ICDE勉強会 [session 5]
Keyword Search 1: Spatial

佐々木　勇和
王 沛

NAGOYA UNIVERSITY

[Session 5] keyword search 1: Spatial

[DE140243.pdf]

- (佐々木) Scalable Top-k Spatio-Temporal Term Querying Anders Skovsgaard, Darius Sidlauskas, Christian S. Jensen

[DE140366.pdf]

- (佐々木) Mercury: A Memory-Constrained Spatio-temporal Real-time Search on Microblogs Amr Magdy, Mohamed F. Mokbel, Sameh Elnikety, Suman Nath, Yuxiong He

[DE140281.pdf]

- (王) Nearest Keyword Set Search in Multi-dimensional Datasets Vishwakarma Singh, Ambuj K. Singh

Scalable Top-k Spatio-Temporal Term Querying

Anders Skovsgaard, Darius Sidlauskas, Christian S. Jensen

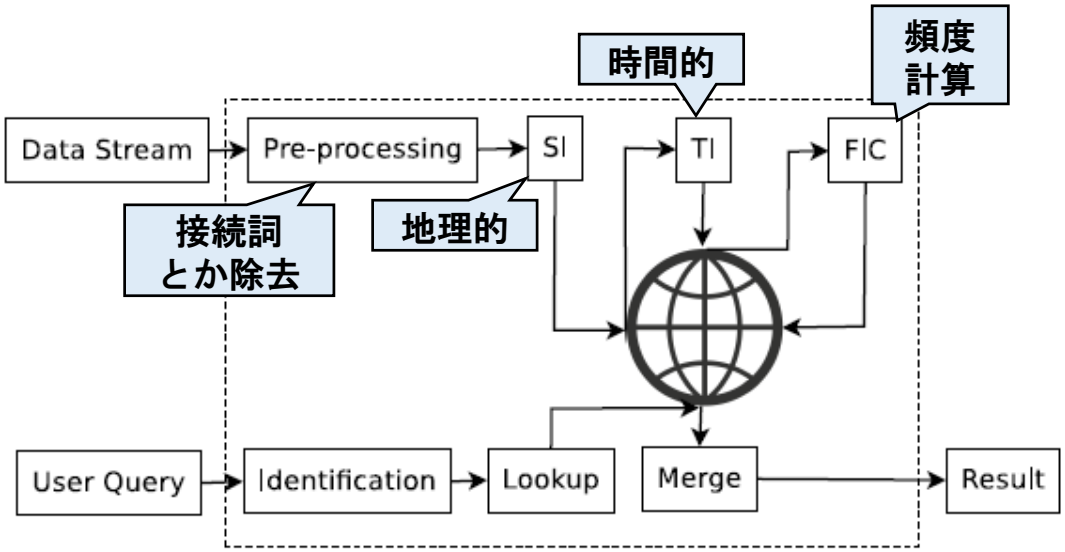
- 動機：**時間的**，**地理的**に出現頻度が多い**単語**を知りたい
  - Ex. 2012年10月2週目ニューヨーク，ツイートで多かった単語は？
    - NYTMetro, Sandy, Evacuation （ハリケーンSandyが直撃）
- 課題：
  - 全ての単語の出現数を数えるのは，メモリの使用量が大きい.
  - 数える単語数を固定にすると，多様性のあるデータに対応できない.

メモリ使用量を増やさず，精度を落とさず，頻度計算

- 提案：**AFIA**システム (Adaptive Frequent Item Aggregator)
  - 地理および時間的に区切って，単語の出現頻度を数える.
  - 数える単語数を動的に変化させる.
  - クエリ処理

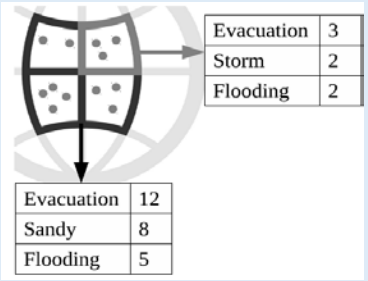
クエリとシステム構成

- **Top-k spatio-temporal term**クエリ
  - 検索する単語数 k, 時間範囲 $I = [t_s, t_e]$，地理的範囲 R を指定
  - 時間$[t_s, t_e]$内かつ地理的範囲R内のデータの中から，出現頻度が多いk単語を取得
- **AFIA**の構成
  - Multi-layerグリッドのインデックス
    - 複数の粒度のグリッドでデータを保持
    - 地理的なグリッドと時間的なグリッドを使用



