification?
- Simply applying pure LSH in a FSD task yields poor performance and a high variance in results
- If the nearest neighbour is far away from all other points, LSH fails to find the true near neighbour

What modification?
- Limit the search to a small number of most recent documents
- However, because there is only a finite number of buckets, in a true streaming setting
  - we would use an unbounded amount of space
  - the number of comparisons we need to make would also grow without a bound
Modified LSH

Limit the number of documents inside a single bucket to a constant. If the bucket is full, the oldest document in the bucket is removed.

Limit to making a constant number of comparisons by comparing each new document with at most $3L$ documents it collided with most frequency.
How to know if a story is about a real event or not?

• Majority of tweets are not real stories, but rather updates on one’s personal life, conversations, or spam - most of which would be of no interest to anyone but a few people

• Approach:
  • Assign a novelty score to each tweet
  • Since the score is based on a cosine distance to the nearest tweet, for each tweet, output which other tweet it is most similar to. This way, we can analyse threads of tweets (a subset of tweets which all discuss the same topic)
  • For each time interval we only output the fastest growing threads.
  • Growth rate also gives us a way to measure a thread’s impact
Timeline

• Event Detection and Tracking in Social Streams
• TwitterMonitor: Trend Detection over Twitter Stream
• En Blogue – Emergent Topic Detection in Web 2.0 Streams
• Streaming First Story Detection with application to Twitter
• Bieber no more: First Story Detection using Twitter and Wikipedia
• Extraction and Compilation of Events and Sub-events from Twitter
• Twevent: a segment based event detection from Tweets
• Open Domain Event Extraction from Twitter
• Structured Event Retrieval over Microblog Archives
• Discover Breaking Events with Popular Hashtags in Twitter
• Social event detection with interaction graph modeling.
Bieber no more: First Story Detection using Twitter and Wikipedia

Miles Osborne and Saša Petrović, Richard McCreadie, Craig Macdonald, and Iadh Ounis

TAIA 2012
Motivation

To analyse whether Wikipedia can be used to improve the quality of discovered events in Twitter
FSD from Twitter

For each incoming tweet, it is compared against the stream of previously seen tweets using a fast hashing strategy.

If current tweet is sufficiently dissimilar from its nearest neighbour, we it is a potential First Story.
Identifying event-related Wikipedia Pages

Events are reflected in terms spikes in page views in Wikipedia

Identify a stream of outliers pages that correspond to abnormally large page views
Multi-stream FSD

Combine two streams as follows:

For each event identified from Twitter, find the closest Wikipedia page, using simple vector space model.

Ideally, false stories should have no matches in the Wikipedia stream.

For genuine First Stories, it should match (at least partially) with Wikipedia titles.
Conclusion

Wikipedia lags Twitter by around two hours.

Hence for truly real-time event detection, the usefulness of Wikipedia as a filter is limited.

For situation where the lag is acceptable, Wikipedia is observed to substantially improve quality of events detected from Twitter.
Timeline

- Event Detection and Tracking in Social Streams
- TwitterMonitor: Trend Detection over Twitter Stream
- En Blogue – Emergent Topic Detection in Web 2.0 Streams
- Streaming First Story Detection with application to Twitter
- Bieber no more: First Story Detection using Twitter and Wikipedia
- Extraction and Compilation of Events and Sub-events from Twitter
- Twevent: a segment based event detection from Tweets
- Open Domain Event Extraction from Twitter
- Structured Event Retrieval over Microblog Archives
- Discover Breaking Events with Popular Hashtags in Twitter
- Social event detection with interaction graph modeling.
Extraction and Compilation of Events and Sub-events from Twitter

Khurdiya, Arpit, et al.

Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology
Extraction and Compilation of Events and Sub-events from Twitter - *Idea*

Given a big event, identify all the sub-events revolving around it.

Extract the event components like actor, action and context.

Build an event map to elucidate the relationship between events and sub-events.

Extraction and Compilation of Events and Sub-events from Twitter - Approach

Extraction and Compilation of Events and Sub-events from Twitter – Approach

Extraction and Compilation of Events and Sub-events from Twitter – Analysis

Subjects and activities are identified with a higher accuracy than context and objects.
• How do we track stories which have context in common?
• Jaguar launched! Jaguars under threat.

