Querying and Mining Geospatial Social Media Data (Part 2)

Gao Cong (丛高)
Nanyang Technological University

Acknowledgement: Many slides are from a SIGMOD’15 tutorial (by Gao & Christian)
### Overview

<table>
<thead>
<tr>
<th>Data exploration</th>
<th>Querying &amp; Mining</th>
<th>Mining &amp; Analytics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Region search</strong>, e.g., burst region, diversified region</td>
<td><strong>Querying</strong>: Subscription queries, snapshot queries</td>
<td><strong>User behaviour modeling</strong></td>
</tr>
<tr>
<td><strong>Region exploration</strong>, e.g., topic exploration</td>
<td><strong>Mining</strong>, e.g., entity extraction, geotagging</td>
<td><strong>Recommendation, personalized query, mobility prediction</strong></td>
</tr>
<tr>
<td><strong>Other queries</strong>: e.g., RNN</td>
<td><strong>Indexing streams and queries</strong></td>
<td><strong>Sentiment analysis</strong></td>
</tr>
<tr>
<td><strong>Beyond single object granularity</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Basic queries</strong>, e.g., Boolean Range Query, Boolean kNN Query</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Spatial index and textual index</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Distributed system</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Static geo-textual data</strong></td>
<td><strong>Streaming geo-textual data</strong></td>
<td><strong>Additional dimensions: user, time, category, review</strong></td>
</tr>
</tbody>
</table>

- **foursquare**, **yelp**, **Google places**, **Twitter**, **Instagram**
- **Sentiment analysis**
- **Region search**, **Region exploration**
- **Basic queries**
- **Spatial index and textual index**
- **Beyond single object granularity**
- **Other queries**
- **Data exploration**
- **Querying**
- **Querying & Mining**
- **Mining & Analytics**
- **Distributed system**
Outline

• Querying geo-textual streams
  ■ Boolean publish/subscribe query
  ■ Top-k publish/subscribe query
  ■ Moving publish/subscribe query
  ■ Other query
• Exploring geo-textual data
• Summary and challenges
Motivation of publish/subscribe

• Streaming geo-textual data (e.g., geo-tagged tweets) often has the quickest first-hand reports of:
  
  - Breaking news
    - E.g., Osama Bin Laden’s death
  
  - Disasters
    - E.g., Bomb blast in Mumbai in Nov. 2008, flooding of Red River Valley in Mar 2009
  
  - Public Health – Disease Outbreaks
    - E.g., Norovirus outbreak at universities, influenza epidemic

Applications of Publish/Subscribe

• Applications
  ■ Location-based services, e.g., Location-aware event, Local news subscription, Location-based E-coupon
    ✦ location-based and keyword-based requirements
    ✦ Real-time requirement (instant feeding)
  ■ Annotation of Points-of-Interest (POIs) with social media feeds: Bridge dynamic (streaming) world and offline world

• Challenges:
  ➢ High arrival rate of geo-textual objects.
    • Over 10 million new tweets with coordinates per day \(^1,2\)
    • Over 100 million new tweets with semantic locations per day \(^1,2\)
  ➢ A large number of subscription queries.

Boolean Range Subscription Query

- Boolean Range Subscription (BRS) Query

\[ q = (\psi, r) \]

- \( \psi \): a set of keywords connected by AND or OR semantics
  
  (dengue AND fever, vomit OR poisoning)

- \( r \): the query region (within 1 km from Hyatt Regency in SF)

- To trigger the action of “pushing”, the following conditions should be satisfied:
  - The Boolean expression, as indicated in \( \psi \), should be satisfied by the object terms.
  - The location of object should be within the query region \( r \).
Boolean Range Subscription Query

- Problem: answering a stream of BRS queries in real time on a stream of geo-textual objects continuously.
Location-Aware Publish/Subscribe Model

Publisher

Subscription Queries Index

geo-textual object

Subscription Query

register

published result

Subscription Query

Subscription Query

Published result

Subscription Query

Subscription Query

Subscription Query

Subscription Query

Subscription Query

Subscription Query

Subscription Query

index

Camille Carnevale
@ccarnevaleNY

I forgot how good a shot of espresso actually feels.