Multi-layerグリッドのインデックス
- 空間：0.1km², 1km², 10km², 100km²
- 時間：1時間，1日，1週間

空間的に区切る



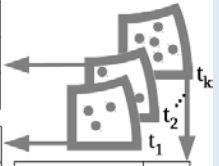
Evacuation	3
Storm	2
Flooding	2

Evacuation	12
Sandy	8
Flooding	5

時間的に区切る

Storm	2
Sandy	2
Evacuation	1

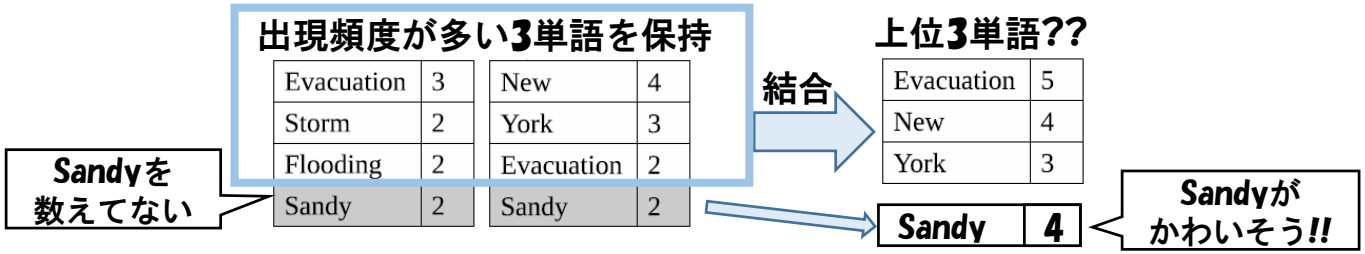
Evacuation	3
Storm	2
Flooding	2



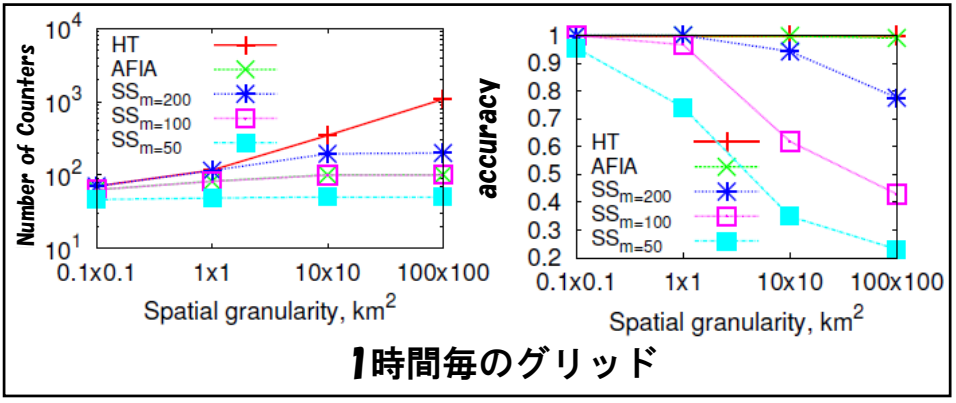
New	4
York	3
Evacuation	2

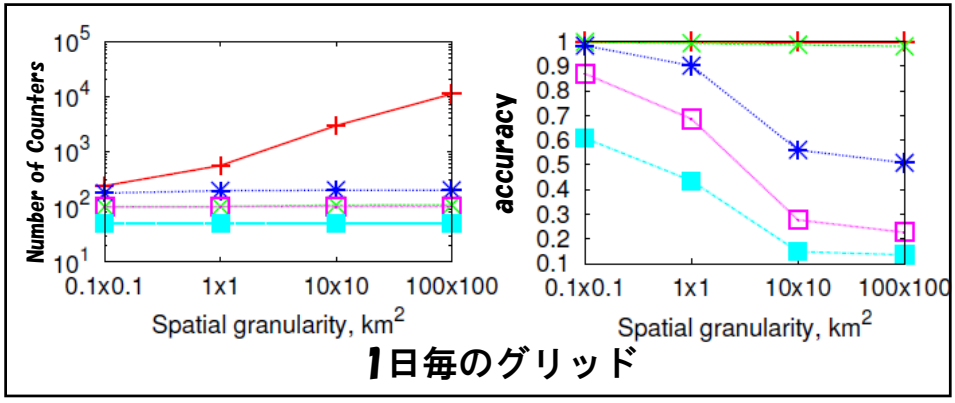
提案のポイントと実験の要約

- 提案のポイント
  - 不整合を出来るだけ減らすように，単語数を調整する．
    - 上位に入らない単語の出現数を数えて，エラーを防ぐ．
  - クエリ処理：上位を保障できる単語とできない単語を区別して出力



