

Towards a Social Media Analytics Platform: Event Detection and User Profiling for Microblogs

Manish Gupta



Rui Li



Kevin Chang



April 8, 2014



gmanish@microsoft.com, ruililab@yahoo-inc.com, kcchang@illinois.edu

Characteristics of Twitter Data

- 140 characters – short documents
- SMS kind of language
- Code mixing (mix of multiple languages)
- Tweets, Retweets, Mentions, Hashtags
- Very fresh news from human sensors
- Large amount of data with huge data rate
 - Many irrelevant messages
 - Many redundant messages
- Self-contained
- Simple discourse structure

Noisy Twitter Text: Challenges

- Lexical Variation (misspellings, abbreviations)
 - `2m', `2ma', `2mar', `2mara', `2maro', `2marrow', `2mor', `2mora', `2moro', `2morow', `2morr', `2morro', `2morrow', `2moz', `2mr', `2mro', `2mrrw', `2mrw', `2mw', `tmmrw', `tmo', `tmoro', `tmorrow', `tmoz', `tmr', `tmro', `tmrow', `tmrrow', `tmrrw', `tmrw', `tmrww', `tmw', `tomaro', `tomarow', `tomarro', `tomarrow', `tomm', `tommarow', `tommarrow', `tommoro', `tommorrow', `tommorrow', `tommorw', `tommrow', `tomo', `tomolo', `tomoro', `tomorow', `tomorro', `tomorrw', `tomoz', `tomrw', `tomz'
- Unreliable Capitalization
 - “The Hobbit has FINALLY started filming! I cannot wait!”
- Unique Grammar
 - “watchng american dad.”

NLP in News vs. Twitter: Thought Experiment

- **Task 1**
 - Read each sentence from today's New York times
 - Except, first randomly permute the sentences
 - Answer basic questions about today's news
- **Task 2**
 - Read a random sample of tweets
 - From high-quality sources
 - Order is picked randomly
 - Answer basic questions about today's news
- **Claim:**
 - Task 2 is easier than task 1.

Tutorial Overview

- Event Detection for Twitter (80 min)
 - Event Detection using Tweet Content
 - Event Detection using Other External Sources
 - Applications of Event Detection
- Break (10 min)
- Event Description for Twitter (25 min)
 - Finding Best Phrase to Summarize an Event
 - Finding Event Types
 - Finding Event Timespans
 - Finding Event Credibility
 - Finding Event Locations
- User Profiling in Social Media (55 min)
 - Content-based Profiling
 - Network-based Profiling
 - Hybrid Approach
 - Co-profiling Attributes and Relationships
- Summary and Discussions (10 min)

Tutorial Overview

- **Event Detection for Twitter (80 min)**
 - Event Detection using Tweet Content
 - Event Detection using Other External Sources
 - Applications of Event Detection
- Break (10 min)
- Event Description for Twitter (25 min)
 - Finding Best Phrase to Summarize an Event
 - Finding Event Types
 - Finding Event Timespans
 - Finding Event Credibility
 - Finding Event Locations
- User Profiling in Social Media (55 min)
 - Content-based Profiling
 - Network-based Profiling
 - Hybrid Approach
 - Co-profiling Attributes and Relationships
- Summary and Discussions (10 min)

Why Detect Events from Twitter?

- Twitter is a great news source
 - Human sensors report very quickly
 - Tweet waves travel faster than earthquake waves!
- Overload of information
 - Show only ranked important events
- Showing 10 relevant tweets is not a great idea, since very few real information needs can be satisfied by a single short piece of text
- Will look at applications of event detection later

Manual Event Detection

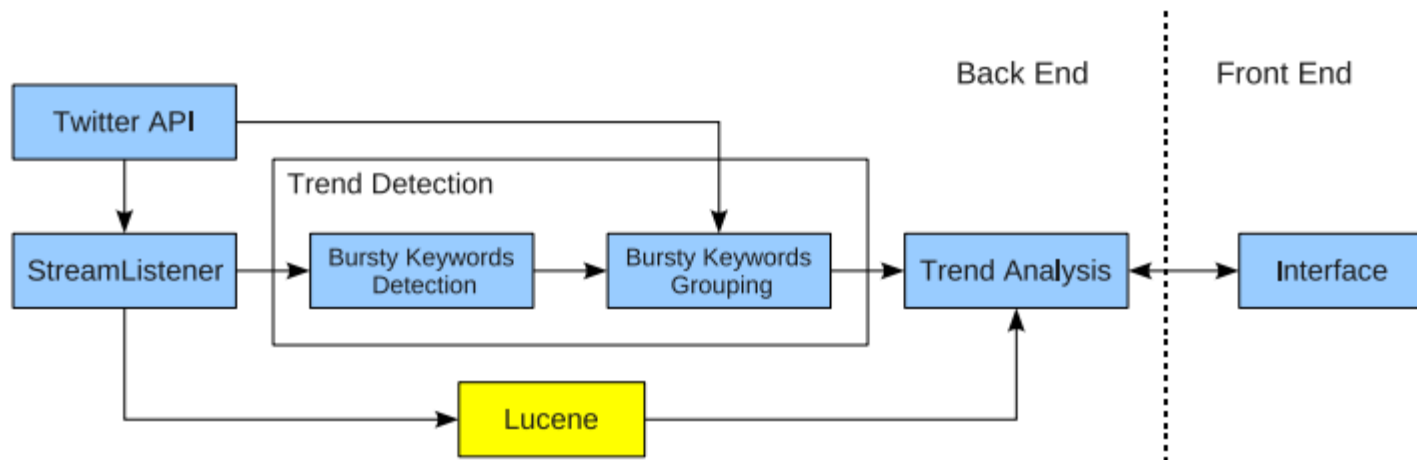
- Twitter partnered with the third-party website WhatTheTrend to provide definitions of trending topics
- WhatTheTrend allows users to manually enter descriptions of why a topic is trending
- Problems
 - Spam
 - Manual (significant efforts)
 - Time lag

Tutorial Overview

- Event Detection for Twitter (80 min)
 - Event Detection using Tweet Content
 - Event Detection using Other External Sources
 - Applications of Event Detection
- Break (10 min)
- Event Description for Twitter (25 min)
 - Finding Best Phrase to Summarize an Event
 - Finding Event Types
 - Finding Event Timespans
 - Finding Event Credibility
 - Finding Event Locations
- User Profiling in Social Media (55 min)
 - Content-based Profiling
 - Network-based Profiling
 - Hybrid Approach
 - Co-profiling Attributes and Relationships
- Summary and Discussions (10 min)

Twitter Monitor

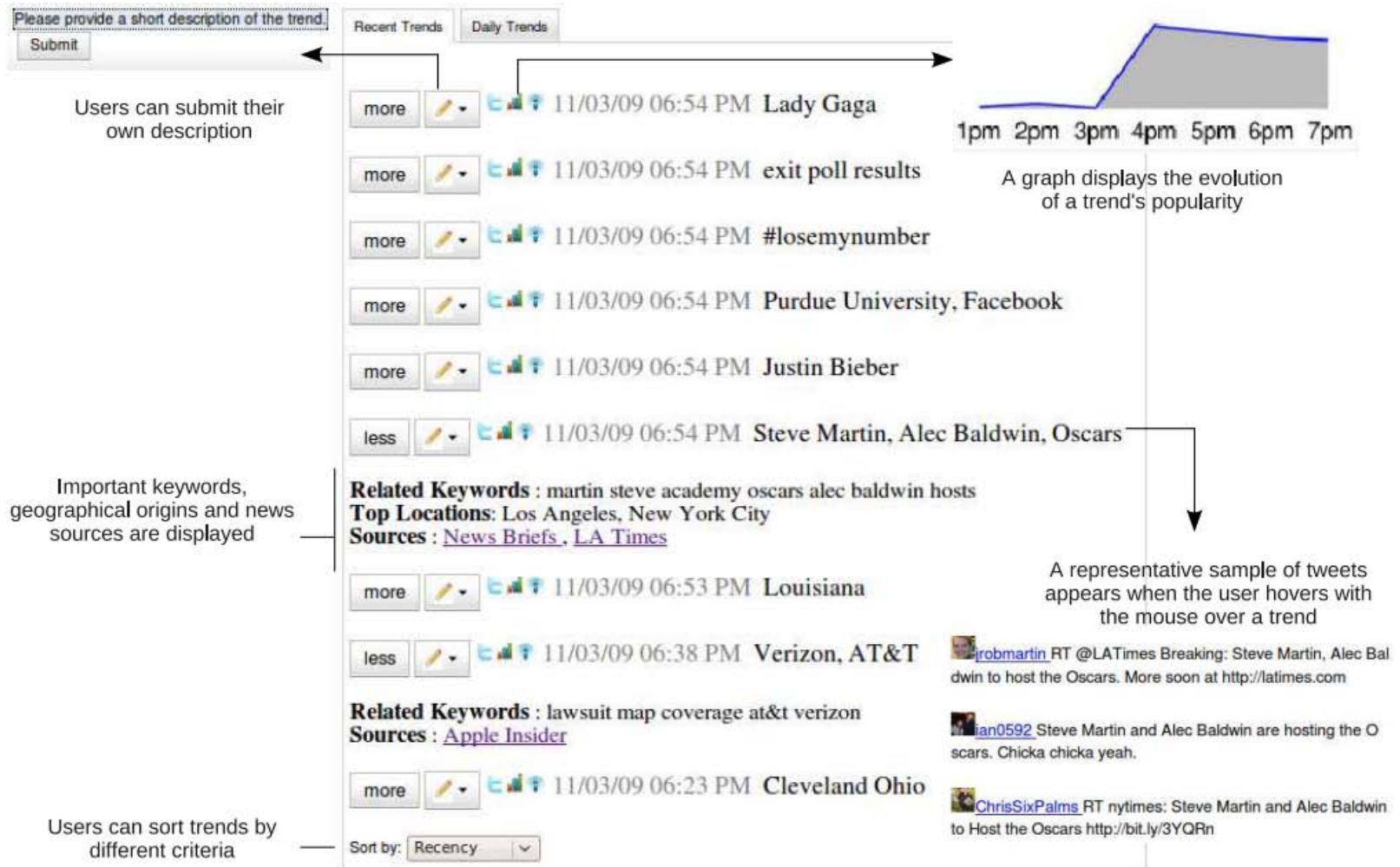
- Michael Mathioudakis and Nick Koudas. TwitterMonitor: trend detection over the twitter stream. SIGMOD '10
- Identifies 'bursty' keywords, i.e., keywords that suddenly appear in tweets at an unusually high rate.
- Groups bursty keywords into trends based on their co-occurrences.
- Extracts additional information from the tweets that belong to the trend, aiming to discover interesting aspects of it.



Twitter Monitor: Trend Analysis

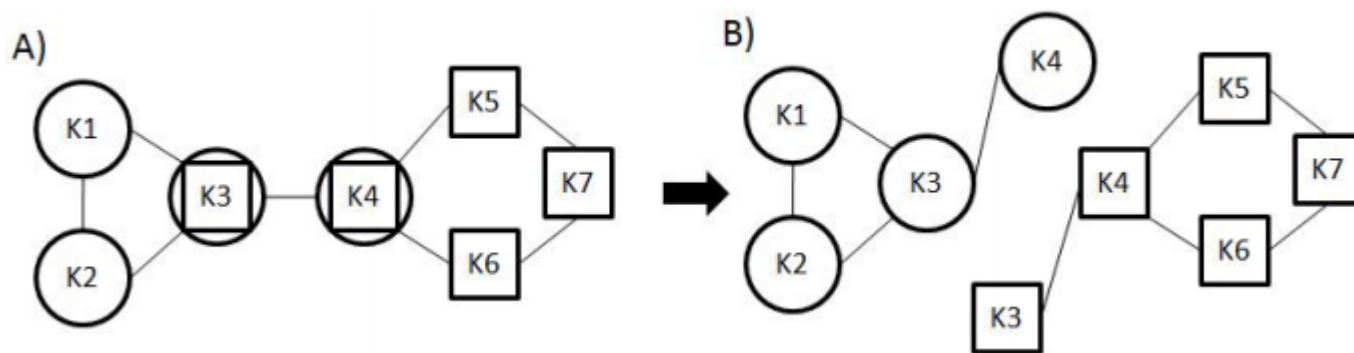
- Find bursty keywords
- Group bursty keywords based on their co-occurrences to get trends (keyword clusters)
- For every trend
 - Identify more keywords which may not be bursty but provide the context of the event
 - Using SVD
 - Identify frequently mentioned entities in tweets containing the trend keywords
 - Links in related tweets
 - Frequent geographical origins of related tweets

Twitter Monitor



Detecting Events using Graph Community Analysis

- Sayyadi, H., Hurst, M., and Maykov, A. (2009). Event Detection and Tracking in Social Streams. ICWSM.
- Extract names entities and noun phrases from tweets
- A KeyGraph is created where nodes are keywords and edges between the nodes are formed when those terms co-occur in a document
 - Nodes are created only if both TF and IDF are high for that node
 - Edge is created between two nodes k_i and k_j if co-occurrence prob is high, $p(k_i|k_j)$ is high and $p(k_j|k_i)$ is high
- Apply community analysis techniques to this graph to discover events (communities of keywords)
 - Remove edges with high betweenness centrality iteratively
 - A keyword can belong to more than 1 event
 - Before removing an edge, if the edge's conditional probability is high, edge and corresponding nodes are duplicated



Detecting Events using LSH (1)

- Saša Petrović, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to Twitter. HLT '10.
- Locality sensitive hashing can be used with the cosine similarity based distance function to compute nearest neighbor documents given a document d .
 - Cosine similarity based nearest neighbor finding can be done using hash functions which project the document vector onto a random hyperplane
 - Increasing the number of such hyperplanes (k) decreases the prob. of chance collisions.
 - But that also decreases prob. of colliding with nearest neighbor.
 - Hence maintain multiple (L) hash tables.

Detecting Events using LSH (2)

- LSH has a problem
 - If the nearest neighbor is far away, LSH does not work
 - So, if the minimum distance of new document d with all documents in colliding bucket (as per LSH) is higher than a threshold, check distance of d with most recent 1000 documents and update min distance if needed.
- Two improvements
 - In each hash table, maintain a constant number of documents per bucket. Remove old documents
 - On collision with buckets in L hash tables, don't compare with all documents in all L hash tables. Instead compare to the 3L documents that collide most frequently with the new document.


Algorithm 2: Our LSH-based approach.

```
input: threshold  $t$ 
1 foreach document  $d$  in corpus do
2   add  $d$  to LSH
3    $S \leftarrow$  set of points that collide with  $d$  in LSH
4    $dis_{min}(d) \leftarrow 1$ 
5   foreach document  $d'$  in  $S$  do
6      $c = \text{distance}(d, d')$ 
7     if  $c < dis_{min}(d)$  then
8        $dis_{min}(d) \leftarrow c$ 
9     end
10  end
11  if  $dis_{min}(d) \geq t$  then
12    compare  $d$  to a fixed number of most
      recent documents as in Algorithm 1 and
      update  $dis_{min}$  if necessary
13  end
14  assign score  $dis_{min}(d)$  to  $d$ 
15  add  $d$  to inverted index
16 end
```

Detecting Events using LSH (3)

- Specifically for Twitter
 - Tweet a links to tweet b if b is the nearest neighbor of a and $1 - \cos(a, b) < t$, where t is a user-specified threshold
 - Then, for each tweet a we either assign it to an existing thread if its nearest neighbor is within distance t , or say that a is the first tweet in a new thread.
 - Once we have threads of tweets, we are interested in threads which grow fastest, as this will be an indication that news of a new event is spreading. Therefore, for each time interval we only output the fastest growing threads. This growth rate also gives us a way to measure a thread's impact.

Detecting Events using CRFs (1)

- Given a repository of tweets, first find named entities using a CRF-based NER on tweets
- Next, find entity-referring phrases.
- Useful to display in connection with events
 - E.g. “**Steve Jobs**” +  + “**October 6**”
- Helpful in categorizing Events into Types
- Examples
 - Apple to Announce iPhone 5 on October 4th! YES!
 - iPhone 5 announcement coming Oct 4th
 - WOOOHOO NEW IPHONE TODAY! CAN'T WAIT!