Scalability of CRFs
• The slow convergence of CRF can be a bottleneck for creating real time event maps

Can we predict sub events?
• Given a partial event map, is it possible to complete the map- applications in event prediction, archaeology.

Will including user interests help?
• Given an event corresponding to a topic, will using the demographic information of the users interested in those topics help in revealing deeper semantic?
Timeline

• Event Detection and Tracking in Social Streams
• TwitterMonitor: Trend Detection over Twitter Stream
• En Blogue – Emergent Topic Detection in Web 2.0 Streams
• Streaming First Story Detection with application to Twitter
• Bieber no more: First Story Detection using Twitter and Wikipedia
• Extraction and Compilation of Events and Sub-events from Twitter
• **Twevent: a segment based event detection from Tweets**
• Open Domain Event Extraction from Twitter
• Structured Event Retrieval over Microblog Archives
• Discover Breaking Events with Popular Hashtags in Twitter
• Social event detection with interaction graph modeling.
Twevent: a segment based event detection from Tweets

Li, Chenliang, Aixin Sun, and Anwitaman Datta

Twevent: a segment based event detection from Tweets - Idea

Detect bursty segments as event segments

Cluster the event segments into candidate events using content similarity and frequency distribution

Wikipedia is exploited to identify the realistic events and to derive the most newsworthy segments to describe the identified events.

Twevent: a segment based event detection from Tweets – Approach

Tweet Segmentation → Event Segment Detection → Event Segment Clustering

Twevent: a segment based event detection from Tweets – Approach

Tweet Segmentation → Event Segment Detection → Event Segment Clustering

TWEET SEGMENTATION

Given a tweet, split it into $m$ non-overlapping and consecutive segments, where each segment is either a unigram or a multi-gram.

An optimization problem where $C$ is the function that measures the stickiness of a segment of a tweet

$$\arg \max_{s_1, \ldots, s_m} C(d) = \sum_{i=1}^{m} C(s_i)$$

$$C(s) = L(s) \cdot e^{Q(s)} \cdot S(SCP(s))$$

- $SCP(s)$ : Symmetric Conditional Probability
- $Q(s)$ : Anchor Text in Wikipedia Article that contains $s$
- $L(s)$ : Moderate preference for longer segments

Twevent: a segment based event detection from Tweets – Approach

Tweet Segmentation → Event Segment Detection → Event Segment Clustering

EVENT SEGMENT DETECTION

t : time window
N_t : number of tweets published within time-window
f_{s,t} : number of tweets containing s published within t

The probability of observing frequency f_{s,t} of segment s in t can be modelled by:

\[ P(f_{s,t}) = \binom{N_t}{f_{s,t}} p_s^{f_{s,t}} (1 - p_s)^{N_t - f_{s,t}} \]

As N_t is large we can model it as a Gaussian Distribution

\[ P(f_{s,t}) \sim \mathcal{N}(N_t p_s, N_t p_s(1 - p_s)). \]

A segment s is a bursty segment in time window t if its tweet frequency f_{s,t} > E[s|t]

The bursty probability is given by:

\[ P_b(s, t) = S \left( 10 \times \frac{f_{s,t} - (E[s|t] + \sigma[s|t])}{\sigma[s|t]} \right) \]

User Frequency \((u_{s,t})\): Number of users who post tweets containing \(s\) during the time period \(t\).

Avoid Spam:

\[ w_b(s, t) = P_b(s, t) \log(u_{s,t}) \]

A bursty segment is a potential event segment ranked among top-K bursty segments by \(w_b(s, t)\) in descending order, where \(K = \sqrt{N_t}\)

---

Twevent: a segment based event detection from Tweets – Approach

Tweet Segmentation → Event Segment Detection → Event Segment Clustering

EVENT SEGMENT CLUSTERING

- Event Segment Similarity
  \[ sim_t(s_a, s_b) = \sum_{m=1}^{M} w_t(s_a, m)w_t(s_b, m)sim(T_t(s_a, m), T_t(s_b, m)) \]

- Event Clustering
  - Use a variant of Jarvis Patrick, k-nearest neighbour clustering algorithm

- Candidate Event Filtering
  - Segment Newsworthiness:
    \[ \mu(s) = \max_{e \in s} e^{Q(e)} - 1 \]
  - Event Newsworthiness:
    \[ \mu(e) = \frac{\sum_{s \in e_s} \mu(s)}{|e_s|} \cdot \frac{\sum_{g \in E_e} sim(g)}{|e_s|} \]
  - After filtering away noisy events, represent each detected event with its member event segments sorted by newsworthiness scores.