Manhattan, NY

FAVORITE

1
Boolean Publish/Subscribe Solution

• High-level framework
  ■ Organize subscription queries s.t. each group can be processed together
  ■ Indexing Subscription queries, which is also a stream
• Quad-tree + Inverted file for indexing (Chen et al.)
  ■ Handle Boolean Expression with AND, OR.
• R-tree: Each node records the set of keywords from descend node (Li et al.)
  ■ Handle Boolean Expression with AND
• AP-tree: Divide the space either by words, or by space (Wang et al)
  ■ Handle Boolean Expression with AND

Chen et al. *An efficient query indexing mechanism for filtering geo-textual data*. SIGMOD’13
Li et al. *Location-aware publish/subscribe*. KDD’13
Wang et al. *AP-Tree: Efficiently Support Continuous Spatial-Keyword Queries Over Stream*, ICDE’15
Other Definition of Subscription Queries?

• An interest region + a Boolean expression
  • Such a query may receive very few (or too many) matching geo-textual objects.
  • It may be difficult for a subscriber to specify the query keywords, and especially the size of a spatial region when they are used as Boolean filters

• Users would prefer to be updated with a few most relevant geo-textual objects in terms of distance, text similarity, and recency
**Temporal Spatial-Key Word Subscription (TaSK) Query**

- **TaSK Query**
  - A set of keywords: *food poisoning vomiting*
  - Location: *Hyatt Regency in SF*
  - $k$ - the number of results: **10**
  - **Objective:** Maintain up-to-date top-$k$ most **relevant** results for each TaSK query over a stream of geo-textual objects.

- **How to measure ‘relevance’?**
  - $S_{tsk}$: **Temporal spatial-keyword score**, a combination of distance (spatial), text relevance (keyword), and object freshness (temporal).

$$
S_{tsk} = S_{sk}(q, o) \cdot D^{-(t_e - o \cdot t_c)}
$$

$$
S_{sk} = \alpha S_{dist}(q, \rho, o, \rho) + (1 - \alpha) S_{rel}(q, \psi, o, \psi)
$$

- **$S_{sk}$**: Spatial-Keyword Score
  - $S_{dist}$: Score of spatial proximity
  - $S_{rel}$: Score of text relevance

- $D^{At}$: Exponential Decaying Factor
Main Idea

👎 For each new geo-textual object we compute its ranking score w.r.t. each query, and update the current top-kth result.
  • Scores of the current top-k results changes!

👍 How to represent, group, and index subscription queries such that queries in one group can be evaluated simultaneously to reduce the computation.

Chen et al. Temporal Spatial-Keyword Top-k Publish/Subscribe. ICDE’ 15
Moving Subscription Queries

- **Moving Query**: Region + Boolean Expression
- **Solution**: Reduce communication overhead by safe region

Guo et al. *Location-Aware Pub/Sub System: When Continuous Moving Queries Meet Dynamic Event Streams*. SIGMOD’15
Other Work

- SKYPE: Top-k Spatial-keyword Publish/Subscribe Over Sliding Window, Wang et al. VLDB’16
  - Sliding Windows setting: Each geo-textual object arrives and expires
  - New techniques to prune the search space for maintaining top-k results

- A location-aware publish/subscribe framework for parameterized spatio-textual subscriptions. Hu et al. ICDE’15
  - Subscription queries: Defined based on both text similarity and spatial proximity. Each subscription has a pre-given threshold

Objective

• Distributed Publish/Subscribe System over a Spatio-Textual Data Stream
  – Running STS queries on streaming spatio-textual objects
    • High input speed, e.g., over 100,000 tuples/second
    • Large volume of STS queries, e.g., over 1 million STS queries
    • Frequent insertions/deletions of STS queries
  – Optimal performance
    • High throughput
    • Low latency
  – Dynamic load adjustment
    • Adapting to the changing workload
**Existing Work**

- **Limitations of existing work**
  - Centralized spatial-keyword publish/subscribe systems
    - A single server cannot handle a large amount of streaming data, e.g., Hu, et al. (ICDE 2015)
  - Distributed content-based publish/subscribe systems
    - Lacking support for the spatio-textual data, e.g., Barazzutti, et al. (ICDCS 2014)
  - Distributed systems for spatial data
    - Not considering the workload composed of frequent insertions/deletions of subscription queries and processing streaming spatio-textual objects, e.g., Aly, et al. (VLDB 2015)
Overviews