- 実験の要約
  - データセット：2013年5月の110,425,053ジオタグ付ツイート
  - 結果：固定数100単語のメモリ使用量で，ほぼ100%の精度を維持
    - [凡例] HT: ハッシュを用いて全単語の頻度保持，$SS_{m=x}$: x個の単語の頻度保持





Mercury: A Memory-Constrained Spatio-temporal Real-time Search on Microblogs

Amr Magdy, Mohamed F. Mokbel, Sameh Elnikety, Suman Nath, Yuxiong He

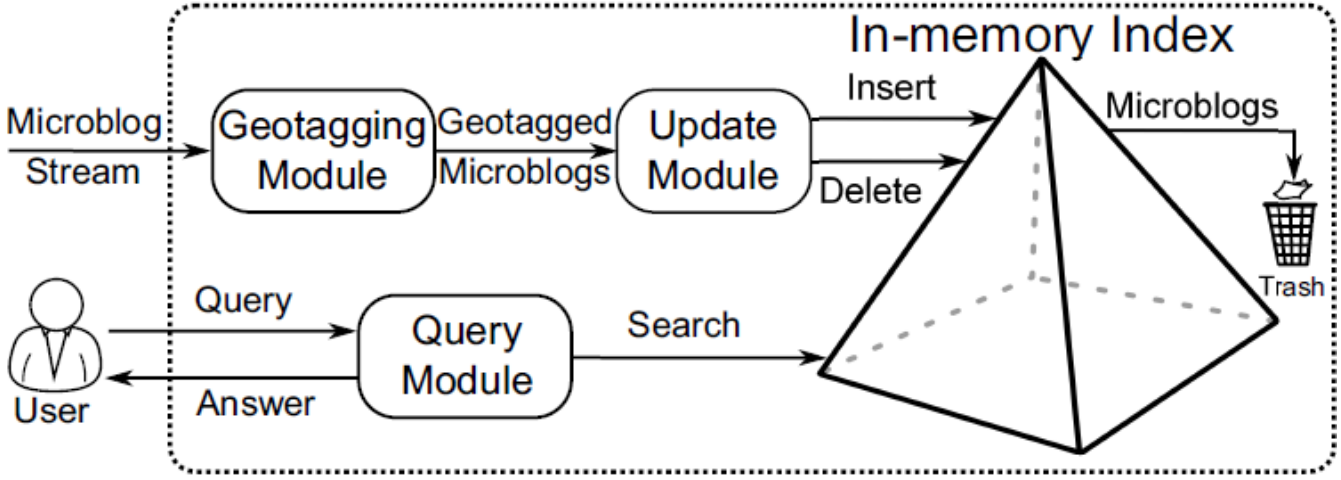
- 動機：任意の場所のリアルタイムなマイクロブログを取得したい.
  - Ex. ボストンマラソンの爆破事件のツイート
    - キーワードなしで，今起こってることが知りたい.
- 課題：
  - 高頻度なブログ生成
  - リアルタイムなクエリ処理

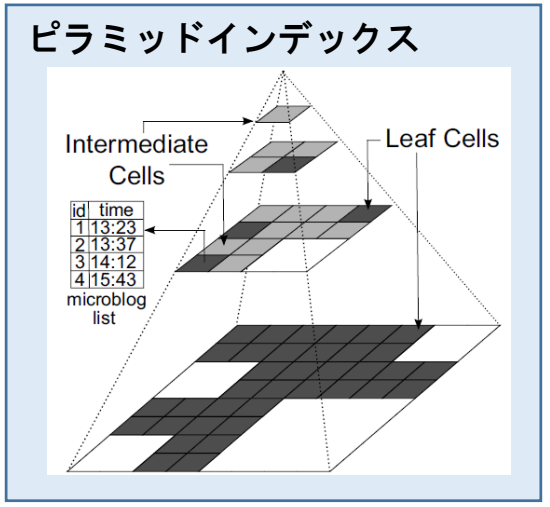
メモリ内での効率的な処理が必要

- 提案：Mercuryシステム
  - メモリ内のインデックス：ピラミッドインデックスを拡張
  - クエリ処理：地理的，時間的に検索領域を枝がり
  - インデックスサイズ調整：不必要そうなデータを削除
  - 負荷削減：精度を少し犠牲にして，メモリ使用量を削減