## Twevent: a segment based event detection from Tweets – Analysis

<table>
<thead>
<tr>
<th>Works well on real time data</th>
<th>• the time complexity is (O(\sqrt{N}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does not incorporate twitter features</td>
<td>• Will the performance enhance on incorporating hashtags and retweets?</td>
</tr>
<tr>
<td>Wikipedia Entites are not recent</td>
<td>• Wikipedia might not have a page on AAP when the party came into picture. The system fails to identify such events</td>
</tr>
</tbody>
</table>

Timeline

- Event Detection and Tracking in Social Streams
- TwitterMonitor: Trend Detection over Twitter Stream
- En Blogue – Emergent Topic Detection in Web 2.0 Streams
- Streaming First Story Detection with application to Twitter
- Bieber no more: First Story Detection using Twitter and Wikipedia
- Extraction and Compilation of Events and Sub-events from Twitter
- Twevent: a segment based event detection from Tweets
- Open Domain Event Extraction from Twitter
- Structured Event Retrieval over Microblog Archives
- Discover Breaking Events with Popular Hashtags in Twitter
- Social event detection with interaction graph modeling.
Open Domain Event Extraction from Twitter

Ritter, Alan, Oren Etzioni, and Sam Clark

Open Domain Event Extraction from Twitter – 
Idea and Approach

Extraction of Events

Classification of events in the categories using latent variable models

Significance Ranking

Open Domain Event Extraction from Twitter

-Extraction of Events

• Annotate a corpus of tweets for sequence labelling – use linear chain CRF

• Features used:
  • Contextual Features
  • Orthographic Features
  • Twitter tuned POS tagger features
  • Dictionary of event terms

Open Domain Event Extraction from Twitter – Idea and Approach

Extraction of Events

Classification of events in the categories using latent variable models

Significance Ranking

Open Domain Event Extraction from Twitter
-Classification of Event Types

• Temporal expressions are resolves using Tempex

• Supervised or Semisupervised Approaches won’t work
  • A priori not known categories are appropriate for Twitter.
  • Manual annotation is expensive
  • The important categories/ entities shift with time
  • Many important categories are relatively infrequent

• Adopt a latent variable model approach
  • Each event indicator phrase in our data, e, is modeled as a mixture of types.
  • Preserves ambiguity

Open Domain Event Extraction from Twitter – Idea and Approach

Extraction of Events → Classification of events in the categories using latent variable models → Significance Ranking

Open Domain Event Extraction from Twitter

-Ranking of Events

• Cannot use frequency - many tweets refer to common events in user's daily lives.

• Important events - those which have strong association with a unique date as opposed to being spread evenly across days on the calendar.

• Use G-square log likelihood ratio to estimate the association between event and its date

Open Domain Event Extraction from Twitter - Analysis

• Scalable
• Open Domain
• Latent variable Model
• Current state of the art for event extraction
• Limited to English

Timeline

• Event Detection and Tracking in Social Streams
• TwitterMonitor: Trend Detection over Twitter Stream
• En Blogue – Emergent Topic Detection in Web 2.0 Streams
• Streaming First Story Detection with application to Twitter
• Bieber no more: First Story Detection using Twitter and Wikipedia
• Twevent: a segment based event detection from Tweets
• Extraction and Compilation of Events and Sub-events from Twitter
• Open Domain Event Extraction from Twitter
• **Structured Event Retrieval over Microblog Archives**
• Discover Breaking Events with Popular Hashtags in Twitter
• Social event detection with interaction graph modeling.
Structured Event Retrieval over Microblog Archives

Donald Metzler, Congxing Cai, Eduard Hovy

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 2012
Motivation

- **Historical** event information.
- **Ranked list** of historical event summaries.
- Viewed **through the lens of users** who experienced or discussed the event while it happened.
- Not only be useful for everyday end-users, but also for **social scientists, historians, journalists, and emergency planners**.
Overview of Framework

Given a query that specifies an event, task is to retrieve a set of relevant structured event representations from a large archive of microblog messages.