Detecting Events using CRFs

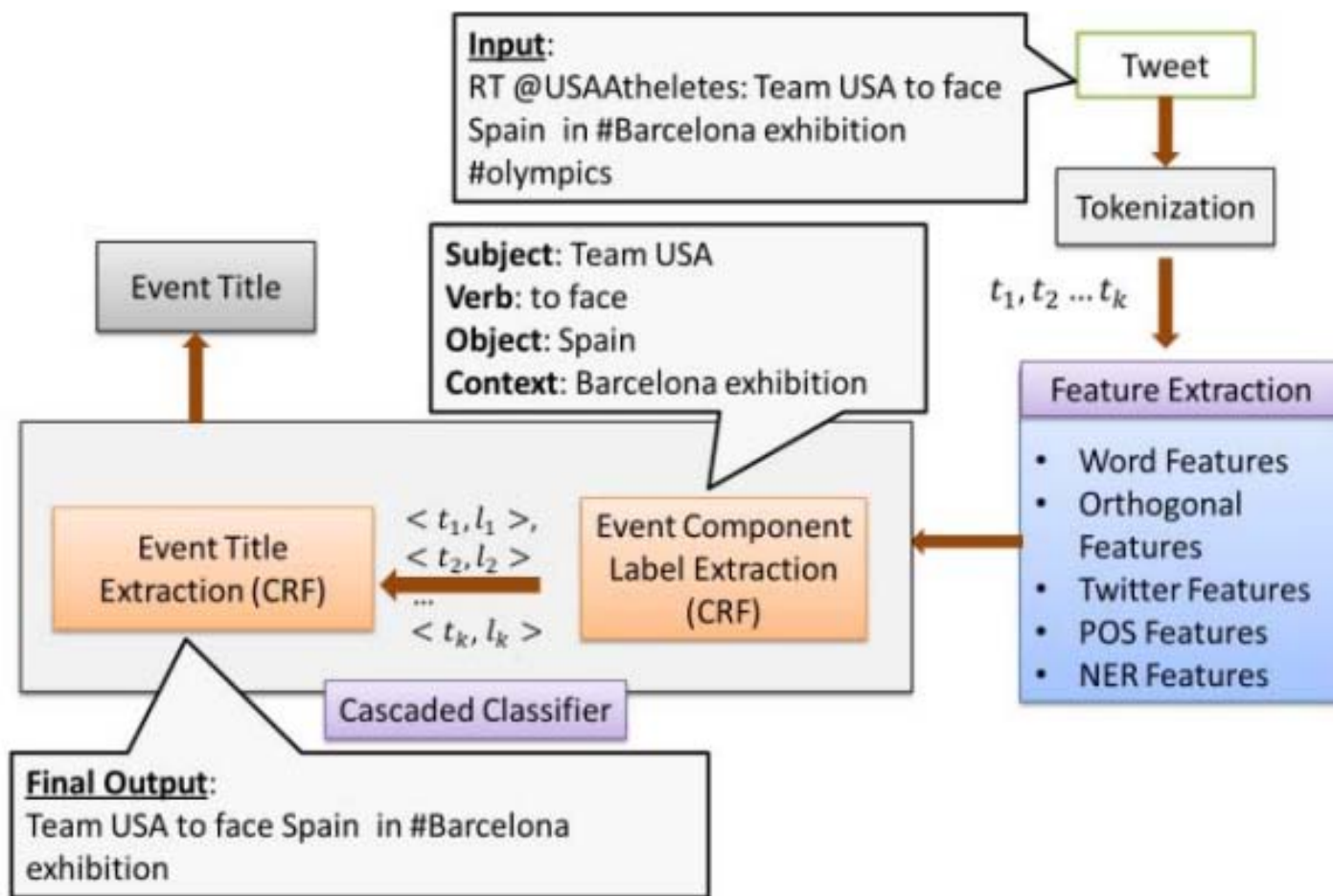
- Using CRF to identify event-referring phrases
 - Contextual features
 - POS tags
 - Adjacent words
 - Dictionary Features
 - Event words gathered from WordNet
 - Brown Clusters
 - Orthographic Features
 - Prefixes, suffixes

Entity	Event Phrase	Date
Steve Jobs	died	10/6/11
iPhone	announcement	10/4/11
GOP	debate	9/7/11
...

Detecting Structured Events using CRFs (1)

- Arpit Khurdiya, Lipika Dey, Diwakar Mahajan, and Ishan Verma. Extraction and Compilation of Events and Sub-events from Twitter. WI-IAT '12
- Two level of CRFs
 - First level identifies a sub-event comprising of actor, action, object, context, date, and location
 - Second level combines various sub-events to get the event title

Detecting Structured Events using CRFs (2)



Detecting Structured Events using CRFs (3)

- Features for the CRF predictor
 - Word Features – Each word in its lemmatized form
 - Orthogonal features – This set of features include capitalization, numeric features etc.
 - Twitter-specific features – hash-tags, user mentions, re-tweets etc.
 - Parts-Of-Speech Tags for a 5-word window
 - Named Entity Tags – These include tags like Names of People, Location, Date etc. assigned to words or sets of words

Detecting Breaking Events using Hashtags (1)

- Anqi Cui, Min Zhang, Yiqun Liu, Shaoping Ma, and Kuo Zhang. Discover breaking events with popular hashtags in twitter. CIKM '12.
- Issues with hashtags
 - A specific hashtag may refer to different objects, i.e., ambiguous.
 - #tea may refer to either the beverage or the Tea Party Movement
 - Different hashtags may describe the same event
 - #taiwanfloods and #morakot both refer to the typhoon Morakot that attacked Taiwan in 2009
 - Hashtags are sensitive to wide topics, so the topics they indicate may not be real-world events, i.e., they are not with a specific time period, location or people involved
- Twitter memes (conversational topics that attract users to share their own personal feelings) like , #iaintafraidtosay and #foramilliondollars are ephemeral but less valuable than the real events

Detecting Breaking Events using Hashtags (2)

- Study three attributes of hashtags
 - (1) Hashtag Instability
 - How unlikely is the hashtag in previous observations or future observations
 - $Inst(H) = \frac{1}{n} \sum_{\hat{P}(x) < p} Inst(x)$
 - $Inst(x) = -\log \hat{P}(x)$
 - p is a threshold
 - $\hat{P}(x) = \Pr(X > x \vee X < 2\mu - x)$ – tail probs. of Gaussian representing popularity of event
 - (2) Authorship Entropy
 - Measures how concentrated are the contributing authors for the hashtag
 - $Ent(h) = -\sum_{i=1}^k \frac{c_i}{n} \cdot \log\left(\frac{c_i}{n}\right)$

Detecting Breaking Events using Hashtags (3)

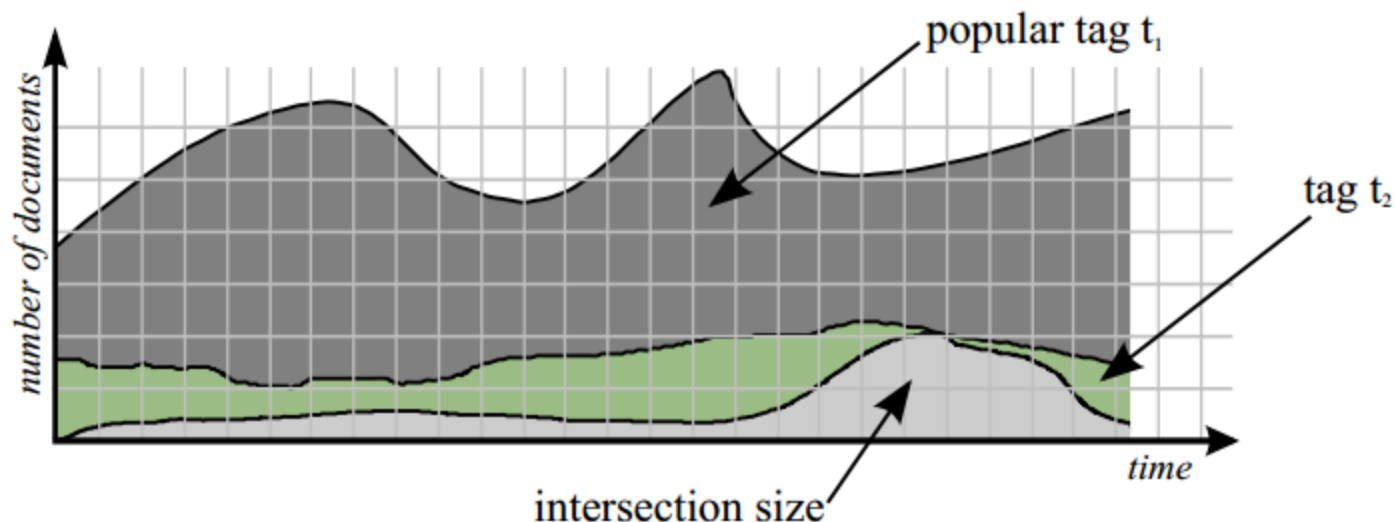
- (3) Twitter Meme Possibility (TMP)
 - How likely the hashtag denotes a Twitter meme
 - “Word length ratio”, i.e., the number of real English words N divided by the length of the hashtag L , and the probability of its appearing at the beginning of a tweet are both good signals
 - Such hashtags are usually written at the beginning of the tweet and contain concatenation of multiple English words
 - $p_{word} = 1 - \frac{N}{L}$ and $p_{pos} = \frac{|\text{tweets starting with } h|}{|\text{tweets containing } h|}$
 - $TMP(h) = p_{word} \cdot p_{pos}$

Categories of Hashtag Subspaces. L=Low, H=High. A=Advertisements, M=Miscellaneous, T=Twitter Memes, B=Breaking Events

Inst.	TMP	Ent.	Cat.	Inst.	TMP	Ent.	Cat.
L	L	L	A	H	L	L	A
L	L	H	M	H	L	H	B
L	H	L	A	H	H	L	A
L	H	H	T	H	H	H	T

Detecting Events using Tag Correlations (1)

- Foteini Alvanaki, Michel Sebastian, Krithi Ramamritham, and Gerhard Weikum. EnBlogue: emergent topic detection in web 2.0 streams. SIGMOD '11.
- Compared to Mathioudakis and Koudas's Twitter Monitor system, this work considers shifts in tag correlations

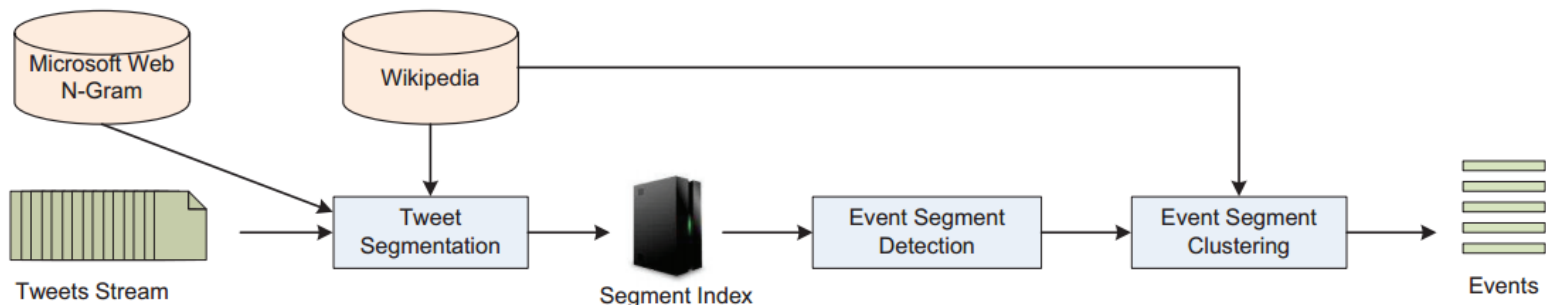


Detecting Events using Tag Correlations (2)

- Framework
 - Seed tag selection
 - Based on popularity
 - Correlation tracking
 - For each tag pair that contains at least one seed tag, track correlations
 - Shift detection
 - Sudden (but significant) increases in the correlation of tag pairs
 - If current correlation is significantly different from the prediction based on the previous correlation values

Detecting Events using Segments (1)

- Chenliang Li, Aixin Sun, and Anwitaman Datta. Twevent: segment-based event detection from tweets. CIKM '12.
- Each tweet is split into non-overlapping segments (i.e., phrases possibly refer to named entities or semantically meaningful information units).
- The bursty segments are identified within a fixed time window based on their frequency patterns, and each bursty segment is described by the set of tweets containing the segment published within that time window.
- The similarity between a pair of bursty segments is computed using their associated tweets.
- After clustering bursty segments into candidate events, Wikipedia is exploited to identify the realistic events and to derive the most newsworthy segments to describe the identified events.



Detecting Events using Segments (2)

- Three main components: tweet segmentation, event segment detection, and event segment clustering.
- Tweet segmentation problem
 - Given a tweet $d \in T$, the problem of tweet segmentation is to split d into m non-overlapping and consecutive segments, $d = \langle s_1 s_2 \dots s_m \rangle$, where a segment s_i is either a word (or unigram) or a phrase (or multi-gram).
 - Optimization problem: $\operatorname{argmax}_{s_1, s_2, \dots, s_m} C(d) = \sum_{i=1}^m C(s_i)$
 - Stickiness of segment s , $C(s) = L(s) \cdot e^{Q(s)} \cdot S(SCP(s))$
 - Where $L(s) = \begin{cases} \frac{|s|-1}{|s|}, & \text{for } |s| > 1 \\ 1, & \text{for } |s| = 1 \end{cases}$ and $\Pr(s)^2$
 - $SCP(s) = \log \frac{1}{\frac{1}{n-1} \sum_{i=1}^{n-1} \Pr(w_1 \dots w_i) \Pr(w_{i+1} \dots w_n)}$
 - $\Pr(s)$ is derived using Microsoft Web N-gram service
 - $Q(s)$ is the probability that s appears as the anchor text in the Wikipedia articles that contain s
 - $S(\cdot)$ is the sigmoid function

Detecting Events using Segments (3)

- Event Segment Detection
 - Given a collection of segments of the tweets published within a fixed time window, bursty segments in terms of frequency would be potentially related to some hot events talked and shared by Twitter users
 - Let N_t be number of tweets published in time window t .
 - p_s be the expected probability of tweets that contain segment s in a random time window
 - $f_{s,t}$ be number of tweets containing s published in t
 - Then we can model $P(f_{s,t}) \sim N(N_t p_s, N_t p_s (1 - p_s))$
 - Expected #tweets containing s in t is $E[s|t] = N_t p_s$
 - For tweets with $f_{s,t} \geq E[s|t] + 2\sigma[s|t]$, $P_b(s, t) = 1$
 - For tweets with $f_{s,t} \in (E[s|t], E[s|t] + 2\sigma[s|t])$, bursty probability is
$$P_b(s, t) = S\left(10 \times \frac{f_{s,t} - (E[s|t] + \sigma[s|t])}{\sigma[s|t]}\right)$$
 - $S(\cdot)$ is the sigmoid function
 - Let $u_{s,t}$ be #users who tweet segment s in time interval t
 - Select top K bursty segments based on $w_b(s, t) = P_b(s, t) \log(u_{s,t})$. They fix $K = \sqrt{N_t}$

Detecting Events using Segments (4)

- Event Segment Clustering
 - Similarity between two segments s_a and s_b is $sim(s_a, s_b) = \sum_{m=1}^M w_t(s_a, m) w_t(s_b, m) sim(T_t(s_a, m), T_t(s_b, m))$
 - $sim(T_1, T_2)$ measures similarity between 2 sets of tweets T_1 and T_2 . Concatenate all tweets in T_1 (and T_2) as a single document and then use cosine similarity with tf-idf scheme.
 - Two event segments appearing in each others' k-nearest neighbors are put into the same cluster
 - Resultant connected components in the graph of segments are candidate events
 - Segment newsworthiness $\mu(s) = \max_{l \in s} e^{Q(l)} - 1$
 - Where l is any sub-phrase of segment s
 - $Q(l)$ is the prob that l appears as anchor text in Wikipedia articles that contain l .
 - Event newsworthiness $\mu(e) = \frac{\sum_{s \in e_s} \mu(s)}{|e_s|} \cdot \frac{\sum_{g \in E_e} sim(g)}{|e_s|}$
 - e_s is the set of segments in event e
 - E_e are the edges in the connected component corresponding to the event e .
 - Events with top event newsworthiness scores are displayed.

Detecting Events using Segments (5): Examples

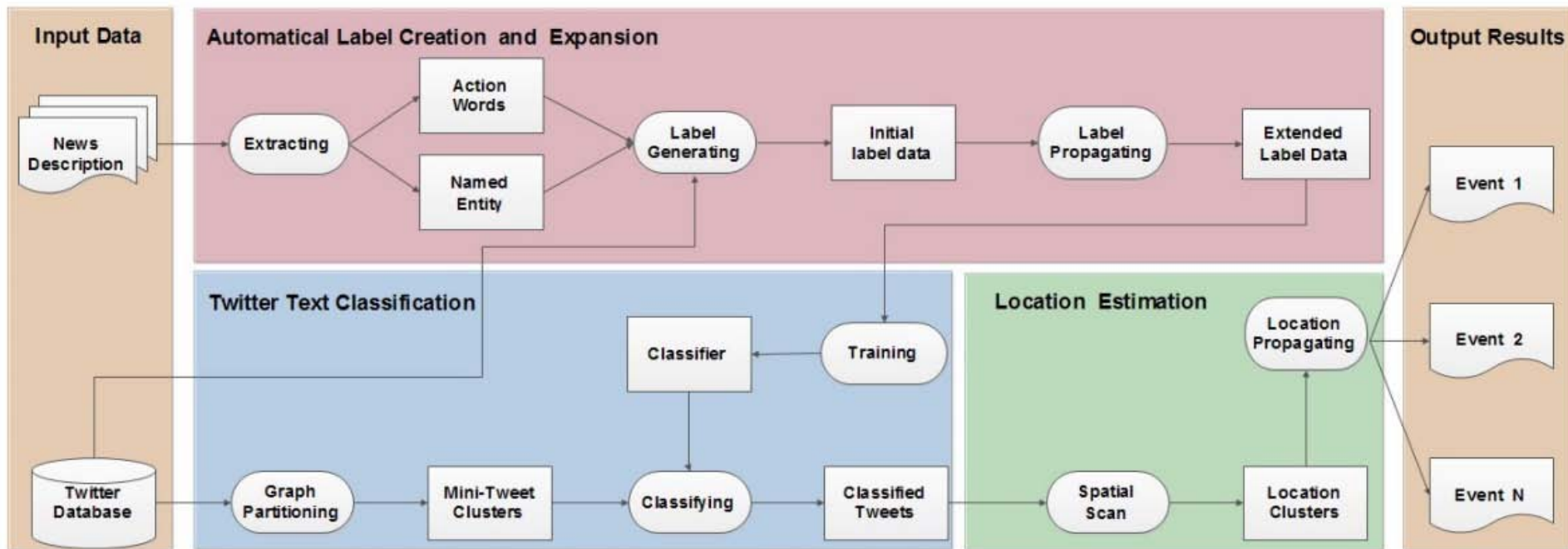
Day	e_{ID}	[Event Segments]: Event Description
7	e_1 .	[steve jobs, imovie, wwdc, iphone, wifi] : iPhone4 was released during WWDC 2010.
	e_2 .	[mtv movie awards, mtv, new moon, twilight, robe] : The movie <i>The Twilight Saga: New Moon</i> was the biggest winner in MTV Movie Awards 2010; it took 4 out of 10 "Best" Awards.
	e_3 .	[yesung, yesung oppa, kyuhyun, oppa, kyu] : Korean popular band Super Junior's showcase was held on June 6, 2010 at Singapore. Yesung Oppa and Kyuhyun Oppa are members of Super Junior.
8	e_4 .	[lady gaga, music video, gaga, mv, alejandro] : The music video <i>Alejandro</i> by Lady GaGa was premiered officially on June 8, 2010.
	e_5 .	[ss501, indonesia, ariel, sama, trend] : No clear corresponding real-life event.
	e_6 .	[singapore, iphone 4g, iphone 3gs, iphone, coming out] : Related to event e_1 . People started to talk about the release date of iPhone 4 in Singapore.
9	e_7 .	[lady gaga, youtube, youtube video, music video, gaga] : Related to event e_4 .
	e_8 .	[twitter, whale, stupid, capacity, over again] : A number of users complained they could not use twitter due to over-capacity. A logo with whale is usually used to denote over-capacity.
	e_9 .	[ipad, iphone, apple, new] : Related to event e_1 .
	e_{10} .	[watching glee, glee, season finale, season, channel] : The season finale of the American TV series <i>Glee</i> was broadcasted on June 8, 2010.
10	e_{11} .	[lady gaga, youtube, youtube video, music video, amber] : Related to event e_7 .
	e_{12} .	[justin bieber, try, pa, took, each] : Related to event e_{15} . The song <i>Never Say Never</i> by Justin Bieber serves as the theme song for the movie <i>The Karate Kid</i> , which was released on June 10, 2010 in Singapore.
	e_{13} .	[yesung, tweeted] : Super Junior's Yesung posted a photo about his pet turtles.
	e_{14} .	[twitter, whale, stupid, capacity, over] : Related to event e_8 .
	e_{15} .	[karate kid, watch movie, movie] : The movie <i>The Karate Kid</i> was released on June 10, 2010 in Singapore.