• The distributed publish/subscribe system for the spatio-textual data stream
  • Consider the workload composed of frequent insertions/deletions of subscription queries and processing streaming spatio-textual objects
  • Different workload partitioning strategies can invoke different amount of total workload to the system and we solve the optimal workload partitioning problem considering both of minimizing the total amount of workload and load balancing

• Efficient dynamic load adjustment algorithms considering
  • Reducing the total amount of workload
  • Invoking small migration cost

• Extensive experiments on Amazon EC2 with real data
The workload partitioning strategy significantly affects the system performance.
Problem Definition

- The load of a worker
  - Spatio-textual objects $O$
  - Query insertions $Q^i$ and query deletions $Q^d$
  - $L_i = c_1 \cdot |O_i| \cdot |Q^i_i| + c_2 \cdot |O_i| + c_3 \cdot |Q^i_i| + c_4 \cdot |Q^d_i|$

The load of a worker is positive correlated to the number of objects and the number of queries.
Problem Definition

- Optimal workload Partitioning
  - Compute \((S_i, T_i)\) for each worker \(w_i\) \((1 \leq i \leq m)\), where \(m\) is the number of workers
  - \(\bigcup_{1 \leq i \leq m} S_i = S, \bigcup_{1 \leq i \leq m} T_i = T\), \(S\) denotes the global space, and \(T\) denotes the lexicon
  - **Objective:** Minimizing \(\sum_{i=1}^{m} L_i\), subject to the constraint that 
    \(\forall 1 \leq i < j \leq m, \frac{L_i}{L_j} \leq \sigma\), where \(\sigma\) is a small constant value larger than 1

Minimize the total amount of workload

Load balancing
Hybrid Partitioning

- Observation
  - Different regions have different data distributions

- $o_1$: I want to watch the EuroCup final
- $o_2$: Movie Civil War is not that good

Diagram:
- Worker 1:
  - Query $q_1$: transgenosis
  - Object $o_1$
  - 2 queries and 1 object

- Worker 2:
  - Query $q_2$: movie
  - Object $o_2$
  - 2 queries and 1 object

Diagram:
- Region $r_1$
  - Query $q_1$: transgenosis
  - 1 query and 1 object

- Partitioning by space
- Partitioning by text
Hybrid Partitioning

- Observation
  - Different regions have different data distributions

\( o_3 \): Lebron is better than Kobe.
\( o_4 \): I like Kobe more than Lebron

Region \( r_2 \)

Partitioning by space
- Worker 1: kobe
- Worker 2: movie

Partitioning by text
- Worker 1: kobe
- Worker 2: lebron

1 query and 1 object
1 query and 1 object
1 query and 2 objects
1 query and 2 objects
Hybrid Partitioning

• Phase I
  – Divide the space into subspaces where text-partitioning performs better and subspaces where space-partitioning performs better

• Phase II
  – Compute the partitions based on the output of Phase I
Hybrid Partitioning

- The algorithm runs by computing an $kd^t$-tree

Partition the workload to 3 workers ($\sigma = 2$)

Phase 1

$N_1$: The objects and queries have high text similarity, i.e., better use space-partitioning

$N_2$: The objects and queries have low text similarity, i.e., better use text-partitioning

**Objects and Queries**

- $o_1$: Opening a restaurant with a cinema nearby
- $o_2$: A shopping mall having a cinema, restaurants, etc.
- $o_3$: An amazing hotel with good environment
- $o_4$: Opening a Chinese restaurant

**Queries**

- $q_1$: restaurant
- $q_2$: restaurant
- $q_3$: cinema
- $q_4$: cinema AND dinner
- $q_5$: iphone
- $q_6$: mac
- $q_7$: surface

**Trees**

- $R_1$
- $R_2$
Hybrid Partitioning

- The algorithm runs by computing an $kd^t$-tree

Partition the workload to 3 workers ($\sigma = 2$)

Phase 2

- $q_1$: restaurant
- $q_2$: restaurant
- $q_3$: cinema
- $q_4$: cinema AND dinner
- $q_5$: iphone
- $q_6$: mac
- $q_7$: surface