クエリとシステム構成

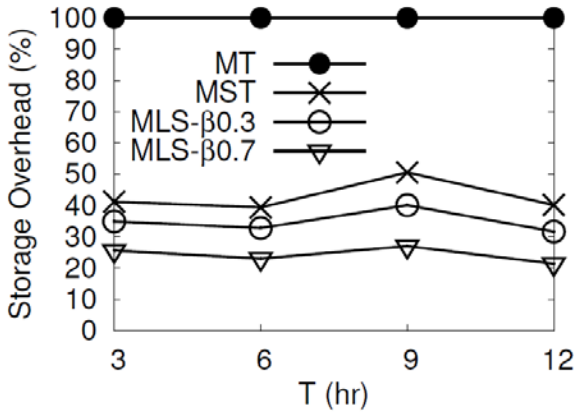
- **Spatio-temporal**クエリ: 指定の位置から近くて最近のブログをk個取得
  - 検索するブログ数 k, 時間 T ， 範囲 R ， ランキング関数F_αを指定
    - T以上時間経過してるブログはいらない
    - ユーザ位置からR以上関連位置が離れているブログはいらない
    - F_αにより地理と時間のどちらが重要かを指定
- **Mercury**の構成
  - ピラミッドインデックス
    - 位置依存データを保持に適したインデックス
    - セル内のデータが多くなるとセルを4分割し，対応する領域の子セルにデータを格納

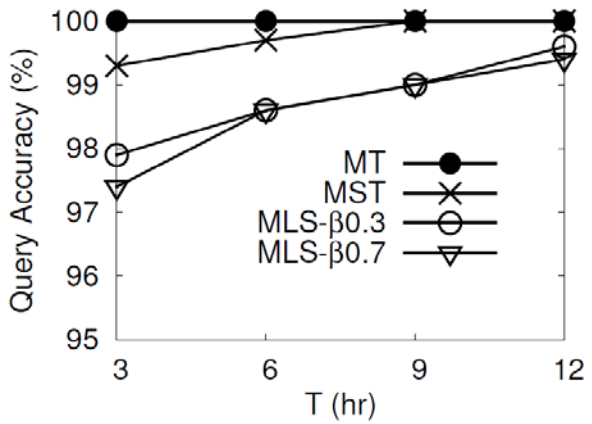


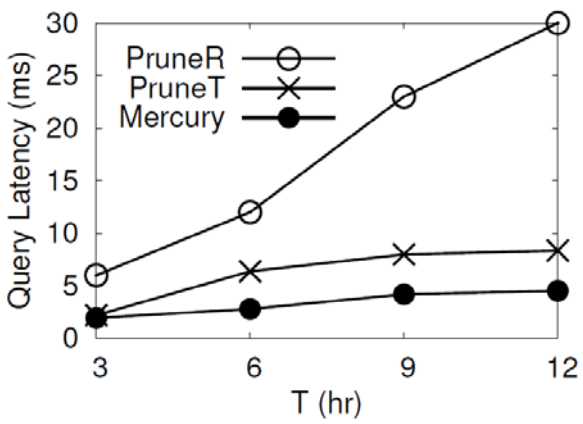


提案のポイントと実験の要約

- 提案のポイント
  - ピラミッドインデックスを拡張
    - 各セルにデータを時間的に降順に格納
    - 高頻度のデータ生成に対応するための一括挿入/削除とセル分割/統合方法を提案
  - インデックスサイズ調整: データ生成頻度を基に，削除するデータを決定
  - 負荷削減: データ生成頻度と地理的な要素を考慮して，削除するデータを決定
- 実験の要約
  - データセット：3.4億以上ツイートと100万回のBing検索
  - 結果：メモリ使用量を削減しつつ，高精度を達成
    - [凡例] MT: 一括削除, MST: インデックスサイズ調整, MLS: 負荷削減（βはパラメータ）
    - [凡例] PruneR: 地理的に枝がり，PruneT: 時間的に枝がり，Mercury: 両方で枝がり