Tutorial Overview

- Event Detection for Twitter (80 min)
 - Event Detection using Tweet Content
 - **Event Detection using Other External Sources**
 - Applications of Event Detection
- Break (10 min)
- Event Description for Twitter (25 min)
 - Finding Best Phrase to Summarize an Event
 - Finding Event Types
 - Finding Event Timespans
 - Finding Event Credibility
 - Finding Event Locations
- User Profiling in Social Media (55 min)
 - Content-based Profiling
 - Network-based Profiling
 - Hybrid Approach
 - Co-profiling Attributes and Relationships
- Summary and Discussions (10 min)

Detecting Events by Label Propagation from News (1)

Ting Hua, Feng Chen, Liang Zhao, Chang-Tien Lu, and Naren Ramakrishnan. STED: semi-supervised targeted-interest event detection in twitter. KDD '13



Detecting Events by Label Propagation from News (2)

- From news articles, extract named entities and action words
- Tweets containing at least 1 named entity and 1 action word is labeled as positive
- Label propagation
 - Identify social ties terms from labeled tweets: Mentions(@), Hashtag(#)
 - Remove infrequent terms
 - Get more tweets from database which contain these terms
 - Label new tweets for a term as positive if $\frac{\text{newly discovered tweets}}{\text{already labeled tweets}} > \text{threshold}$ for term t
 - Iterate label propagation until no new tweets are found

Detecting Events by using Information from Knowledge Bases

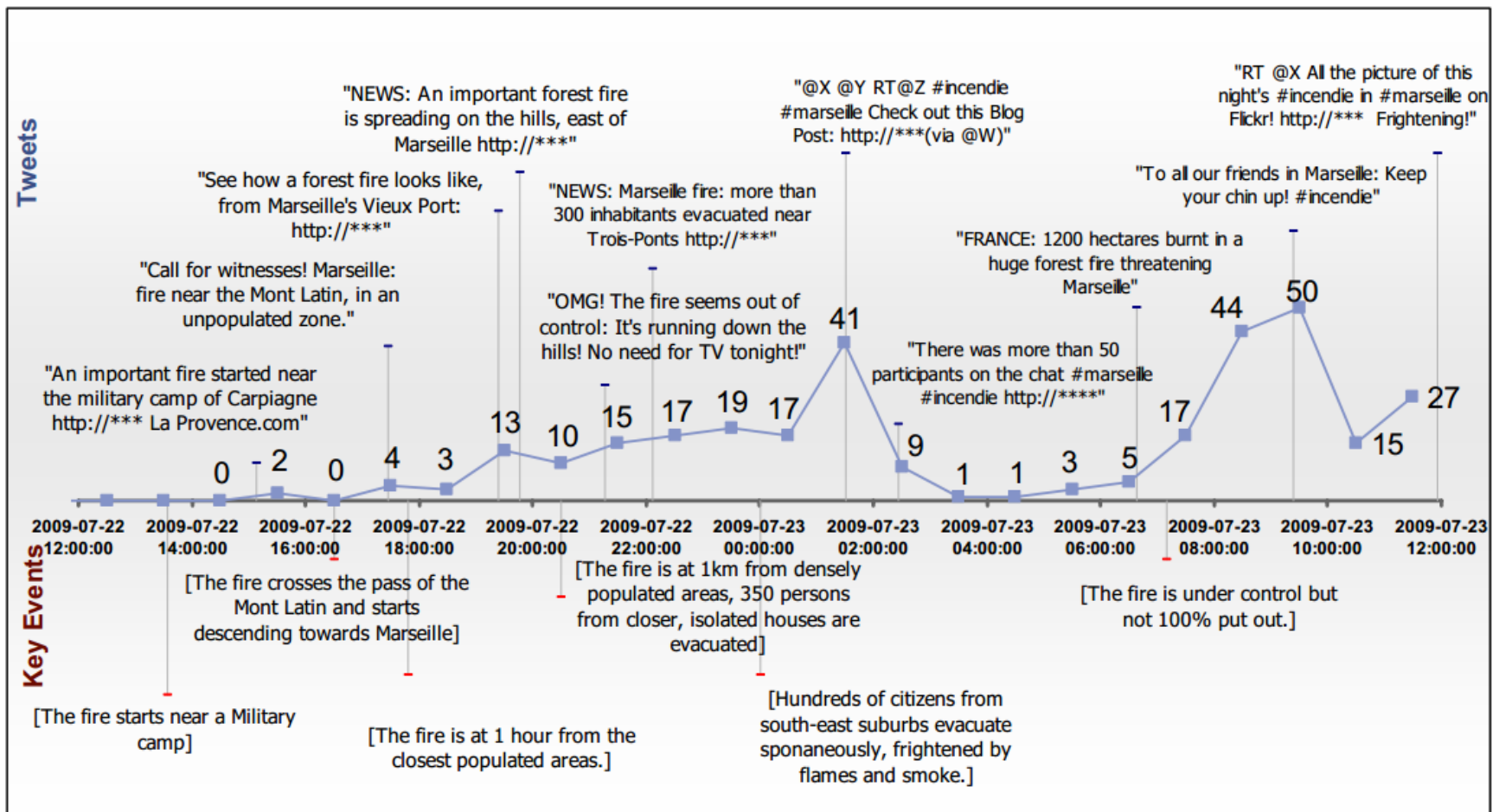
- Heather S. Packer, Sina Samangooei, Jonathon S. Hare, Nicholas Gibbins, and Paul H. Lewis. Event detection using Twitter and structured semantic query expansion. CrowdSens '12
- Given an event query, extract tweets containing the query
- From these tweets extract entities
- Find related entities from knowledge bases thereby extending the query
- Use these new entities to retrieve more tweets relevant to the event, thereby summarizing the event in a more comprehensive manner.

Tutorial Overview

- Event Detection for Twitter (80 min)
 - Event Detection using Tweet Content
 - Event Detection using Other External Sources
 - **Applications of Event Detection**
- Break (10 min)
- Event Description for Twitter (25 min)
 - Finding Best Phrase to Summarize an Event
 - Finding Event Types
 - Finding Event Timespans
 - Finding Event Credibility
 - Finding Event Locations
- User Profiling in Social Media (55 min)
 - Content-based Profiling
 - Network-based Profiling
 - Hybrid Approach
 - Co-profiling Attributes and Relationships
- Summary and Discussions (10 min)

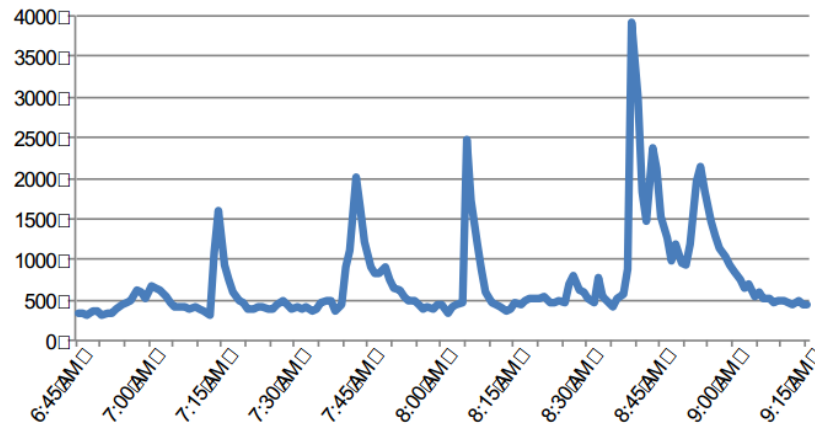
Detecting Forest-fires

Detecting Forest-fires: Bertrand De Longueville, Robin S. Smith, and Gianluca Luraschi.
"OMG, from here, I can see the flames!": a use case of mining location based social networks to acquire spatio-temporal data on forest fires. LBSN '09



Detecting Sporting Events (1)

- Jeffrey Nichols, Jalal Mahmud, and Clemens Drews. Summarizing sporting events using twitter. IUI '12
- Generate a journalistic summary of events from tweets
- Spikes are used to identify important moments to describe an event
- Sporting events consist of a sequence of moments, each of which may contain actions by players, the referee, the fans, etc.



Twitter volume graph for the 2010 World Cup game of US vs. Slovenia. The x-axis is time and the y-axis is volume as measured in tweets/minute.

Detecting Sporting Events (2)

Game	Actual Events	Detected		R	P
		Moments	Events		
US vs. Slovenia	13	9	8	0.62	0.89
Germany vs. Serbia	16	8	11	0.69	0.92
Australia vs. Serbia	11	9	10	0.91	0.91

Key Event Type	Recall
Goal	1.0
Red Card	1.0
Yellow Card	0.53
Penalty	1.0
Game Start	0.67
Game End	1.0
Half Time	0.33
Disallowed Goals	1.0

Detecting Sporting Events (3)

Game	Spike	Manual Summary	Our Summary
US vs. Slovenia	1	In the first 15 mins of the soccer game between USA and Slovenia, Slovenia is leading with a goal by Birsa. Birsa scored an easy goal from midfield to the right of the goal, as USA left that shot wide open. Terrible defense by USA team, too much space left open.	Good goal for Slovenia and the USA once again starts a game terrible. Birsa gives #SVN 1-0 lead with smart shot. Howard didn't even look like he saw that one coming.
Germany vs. Serbia	3	Klose argues with referee, gets second yellow cards and is out of the game. Germany down to 10 men. 1-0 Serbia.	Germany screwed by the refs and a red card for Klose; seconds later, a pretty goal by the Serbs. yellow seems to be a very popular colour in this game.
Australia vs. Serbia	9	Serbia Australia match ends with 2-1. With a result of 1:0 between Germany and Ghana this means that Ghana and Germany will advance to the knock out rounds and serbia and australia will be out.	Australia won 2-1 on serbia, Germany won 1-0 vs Ghana, Germany and Ghana goes on to the next round. Great win by #aus but not good enough to go through. Final score #Aus 2 #Srb 1.

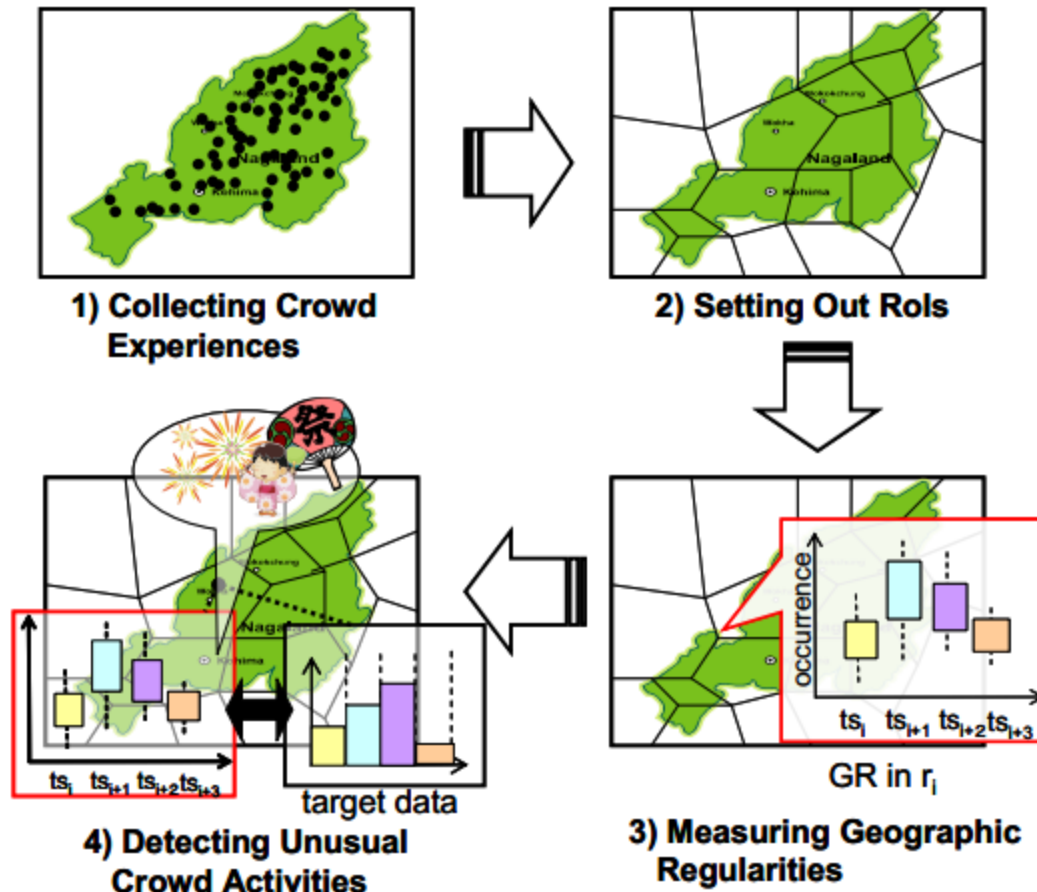
	Mean
Readability	6.01
Grammaticality	5.60
Content	5.19

7 point Likert scale

Detecting Local Festivals (1)

- Ryong Lee and Kazutoshi Sumiya. Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection. LBSN '10
- To detect such unusual geo-social events, they depend on geographical regularities deduced from the usual behavior patterns of crowds with geo-tagged microblogs.
- By comparing these regularities with the estimated ones, they decide whether there are any unusual events happening in the monitored geographical area.

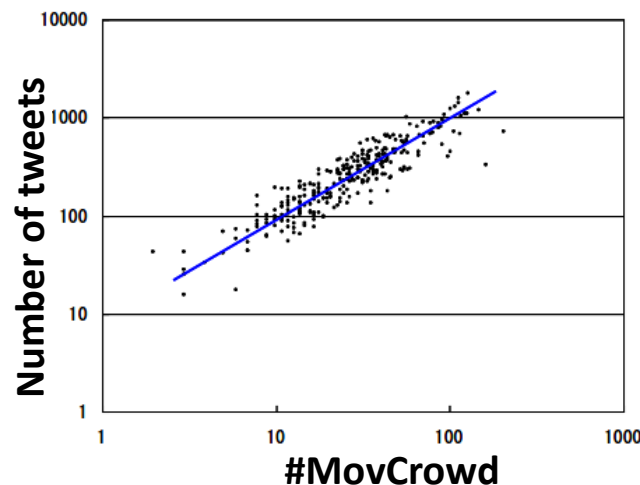
Detecting Local Festivals (2)



