LMU
LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

Institut für Informatik

GeoScope:

Online Detection of Geo-Correlated Information Trends in Social Networks

Data Science - Hauptseminar SS 2016

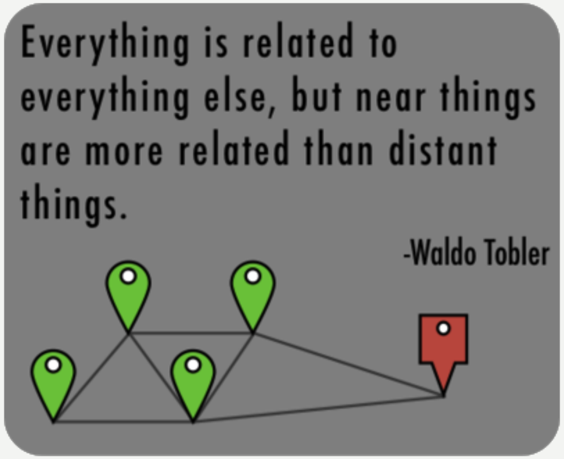
Michael Prummer, Jennifer Ling, Sandra Zollner

Agenda

1. GeoScope Introduction and Basics
2. GeoScope System
3. GeoScope Experiments with Twitter Data
4. Project Ideas

Geo-Trends

First Law of Geography
"Everything is related to everything else, but near things are more related than distant things."



Social networks:

- Choice of friends
- Topic interest
- Tendency to talk about events that are close-by
- Use of language and sentiment

Geo-trends:

→ Local emergencies, political demonstrations, cultural events, etc.

GeoScope

Algorithmic tool to detect geo-trends

- Correlations between topic and location pairs in a sliding window
- Sublinear space and running time
- Guarantees detecting all trending correlated pairs

Problems:

- Large amount of noisy data shared on social networks (500M Tweets / day)
- Topic and location definition
- Filter global events

Detecting Geo Trends

Premise 1: “The frequency of any topic t_x and any location l_i in the current time window should be reported in an accurate and timely fashion.”

→ Pairs must be efficiently retrievable at any particular time.

Data stream: $\{(l_1, t_1), (l_6, t_1), (l_2, t_1), (l_3, t_5), (l_3, t_2)\}$

l_x = Location

t_x = Topic

Detecting Geo Trends

Premise 2: "(...) A location-topic pair (l_i, t_x) is significantly correlated if at least Φ fraction of all mentions from location l_i are about topic t_x and at least ψ fraction of all mentions about topic t_x are from location l_i."

→ Location and topic must be heavy-hitters for each other.

Φ := Dominance of topic t_x in location l_i

ψ:= Support of location l_i for topic t_x

Detecting Geo Trends

Premise 3: "Geo-trend detection should identify a list of "all" and "only" the locations that are at least Θ-frequent in the current time window and limit the reported correlations to such locations."

→ Filter out insignificant information: $F(l_i) > \Theta * N$

→ Eliminating unpopular locations

Θ:= Location-Topic significance

Example

List<location, topic> = $\{(l_1, t_1), (l_2, t_1), (l_3, t_1), (l_1, t_2), (l_1, t_3), (l_2, t_3), (l_2, t_3)\}$

User defined: $\Phi = \psi = 0.5$ [0, 1]

(l_2, t_3): $\Phi = 0.67$, $\psi = 0.67$

(l_1, t_2): $\Phi = 0.33$, $\psi = 1$

Φ := Dominance of topic in location

Ψ := Support of location for topic

Θ := Location-Topic significance

Problem definition

- Given:
  - Data stream S of location-topic pairs: (l_i, t_x)
  - 3 user defined thresholds θ, ϕ, ψ in interval [0,1]
- Goal: In sliding window (time/number limit) keep track of:
  - Frequencies $F(l_i)/F(t_x)$
  - All Pairs with $F(l_i) > \lceil \theta N \rceil$, $F(l_i, t_x) > \lceil \phi F(l_i) \rceil$ and $\mathrm{F}(l_i, t_x) > \lceil \psi F(t_x) \rceil$

  → Premises satisfied!

- Detect geo-trends by keeping track of all location-topic pairs and their frequencies within the current time window.