$o_1$: Opening a restaurant with a cinema nearby
$o_2$: A shopping mall having a cinema, restaurants, etc.
$o_3$: An amazing hotel with good environment
$o_4$: Opening a Chinese restaurant
Hybrid Partitioning

- The algorithm runs by computing an $kd^t$-tree

<table>
<thead>
<tr>
<th>$O_1$</th>
<th>$O_2$</th>
<th>$O_3$</th>
<th>$O_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_2$: restaurant</td>
<td>$q_3$: cinema</td>
<td>$q_4$: cinema AND dinner</td>
<td>$q_5$: iphone</td>
</tr>
<tr>
<td>$q_1$: restaurant</td>
<td></td>
<td></td>
<td>$q_6$: mac</td>
</tr>
<tr>
<td>$q_7$: surface</td>
<td></td>
<td></td>
<td>$O_4$</td>
</tr>
</tbody>
</table>

Partition the workload to 3 workers ($\sigma = 2$)

Phase 2

Partition by space

$R_2$

$R_3$

$R_4$

$$\frac{\text{Load}(w_o)}{\text{Load}(w_l)} = 3 > \sigma$$

$o_1$: Opening a restaurant with a cinema nearby

$o_2$: A shopping mall having a cinema, restaurants, etc.

$o_3$: An amazing hotel with good environment

$o_4$: Opening a Chinese restaurant
Hybrid Partitioning

- The algorithm runs by computing an $kd^t$-tree

Partition the workload to 3 workers ($\sigma = 2$)

Phase 2

\[ \frac{\text{Load}(w_o)}{\text{Load}(w_l)} = 1.5 < \sigma \]

Partition by space

Partition by text

$q_1$: restaurant
$q_2$: restaurant
$q_3$: cinema
$q_4$: cinema AND dinner
$q_5$: iphone
$q_6$: mac
$q_7$: surface

$N_1$  $N_2$

$N_3$  $N_4$  $N_5$  $N_6$

$R_3$  $R_4$  $R_2$  $R_2$

$L \approx 2$  $L \approx 2$  $L \approx 1$  $L \approx 2$

$O_1$: Opening a restaurant with a cinema nearby
$O_2$: A shopping mall having a cinema, restaurants, etc.
$O_3$: An amazing hotel with good environment
$O_4$: Opening a Chinese restaurant

$O_5$: Opening a restaurant with a cinema nearby
$O_6$: A shopping mall having a cinema, restaurants, etc.
$O_7$: An amazing hotel with good environment
Hybrid Partitioning

- The $kd^t$-tree is transformed to a $grid^t$ index

```plaintext
\begin{align*}
\text{root} & \quad \rightarrow \quad g_1 \quad \rightarrow \quad g_2 \\
N_1 & \quad \rightarrow \quad N_2 \\
N_3 & \quad \rightarrow \quad N_4 \quad \rightarrow \quad N_5 \quad \rightarrow \quad N_6 \\
R_3 & \quad \rightarrow \quad R_4 \quad \rightarrow \quad R_2 \quad \rightarrow \quad R_2 \\
\{\text{surface}\} & \quad \{\text{iphone, mac}\} \\
\text{worker} & \quad \rightarrow \quad \text{worker} \quad \rightarrow \quad \text{worker} \\
&w_1 \quad w_1 \quad w_2 \\
\end{align*}
```
Outline

• Introduction & Objective
• System Architecture
• Algorithms
  – Hybrid Partitioning
  – Dynamic Load Adjustment
• Experiments
• Conclusions
Dynamic Load Adjustment

• The load of workers may change over time due to the changing data distribution

• The dynamic load adjustment is triggered
  – Migrate STS queries in the unit of one cell in the \(grid^t\) index
  – Phase I: Check whether some cells in different workers can be split or merged to reduce the total workload
  – Phase II: Minimum cost migration
Minimum Cost Migration

- Minimum Cost Migration
  - Consider a worker $w_o$ with the set of cells $G_o$
  - Compute a subset of cells $G_s$ in $G_o$ to be migrated
    - $G_s = \arg\min_{G_s \subseteq G_o} \sum_{g \in G_s} S_g$ such that $\sum_{g \in G_s} L_g \geq \tau$
    - Where $S_g$ is the total size of the queries in cell $g$ and $\tau$ is the amount of load to be migrated