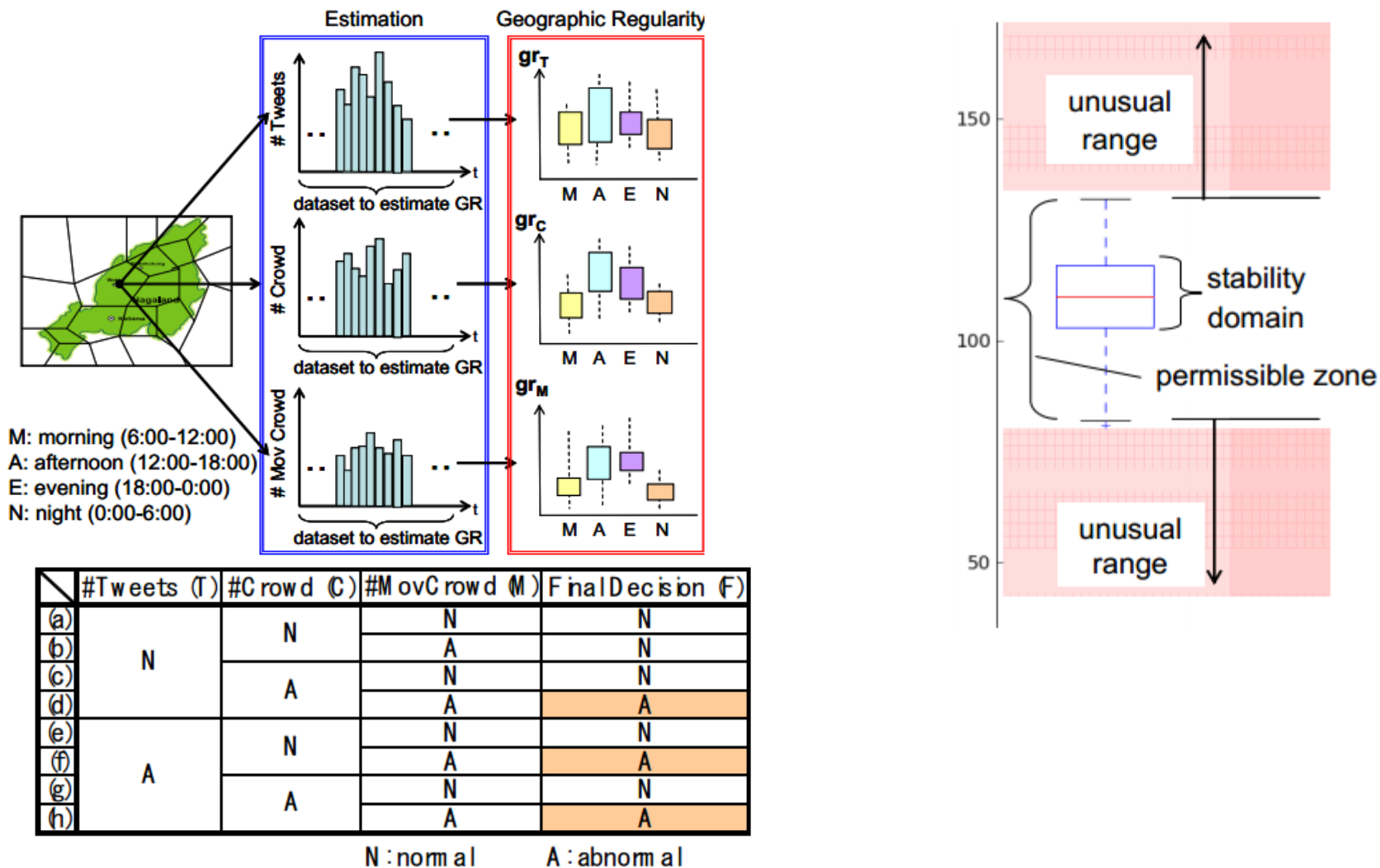
Rols = Regions of Interest
Rols (sub-regions) are computed from the region by running K-Means on geographically distributed points

Detecting Local Festivals (3)

- Measuring Geographical Regularity
 - #Tweets: the number of tweets that were written in an RoI within a specific period of time.
 - #Crowd: the number of Twitter users found in an RoI within a specific time period.
 - #MovCrowd: : 1) Inner: A crowd in an RoI moves only inside the region without going outside. 2) Incoming: There are some people coming from outside, and 3) Outgoing: Conversely, some people move outside the RoI. For simplification, they consider #MovCrowd as inner+incoming



Detecting Local Festivals (4)



Detecting Local Festivals (5)

Town festivals held in Japan for 7/17–7/19

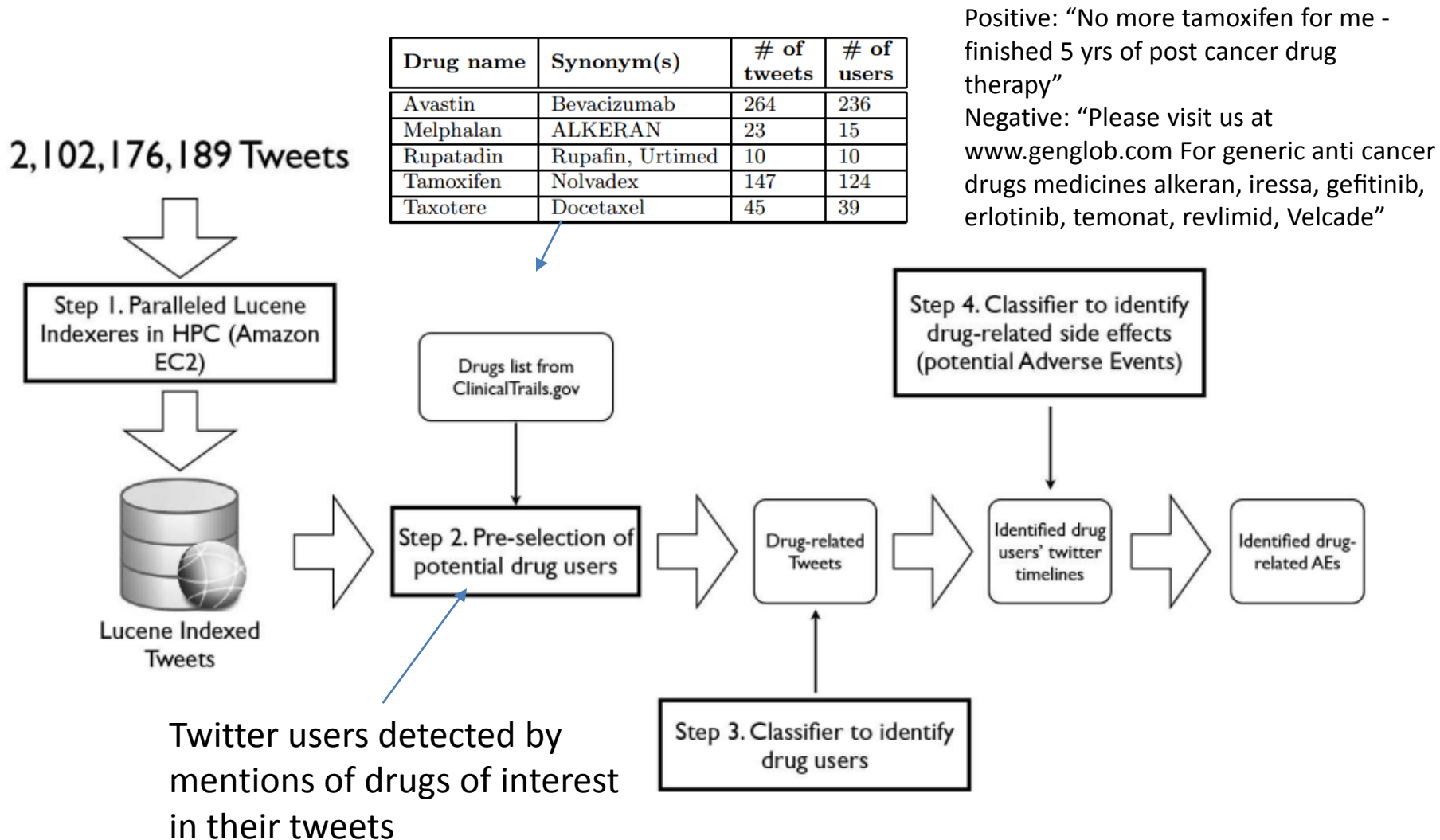
No.	Event name	Place	Event day(s)
1	Kyoto Gion Festival	Sakyo, Kyoto, Kyoto	7/17
2	Shishkui Gion Festival	Shishkui, Kaiyoumachi Tokushima	7/17, 7/18
3	Towada Kosui Festival	Towada, Aomori	7/17
4	Tamamura Firework	Tamura, Gunma	7/17
5	Ise Firework	Nakajima, Ise, Mie	7/17
6	Akiyoshi Firework	Syoho, Mie, Yanaguchi	7/17
7	Kannonji Festival	Kannonji, Kagawa	7/17, 7/18
8	Muroto Festival	Muroto, Kochi	7/18
9	Sanoyoi Carnival	Arao, Kumamoto	7/18
10	Umihohi Festival in Nagoya	Nagoya, Aichi	7/19
11	Housui Festival	Noboribetsu, Hokkaido	7/17
12	Shinmatsudo Festival	Matudo, Chiba	7/17, 7/18
13	Nanao Festival	Nanao, Ishikawa	7/17
14	Oota Festival	Oota, Gunma	7/17
15	Toukou Festival	Arita, Wakayama	7/18

The algorithm could detect 13 of the 15 festivals

Detecting Drug Related Adverse Events (1)

- Jiang Bian, Umit Topaloglu, and Fan Yu. Towards large-scale twitter mining for drug-related adverse events. SHB '12
- Given the high frequency of user updates, mining Twitter messages can lead us to real-time pharmacovigilance.
- To mine Twitter messages for AEs, the process can be separated into two parts: 1) identifying potential users of the drug; 2) finding possible side effects mentioned in the users' Twitter timeline that might be caused by the use of the drug concerned.

Detecting Drug Related Adverse Events (2)



Detecting Drug Related Adverse Events (3)

- Features for identifying drug users
 - Textual features that construct a specific meaning in the text:
 - Bag-of-words features that indicate an action or a state that the user has taken the drug
 - Number of hash-tags occurred in the document
 - Number of reply-tags occurred in the document
 - Number of words that indicate negation
 - Number of URLs
 - Number of pronouns
 - Number of occurrences of the drug name or its synonyms
 - Semantic features that express the existence of semantic properties (i.e., based on Unified Medical Language System (UMLS) Concept Unique Identifiers (CUIs) extracted from the Tweets)
 - Number of CUIs in each Semantic Type
 - Number of CUIs in each Semantic Group

Detecting Traffic Events

- Sílvio S. Ribeiro, Jr., Clodoveu A. Davis, Jr., Diogo Rennó R. Oliveira, Wagner Meira, Jr., Tatiana S. Gonçalves, and Gisele L. Pappa. Traffic Observatory: A System to Detect and Locate Traffic Events and Conditions using Twitter. LBSN '12
- Four phases
 - Preprocessing of the messages' content
 - Traffic event identification/detection
 - Using manually generated list of terms
 - Detection of locations using exact string matching
 - Enhancement of the location information using approximate string matching
 - To handle typos, shortened place names, nicknames, historical names

Most frequent traffic events and conditions found in the dataset

Event/Condition	Number of Tweets	Event/Condition	Number of Tweets
slow	2000	stopped	209
accident	582	free	198
stuck	499	jammed	100
regular	373	demonstration	86
intense	305	blocked	48
pay attention	277	complicated	31

Detecting Epidemics (1)

- Vasileios Lampos, Tijl De Bie, and Nello Cristianini. Flu detector: tracking epidemics on twitter. ECML PKDD'10
- Goal: Compute flu-score from Twitter corpus on a daily basis
- Solution: Learn a set of weighted keywords (or markers or features)
- Given a set of markers $\{m_i\}, i \in [1, n]$, their respective weights $\{w_i\}, i \in [1, n]$, and a set of tweets $\{t_j\}, j \in [1, k]$, Twitters' flu-score is $f_S(T, W, M) = \sum_j \sum_i w_i \times \frac{g(t_j, m_i)}{k}$ where $g(t_j, m_i) = 1$ if tweet t_j contains marker m_i , else it is 0

Detecting Epidemics (2)

- Learning weights and markers
 - Create a pool of candidate features by using encyclopedic and informal references related to influenza ($\theta=2675$ features denoted by $C = \{c_u\}, u \in [1, \theta]$)
 - Given a set T of k tweets, each feature c_u has a flu-score $f_{cu} = f_C(T, c_u) = \sum_j \frac{g(t_j, c_u)}{k}$
 - Across a time period of h days, time series $f_{cu}^{(h)}$ can be obtained resulting into a $h \times \theta$ matrix X
 - For same time period, golden flu values can be obtained from HPA ILI: y
 - Bolasso method (Lasso with L1 regularization for regression) can be used to obtain a sparse solution
 - $\min_w \|X^{(h)}w - y^{(h)}\|_2^2 \text{ s.t. } \|w\|_1 \leq t$

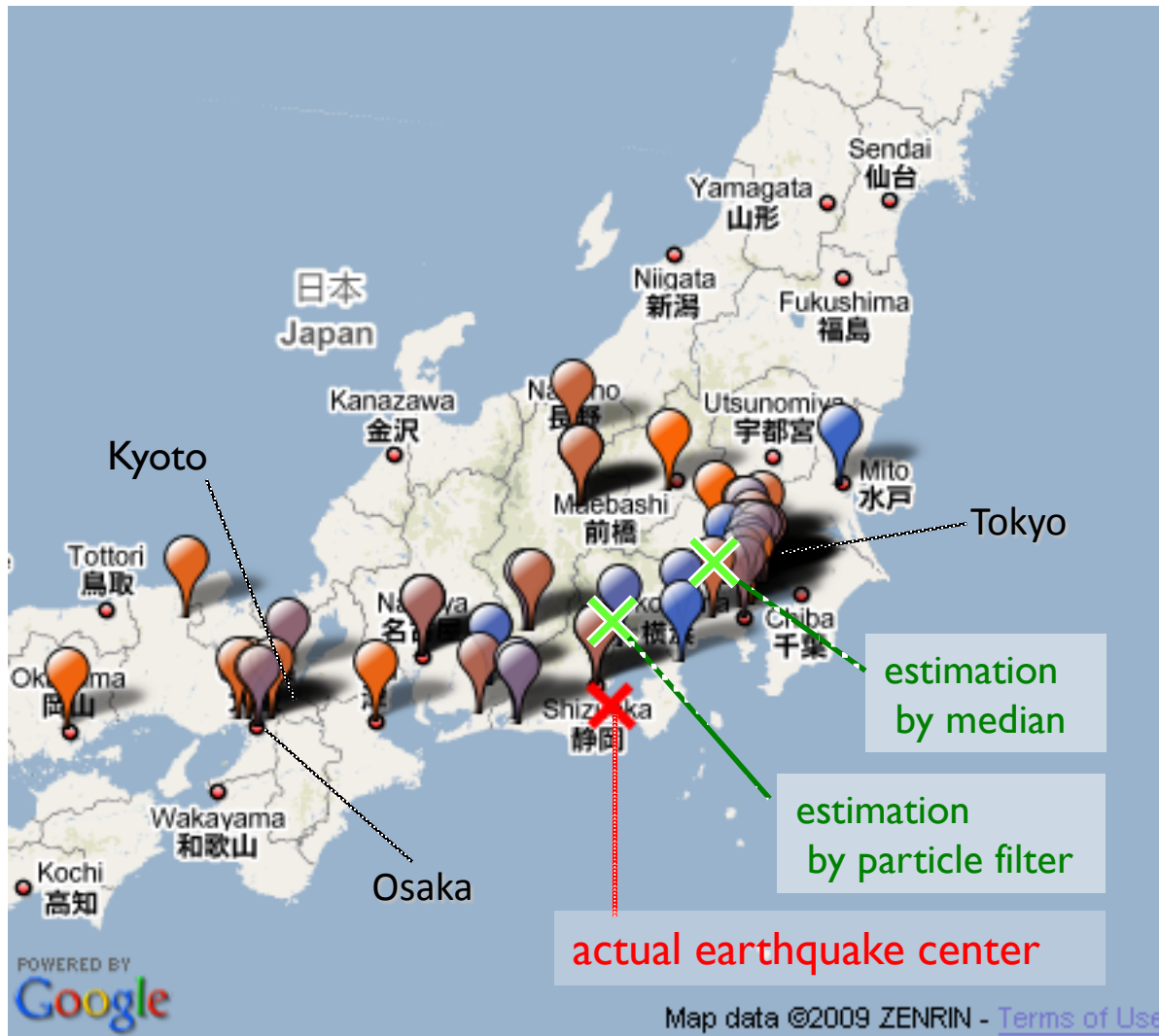
Detecting Earthquakes (1)

- Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. WWW '10
- Search tweets including keywords related to a target event
 - “shaking”, “earthquake”
- Classify tweets (SVM) into a positive class or a negative class
 - “Earthquake right now!!” ---positive
 - “Someone is shaking hands with my boss” --- negative
 - Features
 - Statistical features: # words in a tweet message and the position of the query within a tweet
 - Keyword features: the words in a tweet
 - Word context features: the words before and after the query word

Detecting Earthquakes (2)

- Probabilistic models for
 - Event detection from time-series data
 - Data fits very well to an exponential function
 - $f(t; \lambda) = \lambda e^{-\lambda t}$ ($t > 0, \lambda > 0$) ... $\lambda = 0.34$
 - Location estimation from a series of spatial information using
 - Kalman filters: useful for Gaussian-like spatial distributions
 - Particle filters: converge to the true posterior even in non-Gaussian, nonlinear dynamic systems.
 - Assume that the sensors are independent
 - Finding: In the case of an earthquakes and typhoons, very little information diffusion takes place on Twitter

Detecting Earthquakes (3): Location Estimation Example



- Particle filters performs better than other methods
- If the center of a target event is in an oceanic area, it's more difficult to locate it precisely from tweets
- It becomes more difficult to make good estimation in less populated areas
- A person has about 20~30 sec before its arrival at a point that is 100 km distant from an actual center

Detecting Emerging Controversial Events (1)

- Ana-Maria Popescu and Marco Pennacchiotti. Detecting controversial events from twitter. CIKM '10
- Controversial event: An event is controversial if it provokes a public discussion in which audience members express opposing opinions or disbelief
- Twitter snapshot=target entity+time period+set of tweets
- Goal: Rank Twitter snapshots by controversy score
- Regression model to detect event snapshots and to compute controversy scores
- Features
 - Twitter-based features: snapshots' linguistic properties, structural and social graph information, the intensity of the discussion about the entity, the distribution of sentiment words in the snapshot, and the level of controversy
 - News buzz features: if an entity is buzzy in news articles at the same time it is buzzy in a Twitter snapshot, then the snapshot is likely to refer to a real-world event.
 - Web and news controversy features: assess the past and present levels of controversy surrounding the target entity in the snapshot.