Two steps

• Timespan retrieval
• Summarization

The result consists of a **start time**, a **duration**, and a small number of **messages** posted during the time interval that summarizes the event.
Temporal Query Expansion

Vocabulary Mismatch problem

Propose a novel unsupervised query expansion technique based on temporal co-occurrence of terms

Given keyword query q, automatically retrieve N timespans for which the query keywords were most heavily discussed. Pseudo-relevant.
Temporal Query Expansion

• For each pseudo-relevant timespan, a burstiness score is computed for all of the terms that occur in messages posted during the timespan.

\[
\text{burstiness}(w,TS_i) = P(w|TS_i) P(w)
\]

• Finally, aggregate the burstiness scores across all pseudo-relevant timespans to generate an over-all score for each term. Use geometric mean

• The k highest weighted terms are then used as expansion terms
## Timespan Ranking

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use the expanded query $q'$ to retrieve relevant timespans</td>
<td></td>
</tr>
<tr>
<td>Scoring function to find relevance of timespan to query $q'$ – using frequency of expansion terms during that timespan</td>
<td></td>
</tr>
<tr>
<td>Identify the 1000 highest scoring timespans with respect to $q'$</td>
<td></td>
</tr>
</tbody>
</table>
Timespan Summarization

• Scoring a microblog message $M$ with respect to an expanded query representation $q'$

$$s(q', M) = \sum_{w \in q'} \beta_w \cdot \log P(w|M)$$

where $\beta_w$ is the burstiness score for expansion term $w$ and $P(w|M)$ is a Dirichlet smoothed language modeling estimate for term $w$ in message $M$
Results

• Retrieval performance varies substantially across the different event type categories

• Mainly due to difference in their characteristics like frequency of occurrence, geographic or demographic interest etc.

• Twitter users are more interested in events related to entertainment, politics, and dramatic crisis related topics, including natural disasters and terrorist attacks.

• Less interested in events related to energy or transportation.
Results

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Duration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>July 16 2010 at 17 UTC</td>
<td>11 h</td>
<td></td>
<td>Ok a 3.6 “rocks” nothing. But boarding a plane there now, Woodward ho! RT @todayshow: 3.6 magnitude #earthquake rocks Washington DC area.</td>
</tr>
<tr>
<td>September 28 2010 at 11 UTC</td>
<td>6 h</td>
<td></td>
<td>RT @Quakeprediction: 2.6 earthquake (possible foreshock) hits E of Los Angeles; <a href="http://earthquake.usgs.gov/earthquakes/recenteqscanv/Fau">http://earthquake.usgs.gov/earthquakes/recenteqscanv/Fau</a>...</td>
</tr>
<tr>
<td>September 04 2010 at 01 UTC</td>
<td>3 h</td>
<td></td>
<td>7.0 quake strikes New Zealand - A 7.0-magnitude earthquake has struck near New Zealand’s second largest city. Reside... <a href="http://ht.ly/18R2rw">http://ht.ly/18R2rw</a></td>
</tr>
<tr>
<td>October 27 2010 at 01 UTC</td>
<td>5 h</td>
<td></td>
<td>RT @SURFER_Magazine: Tsunami Strikes Mentawai: Wave Spawned By A 7.5-Magnitude Earthquake Off West Coast Of Indonesia <a href="http://bit.ly/8Z9Lbv">http://bit.ly/8Z9Lbv</a></td>
</tr>
</tbody>
</table>
Timeline

• Event Detection and Tracking in Social Streams
• TwitterMonitor: Trend Detection over Twitter Stream
• En Blogue – Emergent Topic Detection in Web 2.0 Streams
• Streaming First Story Detection with application to Twitter
• Bieber no more: First Story Detection using Twitter and Wikipedia
• Twevent: a segment based event detection from Tweets
• Extraction and Compilation of Events and Sub-events from Twitter
• Open Domain Event Extraction from Twitter
• Structured Event Retrieval over Microblog Archives
• Discover Breaking Events with Popular Hashtags in Twitter
• Social event detection with interaction graph modeling.
Discover Breaking Events with Popular Hashtags in Twitter

Anqi Cui, Min Zhang, Yiqun Liu, Shaoping Ma, Kuo Zhang

CIKM 2012
Hashtags are typically used to mark keywords or topics in a tweet. Hence they provide useful metadata for the topic and annotation of the tweet.