- Algorithms
  - Dynamic programming algorithm
    - High time complexity and large memory usage
  - Greedy algorithm
GR Algorithm

- **Greedy algorithm**
  - Sort the cells in increasing order of $\frac{|S_g|}{L_g}$
  - $\tau$ denotes the amount of load to be migrated

\[
\begin{align*}
\sum_{i=1}^{t} \sum_{g \in G_{S_i}} L_g < \tau \quad \text{and} \quad \forall g' \in G_{L_t}, \sum_{i=1}^{t} \sum_{g \in G_{S_i}} L_g + L_{g'} & \geq \tau
\end{align*}
\]

All candidate results:

- $G_{S_1} \cup \{g\}$, where $g \in G_{L_1}$
- $G_{S_1} \cup G_{S_2} \cup \{g\}$, where $g \in G_{L_2}$
  
- $\vdots$
  
- $\bigcup_{1 \leq i \leq t} G_{S_i} \cup \{g\}$, where $g \in G_{L_t}$
  
- $\vdots$
Outline

- Background and motivation
- Spatial keyword queries on static geo-textual data
- Querying geo-textual streams
- Exploring geo-textual data
  - Region search
    - Most dense region
    - Best region search
    - Best region search on road network
    - Interactive exploration
  - Region exploration
- Summary and challenges
MaxRS problem:

- Input: a set of spatial object, and a rectangle of a given size.
- Output: the position of the rectangle such that the sum of the objects covered by the rectangle is maximized.

**In-memory solution:**
Imai and Asano, Journal of Alg. 1983
Nandy et al. Computers & Mathematics with Applications 1995

**Secondary memory solution:**
Region Search on Road Networks

Dining, NYC
Region Search on Road Networks

Shopping, NYC
Problem Formalization

- Road network graph $G$
  - A node represents a road junction point or a location, associated with a set of keywords
  - An edge represents a road segment
- Nodes are weighted w.r.t. Query
  - Relevance to the query
  - Other query-independent weights (e.g., popularity or rating) are also possible
- Query Region $R$
  - A connected subgraph of $G$
  - Shape of the region is not easily described by predefined shape and size.

Cao et al. *Retrieving regions of interest for user exploration*. VLDB’14
“Hot Region” Query

- \( q = \langle \lambda, \psi, \Delta \rangle \)
  - \( \lambda \): a rectangular query range
  - \( \psi \): keywords
  - \( \Delta \): a road segment length constraint
- Retrieves the region with largest weight given the length constraint and the query range
- Example: \( \lambda = \) the whole graph, \( \Delta = 6 \)
- Result: \(<v_2, v_4, v_5, v_6>\)
Region Search: Most Diversified Region

Query for a region of a given size that has the most diverse collection of attractions

- There are more POIs in region $r_1$ (5 POIs)
- There are more types of POIs in region $r_2$ (3 types)
Region Search

- Users would like to search for “best” region of a given size to explore the data
  - Different users prefer regions of different sizes to explore.
  - Different users prefer “best” regions based on different criteria
    - Most dense region: SUM as aggregation function (Choi et al. VLDB02, Imai and Asano, Journal of Alg. 1983)
    - More general aggregation function—**submodular monotone function**
Submodular function

- In the example, user specifies the size of query rectangle, and an aggregation function.
- The aggregation score function:
  - It’s submodular, with a “diminishing return” property.
Example:
Given $L(o_1) = \{a, b, c\}, L(o_2) = \{a, d\}, L(o_3) = \{c\}$, consider function $f(X) = |\bigcup_{o \in X} L(o)|$, we have

$$1 = f(\{o_1, o_2, o_3\}) - f(\{o_2, o_3\}) < f(\{o_1, o_3\}) - f(\{o_3\}) = 2$$

Marginal gain from adding $o_1$ to $\{o_2, o_3\}$  
Marginal gain from adding $o_1$ to $\{o_3\}$

The marginal gain from adding an element to an input set decreases as the size of the input set increases
Problem definition