Session 5：Keyword Search I – Spatial

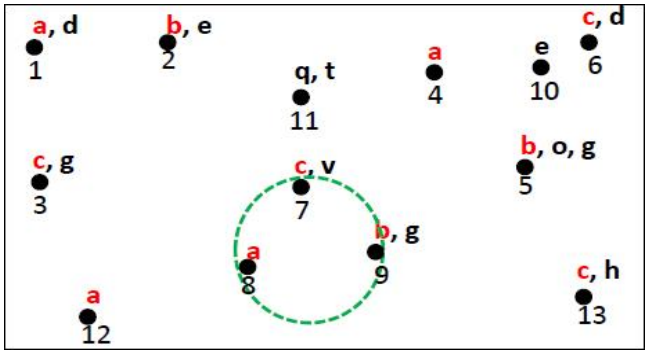
担当：王　(名古屋大学)

Some figures are copied from ICDE 2014 proceedings

Nearest Keyword Set Search in Multi-dimensional Datasets

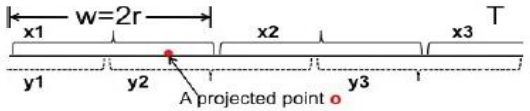
- Vishwakarma Singh, Ambuj K. Singh(U. California, Santa Babara)
- 問題設定（NKS）
  - Rはd次元のN個の点データ集合
  - 各点は各自のPointID(ユニーク)を持つ
  - 各点Dはいくつかの Keywordsを持っている
  - 点の集合の距離の定義：最大直径
  - クエリQはKeywordsの集合
  - QのKeywordsを全部含んでいる最近傍の点の集合を発見

query={a,b,c}

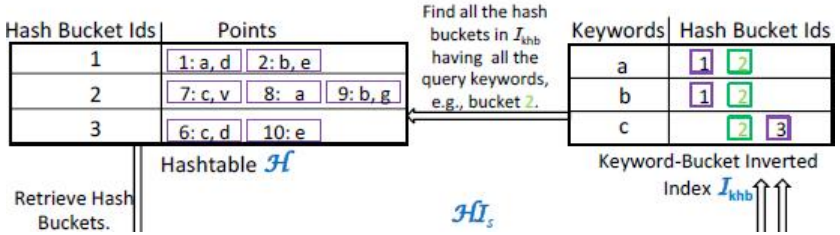


新たなIndex構造手法

- Index: Mutiple hashtables + 対応inverted list
- Step1: 全部の点をあるベクトルに投影

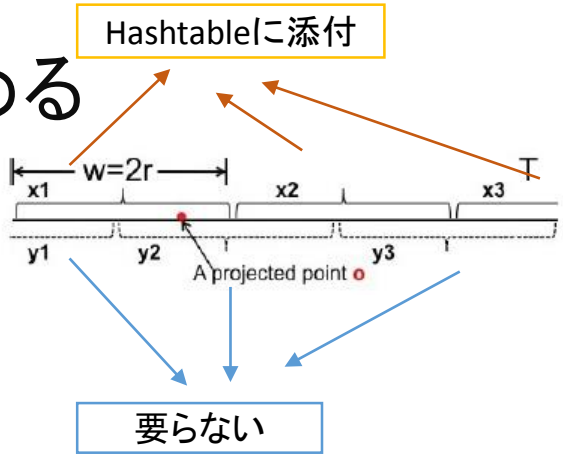


- Step2: 投影された線を広さwのオーバラップ区間に分ける
- Step3：x1、x2、x3、y1、y2、y3はhashID, Hashtableをつくる
- Step4: 各hashtableのkeywordを合併、hashIDとkeywordの関係のinverted listを作る



アプローチ

- 定理：全部の点を含んでいる区間が必ず存在
- 正確な解：
  - Step1: パラメータrを設定、hashtableを作る
  - Step2: Inverted Listを使って、候補解を生成
  - Step3：解を選択、top-kの解を取る
- 近似解：hashtableの作る条件を緩める
- オーバラップを削除



実験と今後の研究

- Dataset: 人工データと実データ
- bR＊Treeと対比
- 評価基準：クエリ時間、空間の使用量、近似率
- 今後：
  - 結果のスコアリング法の改善
  - Keywordに重さを付ける
  - 結果がクエリの全部のkeywordを含む
  - ↓
  - 部分のkeywordを含む