Detecting Emerging Controversial Events (2)

<i>Family</i>	<i>Type</i>	<i>Features</i>
<i>Twitter-based features</i>		
Linguistic	TW-LING-NOU	Percentage of tokens that are nouns (PoS come from the English dictionary in Sec. 2)
	TW-LING-VRB	Percentage of tokens that are verbs.
	TW-LING-BAD	Percentage of tokens that are <i>bad words</i> (we use the bad words lexicon described in Sec. 2.
	TW-LING-QST	Percentage of tweets containing at least one question (i.e. sentence with a question mark).
	TW-LING-LEV	Average Levenshtein distance between tweets.
	TW-LING-ENG	Percentage of tokens which match any word in the English dictionary described in Sec. 2.
	TW-LING-ENT-OC	Average number of mentions of the target entity across all tweets.
	TW-LING-ENT-VB	Percentage of verbs whose corresponding subject is the target entity.
Structural	TW-LING-ENT-TW	Percentage of tweets containing at least one verb whose subject is the target entity.
	TW-STRC-TOK	Number of tokens in the snapshot.
	TW-STRC-TWE	Number of tweets in the snapshot.
	TW-STRC-RET	Percentage of tweets that are retweets.
	TW-STRC-REP	Percentage of tweets that are replies.
	TW-STRC-USR	Average number of tweets per user.
	TW-STRC-TIM	Two features, representing mean and std.dev. of the distribution modeling tweets' timestamps.
Buzziness	TW-STRC-HST	Number of unique hashtags with respect to the total number of hashtags.
	TW-BUZZ	Estimates entity buzziness: $TW-BUZZ = \frac{ s }{(\sum_{i \in prev(s, N)} s_i)/N}$, where $ s $ is the number of tweets in the snapshot; $prev(s)$ are snapshots referring to the same entity of s , in N time periods previous to s (we set $N = 2$).
Sentiment	TW-SENT-POS	Fraction of positive tweets (i.e. $pol(t) > 0$)
	TW-SENT-NEG	Fraction of negative tweets (i.e. $pol(t) < 0$)
	TW-SENT-NEU	Fraction of neutral tweets (i.e. $pol(t) = 0$)

Detecting Emerging Controversial Events (3)

Controversy	TW-CONT-MIX	Estimate how many mixed positive and negative tweets are in the snapshot: $\text{TW-CONT-MIX} = \frac{\min(Pos , Neg)}{\max(Pos , Neg)} \cdot \frac{ Pos + Neg }{ Pos + Neg + Neu }$, where Pos , Neg and Neu are the sets of tweets with positive, negative and neutral polarity.
	TW-CONT-TSY	The contradiction score adopted by [7]: $\text{TW-CONT-TSY} = \frac{\theta \cdot \sigma^2}{\theta + (\mu)^2} \cdot W$, where μ and σ^2 are respectively the mean and the variance of the polarity scores $pol(t)$ of the tweets; parameters θ and W are set as in [7].
	TW-CONT-HST	Four features, representing the fraction over the total number of hashtags in the snapshot, of the following hashtags: ‘#controv’, ‘#scandal’, ‘#unheard’ and ‘#wft’.
	TW-CONT-TRM	Percentage of tweets with least one controversy word from our controversy lexicon in Sec. 2.
External features		
News buzz	EX-BUZZ-1	Number of articles aligned with the given snapshot.
	EX-BUZZ-2	Change in the amount of news coverage for the given entity with respect to the recent past: $\text{EX-BUZZ-2} = \frac{ articles - (\sum_{1 \leq i \leq N} articles_i) / N}{ articles }$, where $ articles_i $ is the number of articles about the target entity in a particular time period previous to s (we use $N = 7$).
Web-News controversy	EX-CONT-HIST	Controversy level of an entity in Web data: $\text{EX-CONT-HIST} = k / \text{controversyLexicon} $, where k is the number of terms in our controversy lexicon, whose co-occurrence pointwise mutual information with the target entity on the Web, is above 2; and $ \text{controversyLexicon} $ is the size of the controversy lexicon.
	EX-CONT-ASS-1	Sum of overall controversy scores (EX-CONT-HIST) for the entities co-occurring with the target entity in the aligned news article set.
	EX-CONT-ASS-2	Average of overall controversy scores (EX-CONT-HIST) for the entities co-occurring with the target entity in the aligned news article set.
	EX-CONT-TRM-1	Average number of controversy terms per news article (over all articles aligned with the snapshot)
	EX-CONT-TRM-2	Max number of controversy terms per news article (over all articles aligned with the snapshot).
	EX-CONT-TRM-3	Number of articles aligned with the snapshot that contain controversy terms .

Summary for Event Detection

- Detecting events from Twitter is difficult because of the unique characteristics of the microblogs
- We saw various ways of detecting interesting events from Twitter
 - Using Twitter content itself
 - Using external knowledge sources
- Finally, we discussed various applications of event detection on Twitter

Further Reading (1)

- Michael Mathioudakis and Nick Koudas. TwitterMonitor: trend detection over the twitter stream. SIGMOD '10
- Sayyadi, H., Hurst, M., and Maykov, A. (2009). Event Detection and Tracking in Social Streams. ICWSM.
- Saša Petrovic, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to Twitter. HLT '10.
- Arpit Khurdiya, Lipika Dey, Diwakar Mahajan, and Ishan Verma. Extraction and Compilation of Events and Sub-events from Twitter. WI-IAT '12
- Anqi Cui, Min Zhang, Yiqun Liu, Shaoping Ma, and Kuo Zhang. Discover breaking events with popular hashtags in twitter. CIKM '12.
- Foteini Alvanaki, Michel Sebastian, Krithi Ramamritham, and Gerhard Weikum. EnBlogue: emergent topic detection in web 2.0 streams. SIGMOD '11.
- Chenliang Li, Aixin Sun, and Anwitaman Datta. Twevent: segment-based event detection from tweets. CIKM '12.
- Ting Hua, Feng Chen, Liang Zhao, Chang-Tien Lu, and Naren Ramakrishnan. STED: semi-supervised targeted-interest event detection in twitter. KDD '13
- Heather S. Packer, Sina Samangooei, Jonathon S. Hare, Nicholas Gibbins, and Paul H. Lewis. Event detection using Twitter and structured semantic query expansion. CrowdSens '12
- Beaux Sharifi, Mark-Anthony Hutton, and Jugal Kalita. Summarizing microblogs automatically. HLT '10
- Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. Open domain event extraction from twitter. KDD '12

Further Reading (2)

- Donald Metzler, Congxing Cai, and Eduard Hovy. Structured event retrieval over microblog archives. HLT '12.
- Chung-Hong Lee, Hsin-Chang Yang, Tzan-Feng Chien, and Wei-Shiang Wen. A Novel Approach for Event Detection by Mining Spatio-temporal Information on Microblogs. ASONAM '11
- Detecting Forest-fires: Bertrand De Longueville, Robin S. Smith, and Gianluca Luraschi. "OMG, from here, I can see the flames!": a use case of mining location based social networks to acquire spatio-temporal data on forest fires. LBSN '09
- Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. WWW '11.
- Jeffrey Nichols, Jalal Mahmud, and Clemens Drews. Summarizing sporting events using twitter. IUI '12
- Ryong Lee and Kazutoshi Sumiya. Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection. LBSN '10
- Jiang Bian, Umit Topaloglu, and Fan Yu. Towards large-scale twitter mining for drug-related adverse events. SHB '12
- Sílvio S. Ribeiro, Jr., Clodoveu A. Davis, Jr., Diogo Rennó R. Oliveira, Wagner Meira, Jr., Tatiana S. Gonçalves, and Gisele L. Pappa. Traffic Observatory: A System to Detect and Locate Traffic Events and Conditions using Twitter. LBSN '12
- Vasileios Lamos, Tijl De Bie, and Nello Cristianini. Flu detector: tracking epidemics on twitter. ECML PKDD'10
- Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. WWW '10
- Ana-Maria Popescu and Marco Pennacchiotti. Detecting controversial events from twitter. CIKM '10

Tutorial Overview

- Event Detection for Twitter (80 min)
 - Event Detection using Tweet Content
 - Event Detection using Other External Sources
 - Applications of Event Detection
- **Break (10 min)**
- Event Description for Twitter (25 min)
 - Finding Best Phrase to Summarize an Event
 - Finding Event Types
 - Finding Event Timespans
 - Finding Event Credibility
 - Finding Event Locations
- User Profiling in Social Media (55 min)
 - Content-based Profiling
 - Network-based Profiling
 - Hybrid Approach
 - Co-profiling Attributes and Relationships
- Summary and Discussions (10 min)

Tutorial Overview

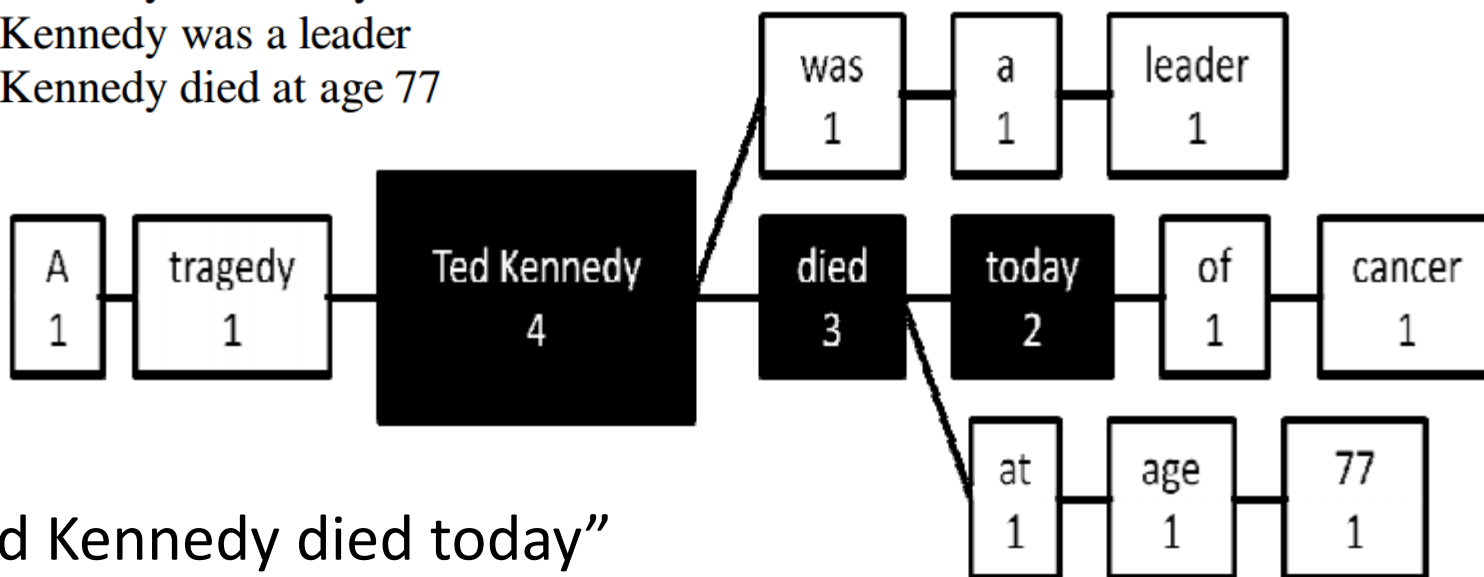
- Event Detection for Twitter (80 min)
 - Event Detection using Tweet Content
 - Event Detection using Other External Sources
 - Applications of Event Detection
- Break (10 min)
- **Event Description for Twitter (25 min)**
 - **Finding Best Phrase to Summarize an Event**
 - Finding Event Types
 - Finding Event Timespans
 - Finding Event Credibility
 - Finding Event Locations
- User Profiling in Social Media (55 min)
 - Content-based Profiling
 - Network-based Profiling
 - Hybrid Approach
 - Co-profiling Attributes and Relationships
- Summary and Discussions (10 min)

Finding the Best Phrase to Describe an Event

- Beaux Sharifi, Mark-Anthony Hutton, and Jugal Kalita.
Summarizing microblogs automatically. HLT '10

Event p = "Ted Kennedy"

1. A tragedy: Ted Kennedy died today of cancer
2. Ted Kennedy died today
3. Ted Kennedy was a leader
4. Ted Kennedy died at age 77



- "Ted Kennedy died today"

Finding the Best Phrase to Describe an Event

- How to describe these events corresponding to a phrase p ?
 - Phrase Reinforcement Algorithm
 - Get all tweets containing p
 - Remove spam and non-English tweets
 - Get the longest sentence from each post which contains p
 - Build a graph representing common sequences of words that occur both before and after p
 - Partial sentence = path with max total weight beginning from root and ending at a non-root node and containing nodes that occur $>T$ times
 - Build graph again by setting p as the partial path
 - Full sentence = path with max total weight beginning from root and ending at a non-root node and containing nodes that occur $>T$ times

Tutorial Overview

- Event Detection for Twitter (80 min)
 - Event Detection using Tweet Content
 - Event Detection using Other External Sources
 - Applications of Event Detection
- Break (10 min)
- Event Description for Twitter (25 min)
 - Finding Best Phrase to Summarize an Event
 - **Finding Event Types**
 - Finding Event Timespans
 - Finding Event Credibility
 - Finding Event Locations
- User Profiling in Social Media (55 min)
 - Content-based Profiling
 - Network-based Profiling
 - Hybrid Approach
 - Co-profiling Attributes and Relationships
- Summary and Discussions (10 min)

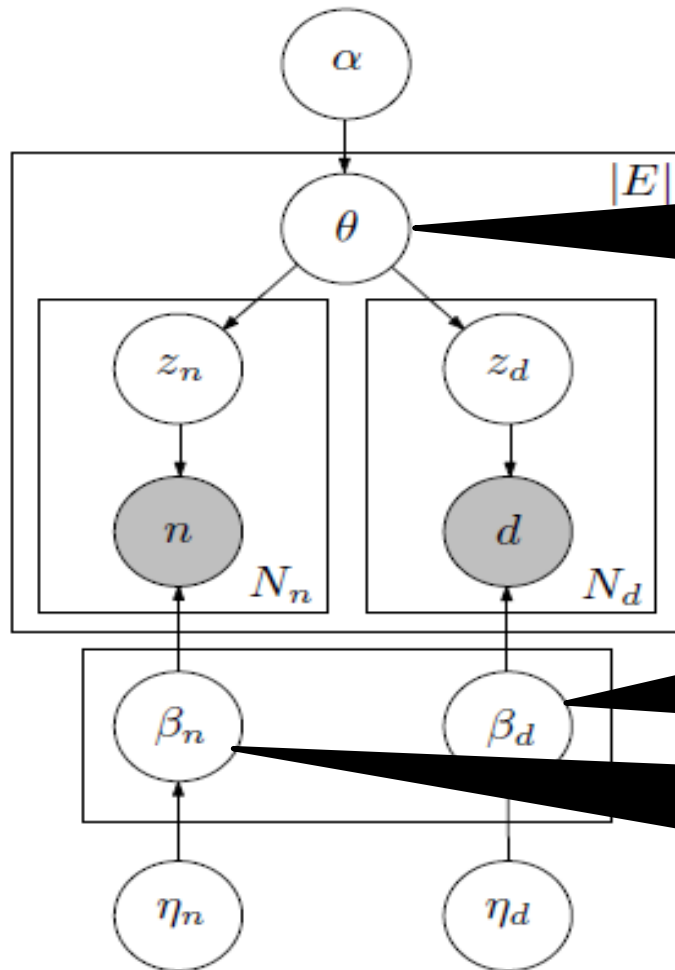
Finding Event Types (1)

- Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. Open domain event extraction from twitter. KDD '12.
- Would like to categorize events into types, for example:
 - Sports
 - Politics
 - Product releases
 - ...
- Benefits:
 - Allow more customized Twitter event calendars
 - Could be useful in upstream tasks

Finding Event Types (2)

- Challenges
 - Many Different Types
 - Not sure what is the right set of types
 - Set of types might change
 - Might start talking about different things
 - Might want to focus on different groups of users
- Solution: Unsupervised Event Type Induction
 - Latent Variable Models
 - Generative Probabilistic Models
 - Advantages:
 - Discovers types which **match the data**
 - No need to annotate individual events
 - Don't need to commit to a specific set of types
 - Modular, can integrate into various applications

Finding Event Types (3)



Each **Event Phrase** is modeled as a mixture of types

$$\begin{aligned} P(\text{SPORTS} | \text{cheered}) &= 0.6 \\ P(\text{POLITICS} | \text{cheered}) &= 0.4 \end{aligned}$$

Each **Event Type** is Associated with a Distribution over Entities and Dates

Finding Event Types (4)

Label	Top 5 Event Phrases	Top 5 Entities
Sports	tailgate - scrimmage - tailgating - homecoming - regular season	espn - ncaa - tigers - ea- gles - varsity
Concert	concert - presale - per- forms - concerts - tickets	taylor swift - toronto - britney spears - rihanna - rock
Perform	matinee - musical - priscilla - seeing - wicked	shrek - les mis - lee evans - wicked - broadway
TV	new season - season fi- nale - finished season - episodes - new episode	jersey shore - true blood - glee - dvr - hbo
Movie	watch love - dialogue theme - inception - hall pass - movie	netflix - black swan - in- sidious - tron - scott pil- grim
Sports	inning - innings - pitched - homered - homer	mlb - red sox - yankees - twins - dl
Politics	presidential debate - osama - presidential can- didate - republican debate - debate performance	obama - president obama - gop - cnn - america

Tutorial Overview

- Event Detection for Twitter (80 min)
 - Event Detection using Tweet Content
 - Event Detection using Other External Sources
 - Applications of Event Detection
- Break (10 min)
- Event Description for Twitter (25 min)
 - Finding Best Phrase to Summarize an Event
 - Finding Event Types
 - **Finding Event Timespans**
 - Finding Event Credibility
 - Finding Event Locations
- User Profiling in Social Media (55 min)
 - Content-based Profiling
 - Network-based Profiling
 - Hybrid Approach
 - Co-profiling Attributes and Relationships
- Summary and Discussions (10 min)

Finding Event TimeSpans (1)

Donald Metzler, Congxing Cai, and Eduard Hovy. Structured event retrieval over microblog archives. HLT '12.