• Given a set of spatial objects $O$, a submodular monotone function $f: 2^O \rightarrow \mathbb{R}$, and the size $a \times b$ of query rectangle,

• The best region search (BRS) problem aims at finding a location $p$ from space $P$

$$p = \arg \max_{p \in P} f(O_{r_p}^{a,b})$$

Here

$f(O_{r_p}^{a,b})$ is the aggregate score of the region of size $a \times b$ centered at $p$

• Challenges of solving such a general problem
  ■ Efficiency: Infinite points in the space to consider.
Exact Solution

• Reduction to Submodular Weighted Rectangular Intersection (SIRI) problem.
  - Infinite candidate points $\Rightarrow O(n^2)$ candidates

(a) The BRS problem
(b) The SIRI problem

Nandy et al. Computers & Mathematics with Applications 1995
Exact Solution

• Main idea:
  1. Cut the space into slices.
  2. In each slice, we prune maximal slabs with small upper bounds.
  3. For the remaining maximal slabs, use a sweep line to scan from left to right to find maximal regions.

• The complexity from $O(n^2)$ to $O(n.n_s)$
  - $n_s$ the number of maximal slabs. In practice it is small, but worse case is still $n$
Approximate Solution

• However, the exact solution takes $O(n^2)$ time in the worst case.

• Thus, we propose an approximate solution with a constant approximation ratio.

• Main idea:
  1. Select a smaller set $T$ of spatial points to represent the original set $O$ of spatial objects.
  2. Generate a new instance of BRS problem on $T$.
  3. Find the exact result of the new instance.
Approximate Solution: c-cover

- A set of spatial points $T$ is a **c-cover** of the set of spatial objects $O$ iff for any object $o_i \in O$, there exists one point $t \in T$ such that $o_i$ is inside the $ca \times cb$ rectangular region centered at $t$.

Example: The set of **black nodes** is a c-cover of **white nodes**.
Approximate Solution: c-cover

Theorem: Finding a minimum c-cover is NP-hard.

To select a c-cover:
- Greedy algorithm: with approximation ratio, low efficiency \( O(n^2\log n) \)
- A quadtree-based heuristic method \( O(n) \)

A quadtree-based heuristic method
Approximate Solution: New Instance

With the selected spatial points, we need to construct a new instance of the BRS problem.

• New aggregation score
  ■ Each spatial object in $\mathcal{O}$ is represented by a spatial point in $T$.
  ■ New aggregation score is based on the representations.

• New size of the query rectangle
  ■ We issue a new query of size $(1 - c)a \times (1 - c)b$. 
Approximate Solution

• Complexity
  - It takes $O(n + n^t \times n^t_s)$ time to find the result, where $n^t$ is the number of spatial points in $T$ and $n^t_s$ is the number of slabs that are actually searched in the new instance.

• Approximation ratio
  - When $c = \frac{1}{2}$, the approximation ratio is $\frac{1}{9}$.
  - When $c = \frac{1}{3}$, the approximation ratio is $\frac{1}{4}$. 
Interactive Data Exploration Using Semantic Windows

Motivation: Users perform various exploration tasks interactively.

Setting:
- A set of n-dimensional objects \( S \), a grid is defined on top of \( S \).
- A window is a union of adjacent cells that constitutes an n-dimensional rectangle.

Problem statement
- Conditions:
  - Content-based: aggregation of the objects inside the window
  - Shape-based: the shape of the window
- Given a set of conditions, the SW query finds all possible windows satisfying all conditions, and return online results quickly

Solution
- Divide space into grid, and enumerate windows (from cells)
- Data-driven search, based on a sample to guide the search

Looking for bright clusters of stars

Kalinin et al. *Interactive data exploration using semantic windows*. SIGMOD ‘14
Outline

• Background and motivation
• Spatial keyword queries on static geo-textual data
• Querying geo-textual streams
• Exploring geo-textual data
   Region search
   Region exploration
     Top-k Spatio-temporal term search
     Selectivity estimation
     Geo-spatial event/topic exploration

• Summary and challenges
Top-$k$ Spatio-Temporal Term Querying

- Example

Marathon runners...
Cambridge Science Festival
Preparing for marathon...
Boston marathon...
Film festival...
International Film Festival...