Acts as indicator of a topic in a large number of tweets. The bursting of hashtags reflects the bursting of events.
Challenges

A specific hashtag may refer to different objects

Different hashtags may describe a same event

Many of the trending topics are Twitter Memes: conversational topics that attract users to share their own personal feelings.
Hashtag Attributes and Categorization

- Hashtag Instability
- Twitter Meme Possibility
- Authorship Entropy
Hashtag Instability

Popular hashtags are associated with events that most people concern with – either breaking events or persistent discussions.

- #sopa, #ff, #nowplaying

Hashtags with respect to breaking events have unexpected changes (mainly increasing) of frequencies.

Hence the concept of instability, i.e., how likely the hashtag has a sudden increase or decrease.
Hashtag Instability

• Assume a random variable $X$ as the frequency of a hashtag $H$ in a time period, with a Gaussian distribution.

• The frequency is modeled around a mean $\mu$, thus the probability $P$ of $x$ away from $\mu$ is computed as

$$\tilde{P}(x) = Pr(X > x \lor X < 2\mu - x)$$

• We aim to discover the most instable situations, hence we focus on the $x$’s whose $P(x) < \rho$.

$$Inst(x) = -\log \tilde{P}(x), Inst(H) = \frac{1}{n} \sum_{\tilde{P}(x) < \rho} Inst(x)$$
Twitter Meme Possibility

Some of the good measurements for the possibility

- Word length ratio - the number of real English words $N$ divided by the length of the hashtag $L$
- the probability of its appearing at the beginning of a tweet

A dynamic programming method for word splitting is applied to obtain the minimal number of words $N$ in the hashtag

Consecutive letters not in the dictionary are split into individual letters.
Twitter Meme Possibility

This measurement generates a probability \( P_{\text{word}} = 1 - \frac{N}{L} \)

For hashtag’s position, we make an estimation on the sampled tweet set, i.e., given a hashtag \( h \)

\[
p_{\text{pos}} = \frac{|\{\text{tweets starting with } h\}|}{|\{\text{tweets containing } h\}|}
\]

Finally, \( \text{TMP}(\text{hashtag}) = p_{\text{pos}} \cdot P_{\text{word}} \)
Authorship Entropy

• Some of the popular hashtags are contributed by only a few authors. These “top” authors contribute more than half of the tweets containing a certain hashtag.

• Many of these users are spammers or news publishers.

• Hence, authorship of hashtags is helpful for detecting automated agents and robots.

• Taking all the $n$ tweets containing a hashtag, the $k$ authors of these $n$ tweets contribute $C_1, C_2, ..., C_k$ ($\sum C_i = n$) times, respectively. The authorship entropy is:

$$Ent(\text{hashtag}) = - \sum_{i=1}^{k} \frac{C_i}{n} \cdot \log \left( \frac{C_i}{n} \right)$$
Categorization

- The three features described are orthogonal dimensions which are independent from each other.

- Considering each feature a lower or a higher value, the hashtag space is divided into eight subspaces

<table>
<thead>
<tr>
<th>Inst.</th>
<th>TMP</th>
<th>Ent.</th>
<th>Cat.</th>
<th>Inst.</th>
<th>TMP</th>
<th>Ent.</th>
<th>Cat.</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>L</td>
<td>L</td>
<td>A</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>A</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>H</td>
<td>M</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>B</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>L</td>
<td>A</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>A</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>H</td>
<td>T</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>T</td>
</tr>
</tbody>
</table>

L = Low, H = High, A = Advertisements, M = Miscellaneous, T = Twitter Memes, B = Breaking Events
Timeline

- Event Detection and Tracking in Social Streams
- TwitterMonitor: Trend Detection over Twitter Stream
- En Blogue – Emergent Topic Detection in Web 2.0 Streams
- Streaming First Story Detection with application to Twitter
- Bieber no more: First Story Detection using Twitter and Wikipedia
- Twevent: a segment based event detection from Tweets
- Extraction and Compilation of Events and Sub-events from Twitter
- Open Domain Event Extraction from Twitter
- Structured Event Retrieval over Microblog Archives
- TAER: Time-Aware Entity Retrieval
- Discover Breaking Events with Popular Hashtags in Twitter
- Social event detection with interaction graph modeling.
Social event detection with interaction graph modeling

Wang, Yanxiang, Hari Sundaram, and Lexing Xie.