July 16 2010 at 17 UTC, for 11 hours

Summary tweets:

- i. *Ok a 3.6 “rocks” nothing. But boarding a plane there now, Woodward ho! RT @todayshow: 3.6 magnitude #earthquake rocks Washington DC area.*
- ii. *RT @fredthompson: 3.6-magnitude earthquake hit DC. President Obama said it was due to 8 years of Bush failing to regulate plate tectonic ...*
- iii. *3.6-magnitude earthquake wakes Md. residents: Temblor centered in Gaithersburg felt by as many as 3 million people... <http://bit.ly/9iMLEk>*

Example structured event representation retrieved for the query “earthquake”.

- A list of timespans during which an instance of the event occurred and was actively discussed within the microblog stream.
- For each timespan, a small set of relevant messages are retrieved for the purpose of providing a high level summary of the event that occurred during the timespan

Finding Event TimeSpans (2)

- Framework
 - Timespan retrieval
 - Summarization
- Query expansion is needed because
 - Microblog messages that are highly related to the query might not contain any of the query keywords
 - Vocabulary mismatch: Keyword might be expressed in another form: possibly shortened or slang. E.g., earthquake may be written as quake or #eq

Finding Event TimeSpans (3)

- Temporal Query Expansion
 - Given query q , extract N timespans (hours) for which q was heavily discussed
 - Rank timespans based on proportion of messages posted during the timespan that contain q
 - Top few timespans are then considered to be pseudo-relevant.
 - For each word in all pseudo-relevant timespans, compute their burstiness score: $burstiness(w, TS_i) = \frac{P(w|TS_i)}{P(w)}$
 - $P(w|TS_i) = \frac{tf_{w,TS_i} + \frac{\mu tf_w}{N}}{|TS_i| + \mu}$ and $P(w) = \frac{tf_w + K}{N + K|V|}$
 - tf_{w,TS_i} is the number of occurrences of w in timespan TS_i
 - tf_w is the number of occurrences of w in the entire microblog archive
 - $|TS_i|$ is the number of terms in timespan TS_i
 - N is the total number of terms in the microblog archive, V is the vocabulary size, and μ and K are smoothing parameters
 - By smoothing $P(w)$, we dampen the effect of overweighting very rare terms.
 - Score of a word is geometric mean of burstiness scores across all pseudo-relevant timespans.
 - Geometric mean ensures that the highest weighted terms are those that have large weights in a large number of the timespans, thereby eliminating spurious terms
 - The k highest weighted terms are then used as expansion terms

Finding Event TimeSpans (4)

- Timespan Ranking
- We have expanded query q'
- Identify the 1000 highest scoring timespans (with respect to q')
- Merge contiguous timespans into a single, longer timespan, where the score of the merged timespan is the maximum score of its component timespans.
- The final ranked list consists of the merged timespans.
- Two scoring functions
 - Coverage Scoring Function
 - $s(q', TS) = \sum_{w \in q'} \beta_w \cdot tf_{w, TS}$
 - Where $tf_{w, TS}$ is the term frequency of w_i in timespan TS and β_w is the expansion weight of term w .
 - Burstiness Scoring Function
 - $s(q', TS) = \cos(\beta_{q'}, \beta_{TS})$ where β_{TS} is the burstiness score for all terms in the interval TS.

Finding Event TimeSpans (5)

- Timespan Summarization
 - Provide a quick overview of the event to the user
 - Retrieve a small set of microblog messages posted during the timespan that are the most relevant to the expanded representation of the original query
 - $s(q', M) = \sum_{w \in q'} \beta_w \cdot \log P(w|M)$
 - β_w is burstiness score of w
 - $P(w|M)$ is Dirichlet smoothed language modeling estimate for term w in message M

Finding Event TimeSpans (6): Examples

Category	Events
Business	layoffs, bankruptcy, acquisition, merger, hostile takeover
Celebrity	wedding, divorce
Crime	shooting, robbery, assassination, court decision, school shooting
Death	death, suicide, drowned
Energy	blackout, brownout
Entertainment	awards, championship game, world record
Health	recall, pandemic, disease, flu, poisoning
Natural Disaster	hurricane, tornado, earthquake, flood, tsunami, wildfire, fire
Politics	election, riots, protests
Terrorism	hostage, explosion, terrorism, bombing, terrorist attack, suicide bombing, hijacked
Transportation	plane crash, traffic jam, sinks, pileup, road rage, train crash, derailed, capsizes

July 16 2010 at 17 UTC, for 11 hours

Ok a 3.6 “rocks” nothing. But boarding a plane there now, Woodward ho! RT @todayshow: 3.6 magnitude #earthquake rocks Washington DC area.

September 28 2010 at 11 UTC, for 6 hours

RT @Quakeprediction: 2.6 earthquake (possible foreshock) hits E of Los Angeles; <http://earthquake.usgs.gov/earthquakes/recenteqscanv/Fau...>

September 04 2010 at 01 UTC, for 3 hours

7.0 quake strikes New Zealand - A 7.0-magnitude earthquake has struck near New Zealand’s second largest city. Reside... <http://ht.ly/18R2rw>

October 27 2010 at 01 UTC, for 5 hours

RT @SURFER_Magazine: Tsunami Strikes Mentawais: Wave Spawned By A 7.5-Magnitude Earthquake Off West Coast Of Indonesia <http://bit.ly/8Z9Lbv>

Tutorial Overview

- Event Detection for Twitter (80 min)
 - Event Detection using Tweet Content
 - Event Detection using Other External Sources
 - Applications of Event Detection
- Break (10 min)
- Event Description for Twitter (25 min)
 - Finding Best Phrase to Summarize an Event
 - Finding Event Types
 - Finding Event Timespans
 - **Finding Event Credibility**
 - Finding Event Locations
- User Profiling in Social Media (55 min)
 - Content-based Profiling
 - Network-based Profiling
 - Hybrid Approach
 - Co-profiling Attributes and Relationships
- Summary and Discussions (10 min)

Finding Credibility of Events (1)

- Classifier (User Features)
 - User has many **friends** and **followers**.
 - User has **linked** his Twitter profile **to his Facebook profile**.
 - User is a **verified** user.
 - User **registered** on Twitter long back.
 - User has made a **lot of posts**.
 - A **description, URL, profile image, location** is attached to user's profile.

Finding Credibility of Events (2)

- Classifier (Tweet Features)
 - It is **complete**. A more complete tweet gives a more complete picture of the truth.
 - A **professionally written** tweet with no slang words, question marks, exclamation marks, full uppercase words, or smileys is more credible.
 - Number of words with **first, second, third person** pronouns.
 - Presence of **supportive evidence** in the form of external URLs.
 - A tweet may be regarded as more credible if it is from the **most frequent location** related to the event.

Finding Credibility of Events (3)

- Classifier (Event Features)
 - Number of **tweets and retweets** related to the event.
 - Number of **distinct URLs, domains, hashtags, user mentions, users, locations** related to the event.
 - **Number of hours** for which the event has been popular.
 - **Percentage tweets** related to the event on **the day** when the event reached its peak popularity.

Tutorial Overview

- Event Detection for Twitter (80 min)
 - Event Detection using Tweet Content
 - Event Detection using Other External Sources
 - Applications of Event Detection
- Break (10 min)
- Event Description for Twitter (25 min)
 - Finding Best Phrase to Summarize an Event
 - Finding Event Types
 - Finding Event Timespans
 - Finding Event Credibility
 - **Finding Event Locations**
- User Profiling in Social Media (55 min)
 - Content-based Profiling
 - Network-based Profiling
 - Hybrid Approach
 - Co-profiling Attributes and Relationships
- Summary and Discussions (10 min)

Finding Locations of Events (1)

- GPS tags from Users' Tweets from GPS-enabled mobile devices
 - Current Tweets
 - Users' historical tweets
- Location field from User Profiles
 - City Names
 - GPS coordinates
- Other profile information that can be utilized to infer users' current location
 - UTC (Coordinated Universal Time) offset in the timezone field of tweets
 - URL domain names (e.g. .com for US, .jp for Japan, .de for Germany and .uk for UK) in profile "URL" field

Finding Locations of Events (2)

- However,
 - Less than one percent of tweets has GPS tags (Cheng et al. 2010)
 - Around 16% users provide locations in their profiles
- So, we need to profile users' locations (as well as their other attributes) from social media, such as users' tweets and their social network.

Summary for Event Description

- For best phrase detection, we discussed the phrase reinforcement algorithm.
- For event type detection, we discussed a latent variable model.
- For finding event timespans, we discussed a method that performs temporal query expansion.
- For event credibility computation, we discussed a classifier with interesting event, tweet and user features.
- For event location prediction, we will discuss ways to predict location of users.

Further Reading

- Beaux Sharifi, Mark-Anthony Hutton, and Jugal Kalita. Summarizing microblogs automatically. HLT '10
- Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. Open domain event extraction from twitter. KDD '12.
- Donald Metzler, Congxing Cai, and Eduard Hovy. Structured event retrieval over microblog archives. HLT '12.
- C. Castillo, M. Mendoza, and B. Poblete. Information Credibility on Twitter. In Proc. of the 20th Intl. Conf. on World Wide Web (WWW), pages 675–684, 2011.
- M. Gupta, P. Zhao, and J. Han. Evaluating Event Credibility on Twitter. In Proc. of the 2012 SIAM Intl. Conf. on Data Mining (SDM), pages 153–164, 2012.

Tutorial Overview

- Event Detection for Twitter (80 min)
 - Event Detection using Tweet Content
 - Event Detection using Other External Sources
 - Applications of Event Detection
- Break (10 min)
- Event Description for Twitter (25 min)
 - Finding Best Phrase to Summarize an Event
 - Finding Event Types
 - Finding Event Timespans
 - Finding Event Credibility
 - Finding Event Locations
- **User Profiling in Social Media (55 min)**
 - Content-based Profiling
 - Network-based Profiling
 - Hybrid Approach
 - Co-profiling Attributes and Relationships
- Summary and Discussions (10 min)

User Profiling is Difficult

- Content is noisy
 - Twitter users often rely on shorthand and non-standard vocabulary for informal communication.
 - Users may be interested in some big cities/events (such as New York, Japan earthquake).
- Social network is noisy
 - Users may connect to their friends, who live in different cities.
 - Users are more likely to follow only a few celebrities.
- A user may have more than one associated locations
 - E.g., a user studies at Illinois and works in California

Four Aspects of User Profiling with a focus on Location Prediction

- *Content based profiling* profiles users' attributes based on their tweets.
- *Network based profiling* profiles users' attributes based on the network (friends).
- *Hybrid based profiling* profiles users' based on both their tweets and the network.
- *Co-profiling* integrates user attribute profiling with other tasks to gain mutual enhancement.

We will mainly focus on profiling locations

Tutorial Overview

- Event Detection for Twitter (80 min)
 - Event Detection using Tweet Content
 - Event Detection using Other External Sources
 - Applications of Event Detection
- Break (10 min)
- Event Description for Twitter (25 min)
 - Finding Best Phrase to Summarize an Event
 - Finding Event Types
 - Finding Event Timespans
 - Finding Event Credibility
 - Finding Event Locations
- User Profiling in Social Media (55 min)
 - **Content-based Profiling**
 - Network-based Profiling
 - Hybrid Approach
 - Co-profiling Attributes and Relationships
- Summary and Discussions (10 min)

Content based Profiling

- Intuition
 - Users at a specific location (e.g., Houston) may tweet some local words in their tweets.
- Generally, we can take a text based classification approach
 - View attributes as labels.
 - Use words and other signals (lexicon, topics) as features.
 - Train classification models with users whose attributes are known.
- Many interesting extensions are proposed.

A Simple Probabilistic Model for Profiling Users' Countries and States

- B. Hecht, L. Hong, B. Suh, and E. H. Chi. Tweets from Justin Bieber's heart: the dynamics of the location field in user profiles. CHI 2011
- Select 10000 words that are discriminative in identifying users from a particular state/country
 - Count: Take top 10000 words based on #occurrences across all tweets from all users
 - Discriminative Score
 - 0 (if $\text{users}(t) < \text{minUsers}$)
 - $\max \frac{P(t|c = C)}{P(t)}$ (if $\text{users}(t) \geq \text{minUsers}$)
- Results
 - ~80% accuracy in identifying the right country
 - ~25% accuracy in identifying the right state

A Probabilistic Model for Profiling Users' Cities (Overview)

- Z. Cheng, J. Caverlee, and K. Lee, “You are where you tweet: a content-based approach to geo-locating twitter users,” in ACM CIKM 2010.
- Two Important Improvements
 - A feature selection component for automatically identifying words in tweets with a strong local geo-scope
 - A lattice-based neighborhood smoothing model for refining a user's location estimate
- Results
 - On average the location estimates converge quickly (needing just 100s of tweets), placing 51% of Twitter users within 100 miles of their actual location.

A Probabilistic Model for Profiling Users' Cities (Basic Model)

- Learn word-city distributions from already-geo-labeled users and their tweets
 - Houston has a large peak for “rockets”
 - It is the home of NASA and the NBA basketball team Rockets
 - $p(\text{city } i | \text{user } u) = \sum_{w \in T_u} p(i|w) \times p(w)$
 - Where $p(w)$ is computed using unigram model for the whole corpus
- 10% users have predicted location within 100 miles of actual location
 - AvgErrDist is 1,773 miles
- Problems
 - Most words are distributed consistently with the population across different cities
 - Most cities, especially with a small population, have a sparse set of words in their tweets

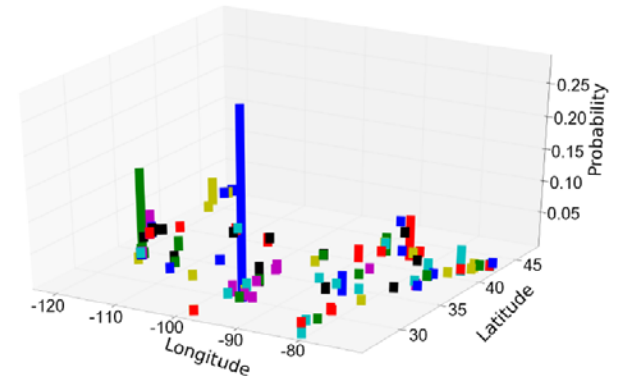


Figure 2: City estimates for the term “rockets”

Questions to ask

1. Is there a subset of words which have a more compact geographical scope compared to other words in the dataset? And can these "local" words be discovered from the content of tweets?
2. In what way can we overcome the location sparsity of words in tweets? Smoothing?

A Probabilistic Model for Profiling Users' Cities (Local Word Selection)

- Identifying local words in Tweets
 - Intuitively, a local word is one with a high local focus and a fast dispersion, that is it is very frequent at some central point (like say in Houston) and then drops off in use rapidly as we move away from the central point.
 - They follow a $Cd^{-\alpha}$ (Backstrom) model
 - d is distance from center
 - C is frequency (focus) at center
 - α is the dispersion parameter
 - Hierarchical Lattice Model
 - Let S be set of occurrences of word w
 - $f(C, \alpha) = \sum_{i \in S} \log C d_i^{-\alpha} + \sum_{i \notin S} \log(1 - C d_i^{-\alpha})$ is the likelihood value for a given center, C and α
 - Iteratively divide US using grid and identify best C and α and center location for each word
 - Learn classifier with C, α , center coordinates to discriminate between local versus non-local words

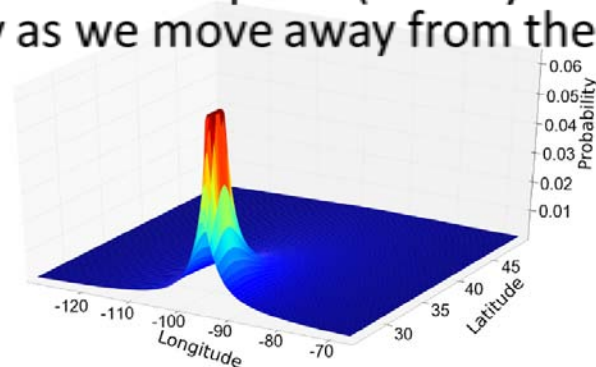


Figure 3: Optimized Model for the Word “rockets”

A Probabilistic Model for Profiling Users' Cities (Smoothing)

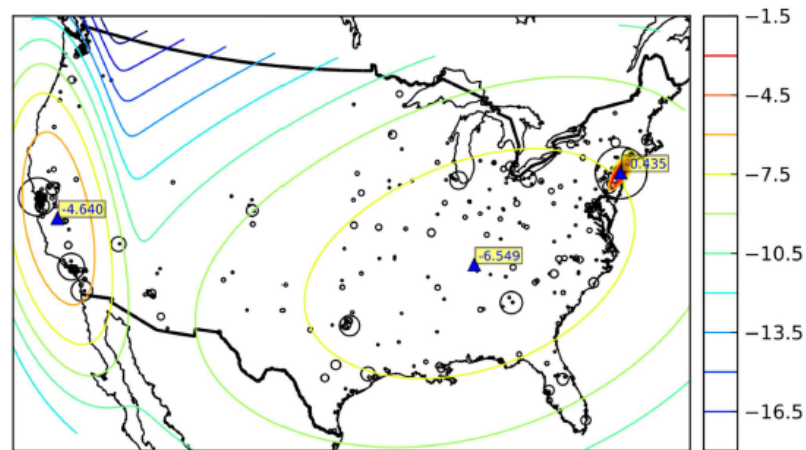
- Overcoming Tweet sparsity
 - Laplace smoothing: $p(i|w) = \frac{1+\text{count}(w,i)}{V+N(w)}$
 - $N(w)$ is total count of w across all cities
 - State-level smoothing: $p_s(s|w) = \frac{\sum_{i \in S_c} p(i|w)}{|S_c|}$ where S_c is the set of cities in state s
 - $p'(i|w) = \lambda p(i|w) + (1 - \lambda)p_s(s|w)$
 - Lattice-based neighborhood smoothing
 - $p(\text{lat}|w) = \sum_{i \in S_c} p(i|w)$ where S_c is set of cities in lat
 - $p'(\text{lat}|w) = \mu p(\text{lat}|w) + (1 - \mu) \sum_{\text{lat}_i \in \text{neighbors}} p(\text{lat}_i|w)$
 - $p'(i|w) = \lambda p(i|w) + (1 - \lambda)p'(\text{lat}|w)$
 - Model based smoothing
 - $p'(i|w) = C(w)d_i^{-\alpha(w)}$

Further Improvement: Selecting Location Indicative Words with Information Gain

- Han Bo, Paul Cook, Timothy Baldwin. Geolocation Prediction in Social Media Data by Finding Location Indicative Words. COLING 2012.
- Finding location indicative words (LIWs) via feature selection
 - High utility LIWs
 - High Term Frequency (TF): there should be a reasonable expectation of observing it for a given user
 - High Inverse City Frequency (ICF): the term should occur in tweets associated with a relatively small number of cities
 - Information gain ratio
 - $IG(w_i) = H(C) - H(C|w_i) \propto \frac{P(w_i) \sum_{j=1}^m P(c_j|w_i) \log P(c_j|w_i) + P(\overline{w_i}) \sum_{j=1}^m P(c_j|\overline{w_i}) \log P(c_j|\overline{w_i})}{H(C)}$
- Information gain ratio-based approach surpasses other methods at LIW selection, outperforming state-of-the-art geolocation prediction methods by 10.6% in accuracy and reducing the mean and median of prediction error distance by 45km and 209km, respectively, on a public dataset.