Spatio-temporal streaming posts

- Problem

**Input:** a top-$k$ term query with region $R$ and timespan $[t_b, t_e]$

**Output:** top-$k$ terms

$t_1$ $t_2$ ... $t_{kg}$ ... $t_{k-1}$ $t_k$

Top $k_g$ Exact results
Approximate results

Selectivity Estimation on Geo-textual Streams

• Input
  - A spatio-textual object stream

• Query
  - Query region ($q.R$)
  - A set of query keywords ($q.T$)

• Task
  - Estimate the number of objects that fall in query region $q.R$ and contain the query keywords $q.T$.

• Solution
  - Three different types of summary for estimation (ASP-tree, KMV, BN)

Twitterstand

- Twitterstand: Continuously acquire breaking news
  - Online clustering algorithm to generate breaking news
  - Users can specify regions of interest and topic (Boolean filter)
  - Periodically update users with new clusters of tweets

Exploring Spatio-temporal Events

- Motivation - Zoomable event cube

- Problem
  - **Input**: $k$ and a selected spatio-temporal cell with some granularity
  - **Output**: Top-$k$ events in the cell:
    1. \{#BostonMarathon, #BostonMarathon2016\}
    2. \{#FilmFest\}
    3. \{#ScienceFestival\}

Feng et al. *STREAMCUBE: hierarchical spatio-temporal hashtag clustering for event exploration over the Twitter stream*. In ICDE’15
Exploring Spatio-temporal Events

- **Step 1. Event detection using Hashtag clustering**
  - Each hashtag is represented as a **bag-of-words** and a **bag-of-hashtags** that co-occurs with the target hashtag within a document.
  - Each event is a **set of hashtags** with similar semantics.
  - Threshold hold based clustering.

- **Step 2. Event aggregation to coarser granularities**
- **Step 3: Event ranking (popularity, Localness, burstness, etc.)**
Geo-spatial Event Detection

Marathon runners...
Cambridge Science Festival
Preparing for marathon...
Boston marathon...
Film festival
International Film Festival...

Research Problem:
• How to cluster tweets
• Extracting features for each cluster to determine if it is an event

Spatio-temporal streaming posts

Walther et al. Geo-spatial event detection in the Twitter stream. ECIR’ 13
The topic exploration problem:
- Given a collection of spatio-temporal documents $D$;
- **Input**: query rectangle region $R$ and timespan $[t_b, t_e]$;
- **Output**: $K$ topics in the region and timespan.

Topic models, e.g., Latent Dirichlet Allocation (Blei, 2005): 

Model Assumptions:
- Document $\rightarrow$ a probabilistic distribution over topics
- Topic $\rightarrow$ a probabilistic distribution over words.

How it works:

Alan Turing was a pioneering English computer scientist, ..., he broke German ciphers, ..., Turing’s parents enrolled him at St Michael’s school... he was prosecuted...
Challenges

Efficiency issue:

• Training topic models online is time consuming
  – The complexity of training LDA is $O(|W|KI)$
  – 3-months tweets (250M) in NYC → 13.85 hrs training time
  – In real scale, the number of tweets in a user specified region and timespan will be large

• User could consider to modify the query in an exploratory manner, and we could not train topic models for each query offline in advance.
Main idea: Organize the documents into a hierarchy structure and *pre-train* some models on the cells to accelerate online training.

Problem 1: Combining algorithm with bounded error

Problem 2

Machine learning techniques + Data management principles
Problem 1: Combining Pre-trained Topic Models

- **Problem**: Given two pre-trained topic models on cells $C_1$ and $C_2$ with $K$ topics, respectively, return a topic model that is mined from the documents in $C_1$ and $C_2$.

