Real-time Local Event Detection in Geo-tagged Tweet Stream
What is a Local Event?

• A local event is certain *unusual activity* with *a considerable number of participants* that gather at *a specific place and time*.
  - E.g., parade, riot, sport game, concert, accident, disaster.

• The local event detection task is important for various applications
  - crime and disaster monitoring
  - personalized activity recommendation
Why Geo-tagged Tweet Stream?

- The geo-tagged tweet stream is an invaluable source for local event detection because of its
  - **sheer size:** more than 10 million geo-tagged tweets are posted every day;
  - **rich information:** any geo-tagged tweet contains a location, a timestamp, and text descriptions.
  - **real-time nature:** a geo-tagged tweet reflects the user’s activity right on the spot.
Our Goal

- Given the geo-tagged tweet stream, we aim to
  - detect all local events in any query time window (batch mode);
  - update the result list in real time as the query window shifts continuously (online mode).
Challenges

1. Extracting interpretable events from overwhelming noise.
   - Raw tweets are usually short and noisy. Hence, truly interesting events are buried in massive noise.

2. On-line and real-time detection.
   - To allow for timely actions, local events need to be fast detected despite the sheer size of the tweet stream.

3. Self-adaptive to the highly evolving stream.
   - The tweet stream is evolving fast. A detector should require little human intervention and be able to automatically capture different events in different stages.
Why Are Existing Methods Insufficient?

• Most event detection methods are designed for detecting *global events*

  ‣ They can successfully detect events that are bursty in the entire stream;

  ‣ Local events are “bursty” in a small region and involve a limited number of tweets.

2011 ICWSM. Event detection in twitter.
2012 CIKM. Twevent: segment-based event detection from tweets.
Why Are Existing Methods Insufficient?

- A few methods for local event detection have been proposed
  - They either cannot extract interpretable events because of overlooking the correlations between keywords; or are incapable of detecting local events in real time.

2009 CIKM. Event detection from Flickr data through wavelet-based spatial analysis.
2013 PVLDB. EventTweet: Online localized event detection in the twitter stream.
Our Insight

• A local event usually leads to many related tweets around the location (**a geo-topic cluster**).

• But **a geo-topic cluster is not necessarily a local event:**
  
  ‣ It may be a routine activity in that region (e.g., shopping).
Overview of Our Method

• We propose a reference-based local event detection method:
  ‣ an **on-line module** finds geo-topic clusters in the query time frame, and regard them as candidate events;
  ‣ a **ranking module** summarizes the routine activities in different regions to filter non-event candidates.
Candidate Event Generation

• The online module finds candidates in the query time frame:
  ‣ Geo-topic cluster: a group of tweets that are geographically close and semantically relevant.
Candidate Event Generation

• To effectively find candidates, we face several challenges:
  ‣ How to simultaneously model the geographical and semantic similarities between tweets?
  ‣ How to capture the correlations between different entities?
  ‣ How do we group the tweets without knowing the number of clusters in advance?
Candidate Event Generation

- A tweet gets an **authority score** from neighbor tweets where
  - the geographical authority is captured by kernel function;
  - the semantic authority is captured by random walk on the keyword co-occurrence graph.
Candidate Event Generation

- A **hub** is a prominent tweet (authority maxima) that has many tweets that are similar to it both geographically and semantically.

- Intuition: the spot where the event occurs acts as a source that produces many relevant tweets around it; the closer we are to the hub, the more likely we observe relevant tweets.
Candidate Event Generation

• Now the task is to find all the hubs in the geo-topic space.

• We design an authority ascent process to find all hubs.

• A hub attracts similar tweets to form geo-topic clusters.
The Ranking Module

- We need to summarize the typical activities (background knowledge) in different regions, and use such knowledge to filter uninteresting clusters.

CluStream is a good fit for this task :)

Reference Time Frame
Query Time Frame
CluStream for Activity Summarization

- CluStream continuously clusters the stream into a number of micro-clusters, and stores the clustering results as different snapshots.
Experimental Settings

- Data: 9M geo-tagged tweets in New York during 3 months.
- Task: 60 queries with different durations (3h, 4h, 5h, 6h), find top-5 local events in each query window.
- Compared Method: EvenTweet (PVLDB’13), Wavelet (CIKM’09)
- Evaluation: The crowdsourcing platform CrowdFlower
Example Local Events

Example 1:

- **Time:** 2014-10-07
- **Tweets:**
  - And we're back. Let's go Brooklyn! #brooklynnets #nets #barclays #ballinshotcallin @Barclays Center http://t.co/fx95cxxxBmu
  - It's good to be back in Brooklyn with the @brooklynnets!!! Thank you @ipavlova!!! #brooklynnets?? http://t.co/JKxcaK6ho2
  - 31-30 Brooklyn in 2Q. @MaccabiElectra Kicking Ass!!!!!
  - I'm at Barclays Center - @brooklynnets for Tel Aviv Maccabi vs Brooklyn Nets in Brooklyn, NY https://t.co/jAXkTOQyL7
  - I'm at Barclays Center - @brooklynnets for Tel Aviv Maccabi vs Brooklyn Nets in Brooklyn, NY https://t.co/xW0fK2MuoT

Example 2:

- **Time:** 2014-09-12
- **Tweets:**
  - Bromeo (@Central Park @SummerStage in New York, NY) https://t.co/fx4OnedlDp
  - I'm at Central Park @SummerStage in New York, NY https://t.co/iNj7AjjVsn http://t.co/zpRnJKHlAE
  - 3 yrs later and i finally made it to a chroomeo concert @ Central Park Summer Stage http://t.co/1WxwmDpAi9
  - Feels Good! @chroomeo @ Central Park summer stage! Next single #comealive #atlanticrecords #whitewomen??
  - #Chromeo #NYC @ Central Park Summer Stage http://t.co/o9T83ZS6bk
Precision
Running Time

The graph shows the running time of different methods (HubSeek, EvenTweet, Wavelet) as a function of the number of tweets. The x-axis represents the number of tweets, ranging from 3.0 to $10^3$, and the y-axis represents the time in seconds, ranging from 0 to 16.

- **HubSeek**: A solid blue line with blue dots.
- **EvenTweet**: A red dashed line with red squares.
- **Wavelet**: A black dotted line with black diamonds.

As the number of tweets increases, the running time for all methods also increases, but the slopes differ, indicating different performance characteristics.
Batch v.s. Online

![Graph comparing Batch and Online performance over updates](image-url)
Thanks!