Further Improvement: GMMs to Handle Local Words with Multiple Peaks

- @Phillies Tweeting from Philly? Predicting Twitter User Locations with Spatial Word Usage. Hau-Wen Chang, Dongwon Lee, Mohammed Eltahery and Jeongkyu Leey. ASONAM 2012.
- Backstrom Model ($Cd^{-\alpha}$) allows for only 1 peak
 - How to handle multi-peak words?
 - E.g., giants for the NFL (football) NY Giants and the MLB (baseball) SF Giants
- Use Gaussian Mixture Model (GMM)
 - $P(c|w) = \sum_{i=1}^K \pi_i N(c|\mu_i, \Sigma_i)$



| giants

Tutorial Overview

- Event Detection for Twitter (80 min)
 - Event Detection using Tweet Content
 - Event Detection using Other External Sources
 - Applications of Event Detection
- Break (10 min)
- Event Description for Twitter (25 min)
 - Finding Best Phrase to Summarize an Event
 - Finding Event Types
 - Finding Event Timespans
 - Finding Event Credibility
 - Finding Event Locations
- User Profiling in Social Media (55 min)
 - Content-based Profiling
 - **Network-based Profiling**
 - Hybrid Approach
 - Co-profiling Attributes and Relationships
- Summary and Discussions (10 min)

Network Based Profiling

- Intuition
 - A user is likely to connect to friends in real life, who are likely to live close to the user.
- Generally, we can take a network (collective) classification approach.
 - Propagate friends' location labels to users in a iterative or non-iterative way.
- Many interesting ideas to extend the basic collective classification approach.

A Simple Propagation Approach for Profiling Users' Locations (Overview)

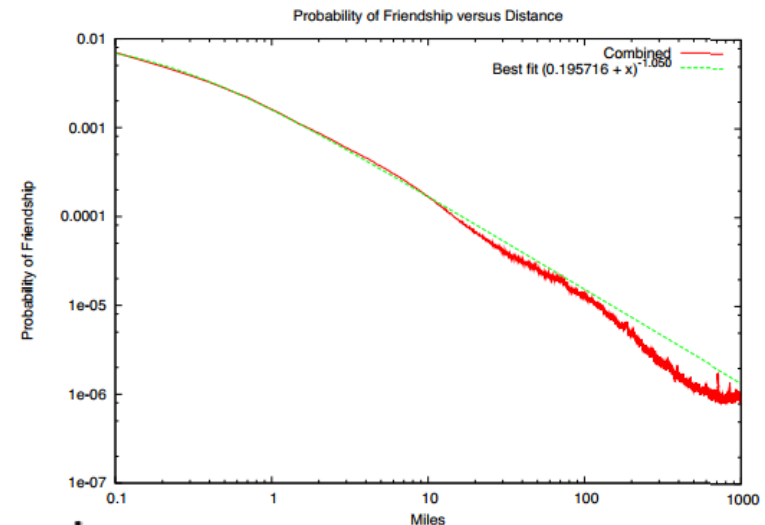
- Clodoveu A. Davis Jr., Gisele L. Pappa, Diogo Rennó Rocha de Oliveira, Filipe de L. Arcanjo. Inferring the Location of Twitter Messages Based on User Relationships. TGIS 2011.
- Propagate locations via relationships
 - Count the most popular locations among the friends of a user, using a simple majority voting scheme
 - The most popular location among friends is set as the location of a user
 - Some rules used in the approach
 - Minimum and Maximum number of friends a user should have in order to have his or her location correctly inferred
 - Minimum number of votes a location needs to be considered as the correct one

A Probabilistic Propagation Approach for Profiling Users' Locations (Overview)

- Lars Backstrom et.al. Find me if you can: improving geographical prediction with social and spatial proximity. In *Proceedings of WWW '10*. 61-70.
- Propagate locations among friends probabilistically
 - Define the probability of being friends based on users' locations
 - Consider distances of locations
 - Is Robust to noisy data
- Results
 - Dataset 2.9 million Facebook users whose locations are known.
 - Achieve 67.5% accuracy within 50 miles and improve IP based baseline by 10% for users who have more ten labeled friends

A Probabilistic Propagation Approach for Profiling Users' Locations (Model)

- Investigate the probability of friendship as a function of distance based on a large data set.
- Two observations
 - The probabilities go down as distance increases
 - The probabilities fit an exponential distribution
- Thus, the probability that two users are friends given their locations can be defined as
 - $p(l_u - l_v) = 0.0019(|l_u - l_v| + 0.196)^{-1.05}$



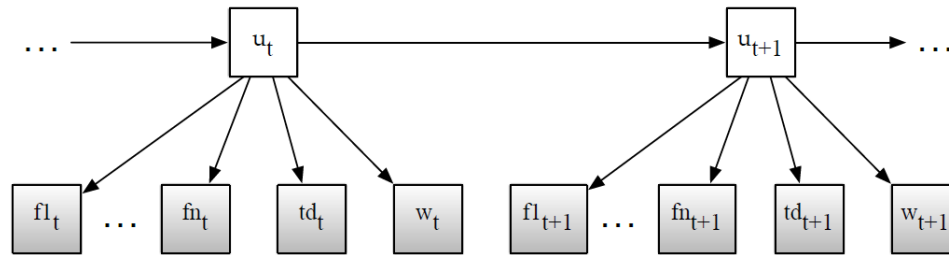
A Probabilistic Propagation Approach for Profiling Users' Locations (Algorithm)

- Define the likelihood function as
$$\prod_{(u,v) \in E} p(|l_u - l_v|) \prod_{(u,v) \notin E} 1 - p(|l_u - l_v|)$$
- Given friends and their location, we can find the location of the user via MLE.
- Some optimization techniques are used to compute the function efficiently.
 - E.g., only consider friends' locations as candidates

A Dynamic Bayesian Network For Profiling Users' Locations (Overview)

- A. Sadilek, H. Kautz, and J. P. Bigham. Finding your friends and following them to where you are. WSDM '12.
- Best paper at WSDM 2012.
- It studies two separated problems:
 - Location profiling
 - GPS and Time Information
 - Fine grained profiling
 - Friendship prediction
- Results
 - Achieved 57% accuracy with friends' locations information.

A Dynamic Bayesian Network For Profiling Users' Locations (Model)



- Dynamic Bayesian Network
- The hidden node represents the location of the target user (u).
- The node td represents the time of day and w determines if a given day is a work day or a free day (weekend or a national holiday).
- Each of the remaining observed nodes ($f1$ through fn) represents the location of one of the target user's friends.
- Supervised
 - Location of ' u ' over the training period is given
 - $\theta^* = \operatorname{argmax}_{\theta} \log P(x_{1:t}, y_{1:t} | \theta)$
- Unsupervised
 - Only u 's friends location during training period is given
 - $\theta^* = \operatorname{argmax}_{\theta} \log \sum_{y_{1:t}} P(x_{1:t}, y_{1:t} | \theta)$

A Dynamic Bayesian Network For Profiling Users' Locations (Algorithm)

- Viterbi Algorithm to find the hidden variables
- $y_{1:t}^* = \operatorname{argmax}_{y_{1:t}} \log(\Pr(y_{1:t}|x_{1:t}))$

Tutorial Overview

- Event Detection for Twitter (80 min)
 - Event Detection using Tweet Content
 - Event Detection using Other External Sources
 - Applications of Event Detection
- Break (10 min)
- Event Description for Twitter (25 min)
 - Finding Best Phrase to Summarize an Event
 - Finding Event Types
 - Finding Event Timespans
 - Finding Event Credibility
 - Finding Event Locations
- User Profiling in Social Media (55 min)
 - Content-based Profiling
 - Network-based Profiling
 - **Hybrid Approach**
 - Co-profiling Attributes and Relationships
- Summary and Discussions (10 min)

Hybrid Approach: Profiling Users' Locations based on both Content and Network

- Integrate both content and network
- Capture additional insights
- We will focus on two works

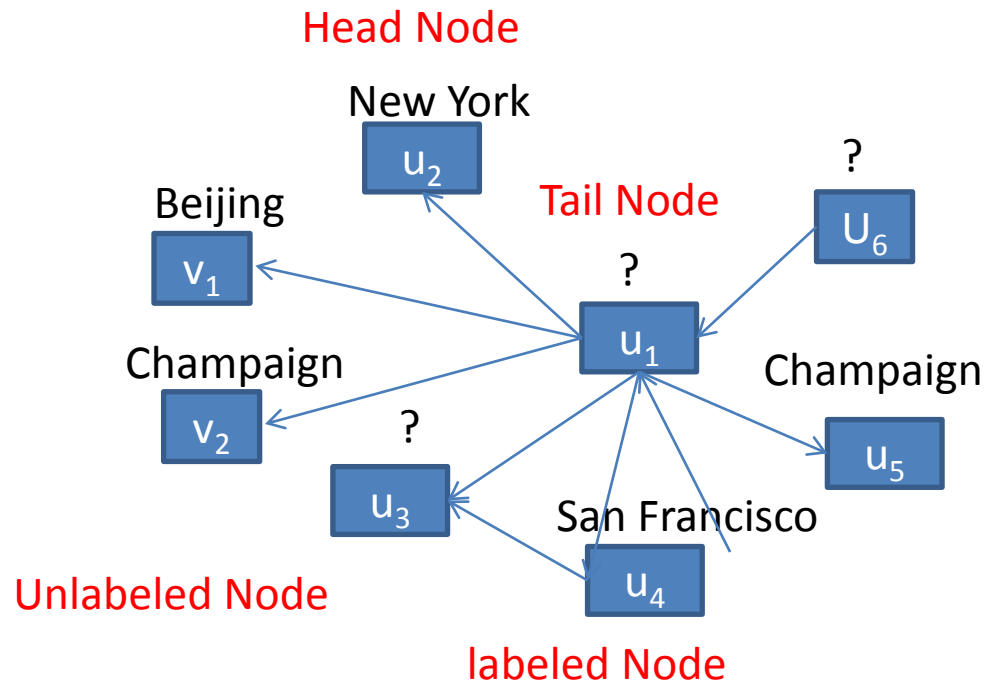
Unified and Discriminative Influence Model for Inferring Home Locations (Overview)

- Rui Li, Shengjie Wang, Hongbo Deng, Rui Wang, Kevin Chen-Chuan Chang: Towards Social User Profiling: Unified and Discriminative Influence Model for Inferring Home Locations. *KDD* 2012.
- Ideas
 - Integrate content and network of users and locations as a directed graph
 - Propose influence model to capture nodes' influences
 - E.g., location of a local friend is more useful than the location from a celebrity
- Results
 - 160K Twitter Users
 - Improve the previous algorithm by 7% with only network data
 - Improve the previous algorithm by about 12% overall

Unified and Discriminative Influence Model for Inferring Home Locations (Representation)

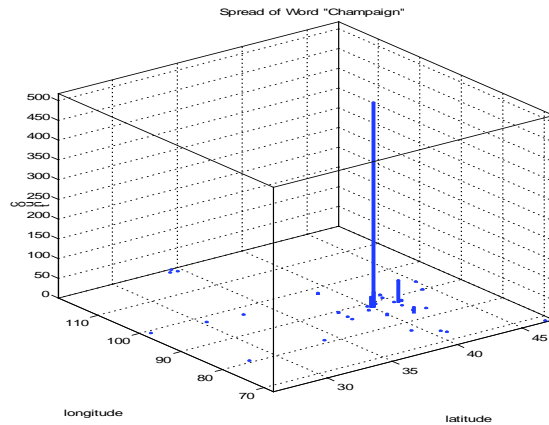
We unify two types of resources (local words and friends) as a directed heterogeneous graph

- * We model it as a directed graph.
- * We aim to infer the locations of unlabeled nodes with locations of labeled nodes.

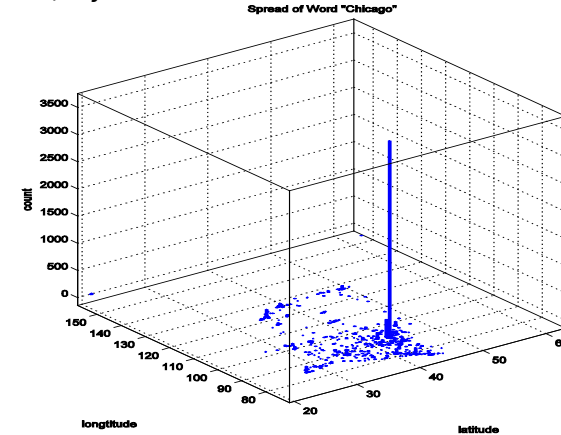


Unified and Discriminative Influence Model for Inferring Home Locations (Model)

How likely a tail node n_j at $L(n_j)$ builds an edge $e < n_i, n_j >$ a head node n_i at $L(n_i)$



Observation 1 *The probability decreases as their distance increases*



Observation 2 *At the same distance, different head (Chicago, Champaign) nodes have different probabilities to attract tail nodes.*

An influence model for each node at $L(n_i)$ with difference influence scopes to capture the probabilities

$$P(e < n_j, n_i > | \theta_{n_i}, L(n_j)) = \frac{1}{2\pi\sigma_{n_i}^2} e^{-\frac{(x_{u_i} - x_{u_j})^2 + (y_{u_i} - y_{u_j})^2}{2\pi\sigma_{n_i}^2}}$$

gmanish@microsoft.com, ruililab@yahoo-inc.com, kcchang@illinois.edu

Unified and Discriminative Influence Model for Inferring Home Locations (Local Algorithm)

- Simple but efficient
- Closed-form solution

Average Distance of a User's Followers

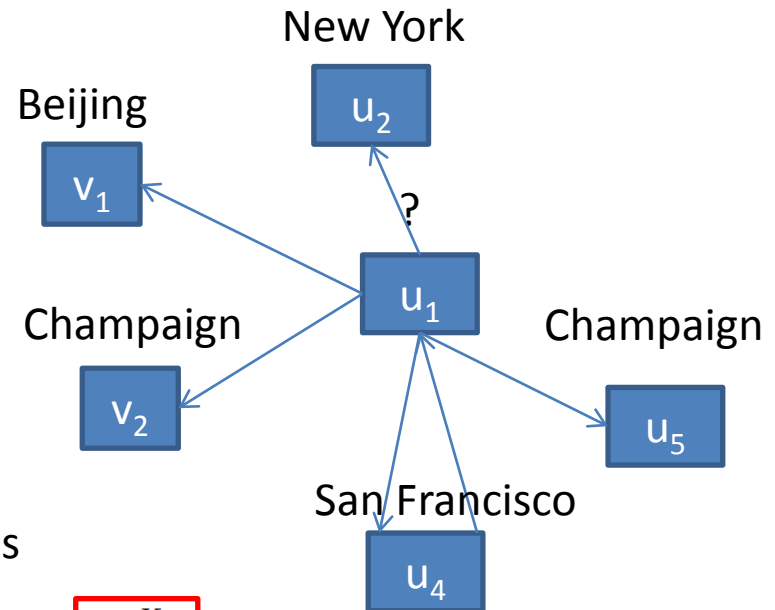
Influence Scope

$$\sigma_{u_i}^2 = \sum_{u_j \in \mathcal{I}_f^*(u_i)} \frac{(X_{u_j} - X_{u_i})^2 + (Y_{u_j} - Y_{u_i})^2}{2|\mathcal{I}_f^*(u_i)|}$$

User Location

Weighted Average of Different Resources

$$X_{u_i} = \frac{\sum_{u_j \in \mathcal{I}_f^*(u_i)} \frac{X_{u_j}}{\sigma_{u_j}^2} + \sum_{u_j \in \mathcal{O}_f^*(u_i)} \frac{X_{u_j}}{\sigma_{u_j}^2} + \sum_{v_j \in \mathcal{O}_t(u_i)} \frac{w_{ij} X_{v_j}}{\sigma_{v_j}^2}}{\sum_{u_j \in \mathcal{I}_f^*(u_i)} \frac{1}{\sigma_{u_j}^2} + \sum_{u_j \in \mathcal{O}_f^*(u_i)} \frac{1}{\sigma_{u_j}^2} + \sum_{v_j \in \mathcal{O}_t(u_i)} \frac{w_{ij}}{\sigma_{v_j}^2}}$$

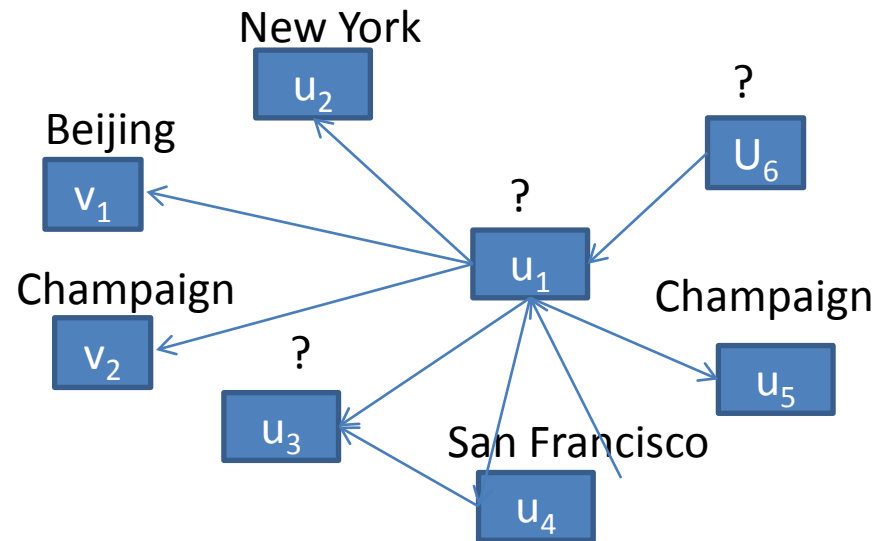


Unified and Discriminative Influence Model for Inferring Home Locations (Global Algorithm)

The local algorithm only uses limited information.