- **Two straightforward solutions**:
  1. Re-train an LDA model on the documents from both cells.
  2. Fix the model of one cell, and re-sample the topics for each individual word from the other cell. (Online LDA. L. AlSumait, et al, 2008)

- **Our solution (Fast set sampling)**:
  - **Motivation**: words assigned to the same topic are highly correlated.
  - Fix the model of one cell, and re-sample the topics for each set of words assigned to the same topic in the pre-trained model on the other cell.
# Comparison Among the Methods

## Pre-trained

<table>
<thead>
<tr>
<th>Re-train Both</th>
<th>Online LDA</th>
<th>Set Sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cell 1</strong></td>
<td><strong>Cell 1</strong></td>
<td><strong>Cell 1</strong></td>
</tr>
<tr>
<td><strong>Cell 2</strong></td>
<td><strong>Cell 2</strong></td>
<td><strong>Cell 2</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Initial</strong></th>
<th><strong>Combined</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell 1</td>
<td>Cell 2</td>
</tr>
<tr>
<td><strong>w1, w2, w3, w5, w7, w2</strong></td>
<td><strong>w1, w2, w3, w4, w6</strong></td>
</tr>
<tr>
<td><strong>w1, w2, w4, w6</strong></td>
<td><strong>w1, w2, w3, w4, w6</strong></td>
</tr>
</tbody>
</table>

| **w1, w2, w3, w4, w7, w2** | **w1, w2, w3, w4, w7, w2** |
| **w1, w2, w3, w4, w7, w2** | **w1, w2, w3, w4, w7, w2** |
| **w1, w2, w3, w4, w7, w2** | **w1, w2, w3, w4, w7, w2** |

79
Comparison Among the Methods

• **Re-training Both (LDA):**
  - It samples topics for individual words in both cells, i.e., $O((|W_1| + |W_2|)KI)$.

• **Online LDA:**
  - It only samples topics for $W_2$, i.e., $O(|W_2|KI)$.

• **Our solution:**
  - It only samples topics for each set of words in $W_2$, i.e., $O((K + |D_2| + |V_2|)KI)$.
  - Bounded error to Online-LDA (Expected Euclidean distance): $\sqrt{2\frac{|W_1 \cap W_2| + |W_2| - 1}{|W_1| + |W_2|}}$

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_i$</td>
</tr>
<tr>
<td>$D_i$</td>
</tr>
<tr>
<td>$V_i$</td>
</tr>
<tr>
<td>$K$</td>
</tr>
<tr>
<td>$I$</td>
</tr>
</tbody>
</table>
Problem 2: Offline Pre-computation

- **Indexing structure**: Octree
- **Cell partitioning algorithm**
  - Typically, an Octree cell is *equally* divided.
  - A better partition → consider the online combining errors:
    - **Observation**: A query may cover the sub-cells $C_1, C_2$ within cell $C$. Combining $C_1$ and $C_2$ has error $f(|W_{C_1} \cap W_{C_2}|)$, where $f$ is an increasing function.
    - **Idea for partition**: Divide $C$ by minimizing $|W_{C_i} \cap W_{C_j}|$ among the sub-cells → better accuracy for online combining.

- Choose a set of cells to pre-compute topic models, given the memory budget and error tolerance.
Next Steps

• More sophisticated ranking!
  ■ Which signals to use?
    ◆ Webpage: quality of a web page, click through, diversity, etc.
    ◆ POI: Popularity and rating, etc
  ■ What is the relevant context for this?
    ◆ Dependence on location
    ◆ Dependence on keywords
    ◆ Dependence on search history
    ◆ Dependence on social network
    ◆ Dependence on time
  ■ How to combine them into a function (e.g., as a sum)?
  ■ Which weight parameters to use (e.g., a weight for each term)?
Next Steps

• **Personalized** queries (Location recommendation + Queries)

• Queryless Search: Which functionality to serve when?
  - Ex: mineral water, dumplings
  - How can context be used for determining user intent?

• Evaluation?
  - Which functionality is best where and when and for who?
  - GeoCLEF
Next steps

• Querying and mining geo-textual data streams
  ■ Storing, indexing data streams: snapshot queries, standing queries, analytics queries
    ◆ relevant trending events
    ◆ Casual relationship between events
  ■ Distributed systems for supporting querying and mining geo-textual data streams
  ■ By bridging the static geo-textual data and streaming geo-textual data, exciting opportunities for data analytics emerge

• Region search and exploration
  ■ What are interesting exploratory search and mining tasks on geo-textual data?
  ■ How to perform them efficiently?
Summary

• Querying geo-textual streams
  - Boolean publish/subscribe query
  - Top-k publish/subscribe query
  - Moving publish/subscribe query
  - Other query
• Exploring geo-textual data