Social Event Detection with Interaction Graph Modeling – Idea

• Detecting social, physical-world events from photos posted on social media sites.

• Incorporate social interaction along with social affinity of the photos to detect events in addition to photo metadata including time, location, tags and description

Social Event Detection with Interaction Graph Modeling – *Idea*

- **Determination of Similarity between photos** - use a random-walk model.
- **Social interaction graph construction** - based on the relationship amongst users, tags and photos.
- **Event Detection** - Train a SVM on features and later use an incremental clustering algorithm.

Social Event Detection with Interaction Graph Modeling – Idea

Determination of Similarity between photos - use a random-walk model.

Social interaction graph construction - based on the relationship amongst users, tags and photos.

Event Detection - Train a SVM on features and later use an incremental clustering algorithm.

Determination of Similarity between photos

• Flickr API exposes several “social” features for each photo - Photo comment, Photo favourite, Photo tag, Related tags, Owner contact list

• Jaccard Set Similarity
  • demphasize a spam-like commenter
  • Incorporate both user and tag set

• Random Walk Model
  • User is interested to browse a photo p in an event E
    • The user can either click on the tags associated with that photo
    • The user can check the photos of the users who commented on the photo
  • Removes spam commenters and spam tags
Social Event Detection with Interaction Graph Modeling – *Idea*

Determination of Similarity between photos - use a random-walk model.

Social interaction graph construction - based on the relationship amongst users, tags and photos.

Event Detection - Train a SVM on features and later use an incremental clustering algorithm.

---

Social Interaction Graph

• Represent the social interaction around photos using a undirected graph with three conceptual layers: photos, users and tags.

• Symmetric Edges:
  • Between a photo and a tag associated with it.
  • Connect related tags returned by the Flickr API that provides related tags.
  • Between a photo and each user who interacts with the photo.

• Asymmetric Edges:
  • Relationship between two users.
Random Walk with Restart algorithm

- The sequence of steps to compute affinity between a pair of photos \((v_s; v_e)\) are as follows:
  - Build the social interaction graph, resulting in the corresponding adjacency matrix \(A\).
  - Create a restart vector \(v\) where all entries are set to zero except for the entry for \(v_s\), which is set to 1.
  - Calculate the steady state vector \(v_{ss}\) using the following formula
    \[
    v_{ss} = \frac{1}{c (I - (1 - c)A)} v.
    \]
    The entry in the steady state vector corresponding to node \(v_e\) will give us—after normalization—the social affinity score between photos \(v_s\) and \(v_e\).
Social Event Detection with Interaction Graph Modeling – Idea

Determination of Similarity between photos - use a random-walk model.

Social interaction graph construction - based on the relationship amongst users, tags and photos.

Event Detection - Train a SVM on features and later use an incremental clustering algorithm.

Event Detection

• Apart from social similarity feature include photo features like Timestamp, Location, Tags, Text Description

• Use and SVM classifier to combine these features into a single feature

• Apply incremental clustering to group the pictures in clusters.
  • For each image p, given a cluster, we calculate the average similarity of the image p with each image in the cluster. Assign p to the cluster with highest similarity.
Social Event Detection with Interaction Graph Modeling – Analysis

- Demonstrate utility of social features on baseline photo features
- Recently, twitter has done significant changes in its image sharing options
- An image is retweeted and favourited faster than tweet.
- Inspired from the work done by Nikhil Rasiwasia, attempts Text to Image search and vice versa.
Conclusion

• We did a thorough analysis of the work done in social media.

• Several techniques are employed:
  • Keygraph
  • Hashtag
  • CRFs

• We also saw application of event detection in structured event retrieval

• Event detection can also be done on images

• Future Prospects:
  • Predicting events
  • Deeper understanding of user generated content – not limited to text, rather exploiting the media – images, videos etc.