The global algorithm aims to use all information.

- Complex but accurate
- Iterative algorithm



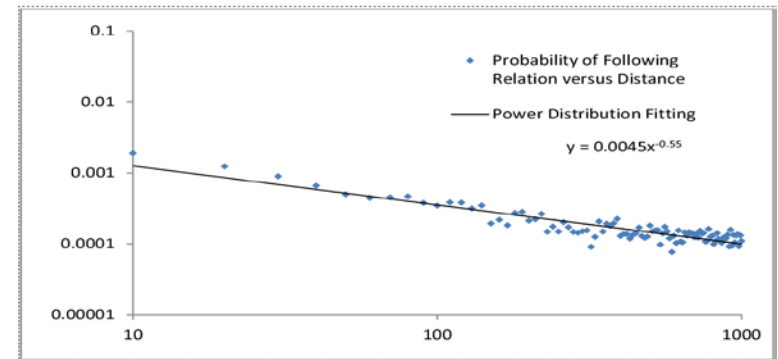
Multiple Location Profiling based on Content and Social Network (Overview)

- Rui Li, Shengjie Wang, Kevin Chen-Chuan Chang. Multiple Location Profiling for Users and Relationships from Social Network and Content. *VLDB* 2012.
- Ideas
 - Use two probabilistic models to connect locations with content and friendship
 - Introduce mixture model to capture a user may related to multiple location
 - E.g., a user may live in California and studies in Illinois
- Results
 - Improve the baseline by 10%
 - Discover users' multiple locations completely

Multiple Location Profiling based on Content and Social Network (Following and Tweeting Model)

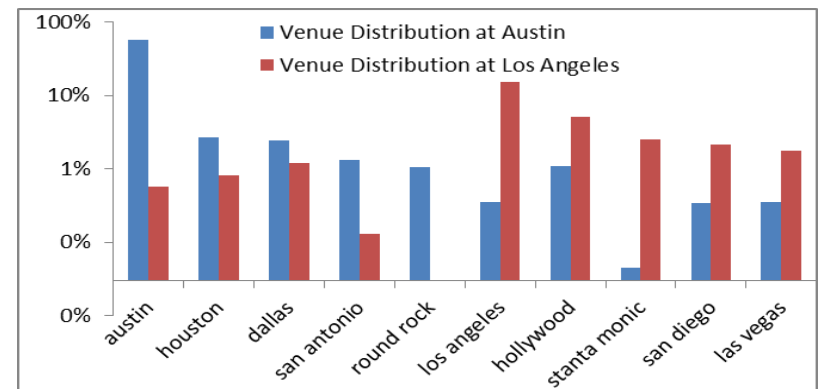
- The location-based following model

$$P(f\langle i, j \rangle | \alpha, \beta, x_i, y_j) = \beta d(x_i, y_j)^\alpha$$



- The location-based tweeting model

$$P(t\langle i, j \rangle | \psi_{1:L}, z_i) = P(v_j | \psi_{z_i}).$$



Multiple Location Profiling based on Content and Social Network (Mixture Model)

- Location profile as a multinomial distribution over locations.

Carol

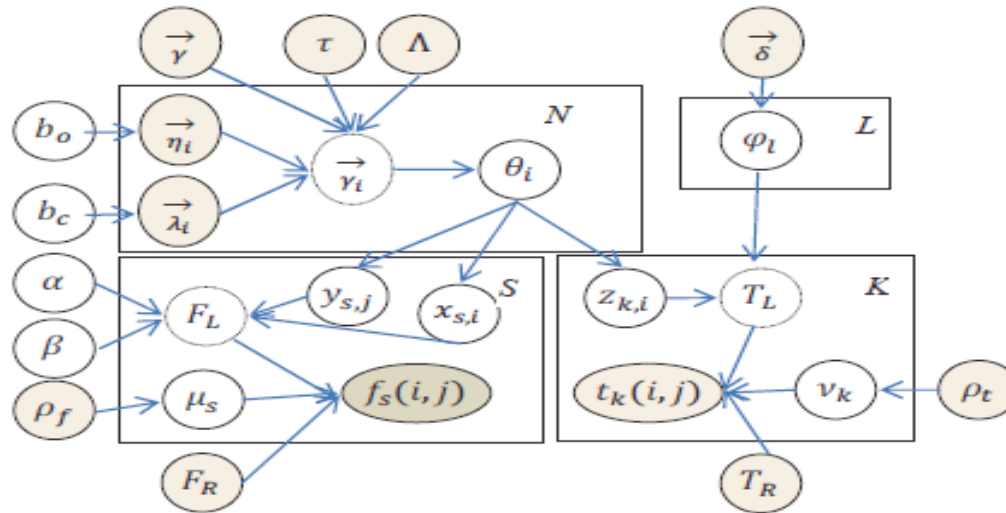


{Los Angeles 0.1, Austin 0.1, ... }

- Each observation is based on one particular location from her profile.

Location-based relationships	
Carol follows Lucy	<i>both Carol and Lucy studied at Austin</i>
Carol tweets Hollywood	<i>Carol lives Los Angeles</i>

Multiple Location Profiling based on Content and Social Network (Complete Model)



$$\begin{aligned}
 & P(\theta_{1:N}, \psi_{1:L}, \mu_{1:S}, x_{1:S}, y_{1:S}, f_{1:S}, \nu_{1:K}, z_{1:K}, t_{1:K} | \Omega) \\
 = & \prod_{i=1}^N P(\theta_i | \vec{\gamma}) \prod_{l=1}^L P(\psi_l | \vec{\delta}) \prod_{k=1}^K P(\nu_k | \rho_t) \prod_{s=1}^S P(\mu_s | \rho_s) \\
 & \prod_{s=1}^S (P(x_{s,i} | \theta_i) P(y_{s,j} | \theta_j) P(f_s \langle i, j \rangle | \alpha, \beta, x_{s,i}, y_{s,j}))^{1-\mu_s} \\
 & \prod_{k=1}^K (P(z_{k,i} | \theta_i) P(t_k \langle i, j \rangle | z_{k,i} = l, \psi_l))^{1-\nu_k} \\
 & \prod_{s=1}^S P(f_s \langle i, j \rangle | F_R)^{\mu_s} \prod_{k=1}^K P(t_k \langle i, j \rangle | T_R)^{\nu_k} \quad (4)
 \end{aligned}$$

Multiple Location Profiling based on Content and Social Network (Gibbs Sampling Algorithm)

The Collapsed Step: We first integrate $\theta_{1:N}$ and $\varphi_{i:L}$ so that we do not need to estimate them.

The Gibbs Sampling Step: Then, we sample from the posterior distribution of each unknown variable given other variables.

- $P(\mu_s | \mu_{-s}, \nu_{1:S}, x_{1:S}, y_{1:S}, f_{1:S}, z_{1:K}, t_{1:K}, \Omega),$
- $P(\nu_k | \nu_{-k}, \mu_{1:S}, x_{1:S}, y_{1:S}, f_{1:S}, z_{1:K}, t_{1:K}, \Omega),$
- $P(x_{s,i} | \mu_{1:S}, \nu_{1:S}, x_{-s:i}, y_{1:S}, f_{1:S}, z_{1:K}, t_{1:K}, \Omega),$
- $P(y_{s,j} | \mu_{1:S}, \nu_{1:S}, x_{1:S}, y_{-s:j}, f_{1:S}, z_{1:K}, t_{1:K}, \Omega),$
- $P(z_{k,i} | \mu_{1:S}, \nu_{1:S}, x_{1:S}, y_{1:S}, f_{1:S}, z_{-k:i}, t_{1:K}, \Omega),$

e.g.,

$$\begin{aligned} & P(x_{s,i} | \mu_{1:S}, \nu_{1:S}, x_{-s:i}, y_{1:S}, f_{1:S}, z_{1:K}, t_{1:K}, \Omega) \\ \sim & \frac{\varphi_{i,l} + \gamma_{i,l} - 1}{\varphi_i + \sum_{l=1}^L \gamma_{i,l} - 1} (d(x_{s,i}, y_{s,j})^\alpha)^{1-\mu_s} \end{aligned}$$

Tutorial Overview

- Event Detection for Twitter (80 min)
 - Event Detection using Tweet Content
 - Event Detection using Other External Sources
 - Applications of Event Detection
- Break (10 min)
- Event Description for Twitter (25 min)
 - Finding Best Phrase to Summarize an Event
 - Finding Event Types
 - Finding Event Timespans
 - Finding Event Credibility
 - Finding Event Locations
- User Profiling in Social Media (55 min)
 - Content-based Profiling
 - Network-based Profiling
 - Hybrid Approach
 - **Co-profiling Attributes and Relationships**
- Summary and Discussions (10 min)

Co-profiling: Integrate User Profiling with Other Tasks

- Integrate user profiling and other related tasks to achieve mutual enhancement
- We will focus on three studies
 - Integrate user profiling with entity matching
 - Integrate user profiling with relationship profiling

Co-profiling: Profiling User Location and Matching Entities in their Tweets (Overview)

- Nilesh Dalvi, Ravi Kumar, and Bo Pang. 2012. Object matching in tweets with spatial models. In *Proceedings of WSDM '12*. 43-52.
- Ideas
 - Entity Matching (Focus): Users locations can help entity (restaurant) matching in tweets.
 - E.g., “Bombay Grill” should be a restaurant in Sunnyvale other than Champaign if the user lives in Bay area .
 - Location Profiling: Entities mentioned in tweets should help predict users’ locations.
 - E.g., If a user mentions “Bombay Grill” in Champaign, he likely lives in UIUC.
- Results
 - Entity matching performance gains over geography-less models
 - Infer locations of the users accurately in practice

Co-profiling: Profiling User Location and Matching Entities in their Tweets (Model)

- The model contains two components

$$P(e, t, u) = P(t|e, u)P(e, u)$$

– Distance Model

- $P(e, u) \propto \alpha(u)\beta(e)(d_0 + d(e, u))^{-k}$
- $\alpha(u)$ is for interests of the user u for entity e .
- $\beta(e)$ is for entity popularity

– Language Model

- $P(t|e, u) = P(t|e)$
- $P(t|e) = \prod_{w \in t} (\theta \cdot P_e(w) + (1 - \theta) \cdot P_{lm}(w))$

Co-profiling: Profiling User Location and Matching Entities in their Tweets (Algorithm)

- EM Algorithm
 - E-step estimates the expectation of hidden variables (e.g., entities that match tokens in tweets)
 - M-step estimates the parameters (e.g., language model, distance model, and users' locations)
 - Some assumptions are used to simplify the estimation

Co-profiling: Profiling User Attributes and Relationship Types in the Ego Network (Overview)

- Rui Li, Chi Wang, and Kevin Chang. 2014. Co-profiling User Attributes and Relationship Types. WWW '14. 43-52.
- Ideas
 - Attribute Profiling (Focus): Different types of relationships propagate different attributes
 - E.g., Colleges should propagate from college mates, while occupation should be propagated from colleagues
 - Relationship Types Profiling: Relationship types can be identified by their shared attributes and network structure
 - E.g., A set of friends who are strongly connected and share occupation might be colleagues.
- Results
 - Dataset: Ego network from LinkedIn
 - Profiling attributes more accurately than the collective classification method
 - Profiling relationships more accurately than the clustering method

Please come to our talk in this conference to see details of our work.

gmanish@microsoft.com, ruililab@yahoo-inc.com, kcchang@illinois.edu

Summary for User Profiling in Social Media

- Both content of tweets and social network connections are useful for location prediction on Twitter
- Location of users, tweets and events is crucial for a large number of applications based on tweet feeds
- We discussed algorithms for using tweet content and the network for user profiling. We also discussed hybrid approaches along with co-profiling techniques.

Further Reading (1)

- B. Hecht, L. Hong, B. Suh, and E. H. Chi. Tweets from justin bieber's heart: the dynamics of the location field in user profiles. CHI 2011
- J. Eisenstein, B. O'Connor, N. A. Smith, and E. Xing. A latent variable model for geographic lexical variation. In Proceedings of EMNLP, 2010.
- Z. Cheng, J. Caverlee, and K. Lee, "You are where you tweet: a content-based approach to geo-locating twitter users," in ACM CIKM 2010.
- S. Kinsella, V. Murdock, and N. O'Hare, "I'm eating a Sandwich in Glasgow": Modeling Locations with Tweets," SMUC 2011
- W. Li, P. Serdyukov, A. P. de Vries, C. Eickhoff, and M. Larson, "The where in the tweet," CIKM 2011.
- Han Bo, Paul Cook, Timothy Baldwin. Geolocation Prediction in Social Media Data by Finding Location Indicative Words. COLING 2012.
- @Phillies Tweeting from Philly? Predicting Twitter User Locations with Spatial Word Usage. Hau-Wen Chang, Dongwon Lee, Mohammed Eltahery and Jeongkyu Leey. ASONAM 2012.
- N. Dalvi, R. Kumar, and B. Pang, "Object matching in tweets with spatial models," WSDM 2012.
- Jalal Mahmud, Jeffrey Nichols, Clemens Drews. Where Is This Tweet From? Inferring Home Locations of Twitter Users. ICWSM 2012.
- John Krumm, Rich Caruana, Scott Counts. Learning Likely Locations. UMAP 2013.
- Clodoveu A. Davis Jr., Gisele L. Pappa, Diogo Rennó Rocha de Oliveira, Filipe de L. Arcanjo. Inferring the Location of Twitter Messages Based on User Relationships. TGIS 2011.

Further Reading (2)

- A. Sadilek, H. Kautz, and J. P. Bigham. Finding your friends and following them to where you are. WSDM '12.
- Rui Li, Kin Hou Lei, Ravi Khadiwala, Kevin Chen-Chuan Chang. TEDAS: a Twitter Based Event Detection and Analysis System. ICDE 2012.
- Rui Li, Shengjie Wang, Hongbo Deng, Rui Wang, Kevin Chen-Chuan Chang: Towards social user profiling: unified and discriminative influence model for inferring home locations. KDD 2012.
- Rui Li, Shengjie Wang, Kevin Chen-Chuan Chang. Multiple Location Profiling for Users and Relationships from Social Network and Content. VLDB 2012.
- Rui Li, Chi Wang, and Kevin Chang. 2014. Co-profiling User Attributes and Relationship Types. WWW '14. 43-52.
- S. Yardi and D. Boyd. Tweeting from the town square: Measuring geographic local networks. In ICWSM, 2010.
- Discovering Geographical Topics In The Twitter Stream. Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alex Smola, Kostas Tsioutsoulouklis. WWW 2012
- Zhiyuan Cheng, James Caverlee, Kyumin Lee, Daniel Z. Sui. Exploring Millions of Footprints in Location Sharing Services. ICWSM 2011.
- <http://mashable.com/2009/06/08/twitter-local-2/>
- <http://www.slideshare.net/pkitano/the-local-business-owners-guide-to-twitter>
- Detecting Forest-fires: Bertrand De Longueville, Robin S. Smith, and Gianluca Luraschi. "OMG, from here, I can see the flames!": a use case of mining location based social networks to acquire spatio-temporal data on forest fires. LBSN '09
- Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. WWW '10
- Sílvia S. Ribeiro, Jr., Clodoveu A. Davis, Jr., Diogo Rennó R. Oliveira, Wagner Meira, Jr., Tatiana S. Gonçalves, and Gisele L. Pappa. Traffic Observatory: A System to Detect and Locate Traffic Events and Conditions using Twitter. LBSN '12

Tutorial Overview

- Event Detection for Twitter (80 min)
 - Event Detection using Tweet Content
 - Event Detection using Other External Sources
 - Applications of Event Detection
- Break (10 min)
- Event Description for Twitter (25 min)
 - Finding Best Phrase to Summarize an Event
 - Finding Event Types
 - Finding Event Timespans
 - Finding Event Credibility
 - Finding Event Locations
- User Profiling in Social Media (55 min)
 - Content-based Profiling
 - Network-based Profiling
 - Hybrid Approach
 - Co-profiling Attributes and Relationships
- **Summary and Discussions (10 min)**

Summary

- We discussed these key components of an analytics platform for microblogging systems
 - Event Detection
 - Event Description
 - User Profiling
- Lots of our components are critical for such a platform, which we did not cover in this tutorial
 - Structured entity extraction
 - Sentiment analysis
 - Predictive analysis
 - Correlations with other media types like news and blogs
 - Influence Analysis
 - Visualization
 - Temporal Analysis

Thanks!