  → Exact Solution is infeasible → approximation method

Data Structure

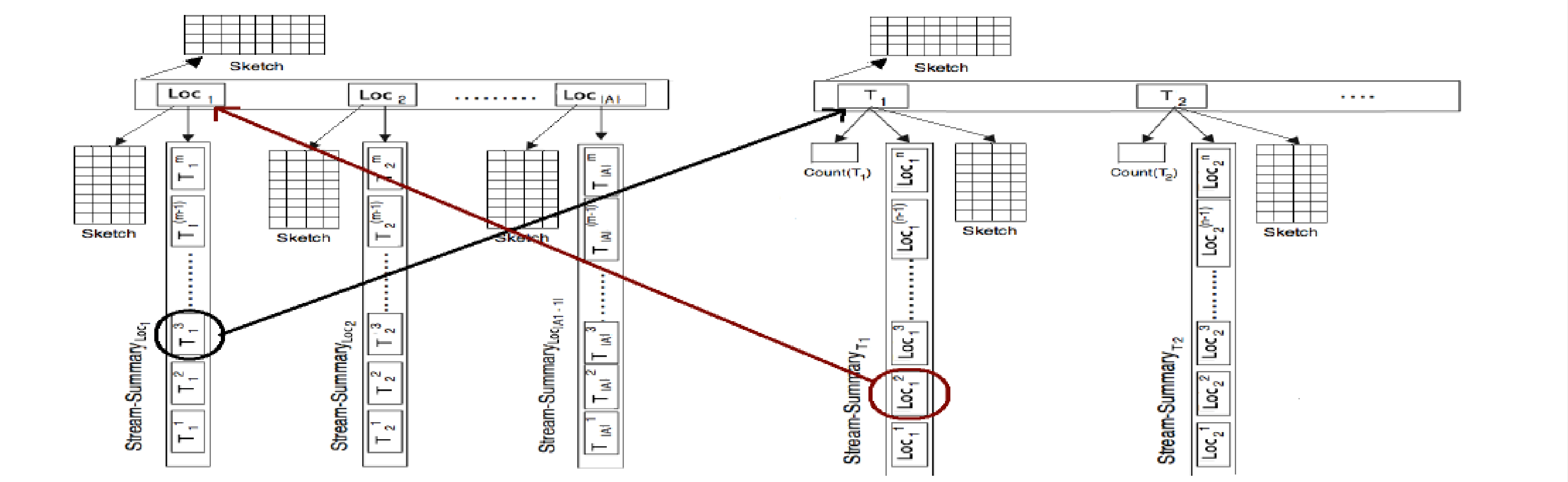


Figure 1: Overview of *GeoScope* Data Structures: *Location-StreamSummary-Table* (on the left) keeps track of ϕ-frequent topics for θ-frequent locations. *Topic-StreamSummary-Table* (on the right) keeps track of ψ-frequent locations for each topic that is ϕ-frequent for at least one location. Here the third most important topic for Loc_1 is T_1 and the second most important location for T_1 is Loc_1

Operations: insert, remove, report

Algorithm 1 Insert (l_i, t_x, ts)

$F(l_i) \leftarrow F(l_i) + 1$
if l_i turned θ-frequent **then**
  Create $StreamSummary_{l_i}$ with timestamp ts for location l_i
if l_i is θ-frequent **then**
  $F_{l_i}(t_x) \leftarrow F_{l_i}(t_x) + 1$
  if t_x turned ϕ-frequent for l_i **then**
    $StreamSummary_{l_i} = StreamSummary_{l_i} \cup \{t_x\}$
    Increase $Count_{t_x}$
for all l_j turned θ-infrequent **do**
  for all $t_y \in StreamSummary_{l_j}$ **do**
    Decrease $Count_{t_y}$
  Delete $StreamSummary_{l_j}$
for all t_y turned ϕ-infrequent for location l_i **do**
  $StreamSummary_{l_i} = StreamSummary_{l_i} \setminus \{t_y\}$
  Decrease $Count_{t_y}$
$F(t_x) \leftarrow F(t_x) + 1$
if $t_x \in$ *Topic-StreamSummary-Table* **then**
  $F_{t_x}(l_i) \leftarrow F_{t_x}(l_i) + 1$
  if l_i turned ψ-frequent for t_x **then**
    $StreamSummary_{t_x} = StreamSummary_{t_x} \cup \{l_i\}$
  for all l_j turned ψ-infrequent for t_x **do**
    $StreamSummary_{t_x} = StreamSummary_{t_x} \setminus \{l_j\}$

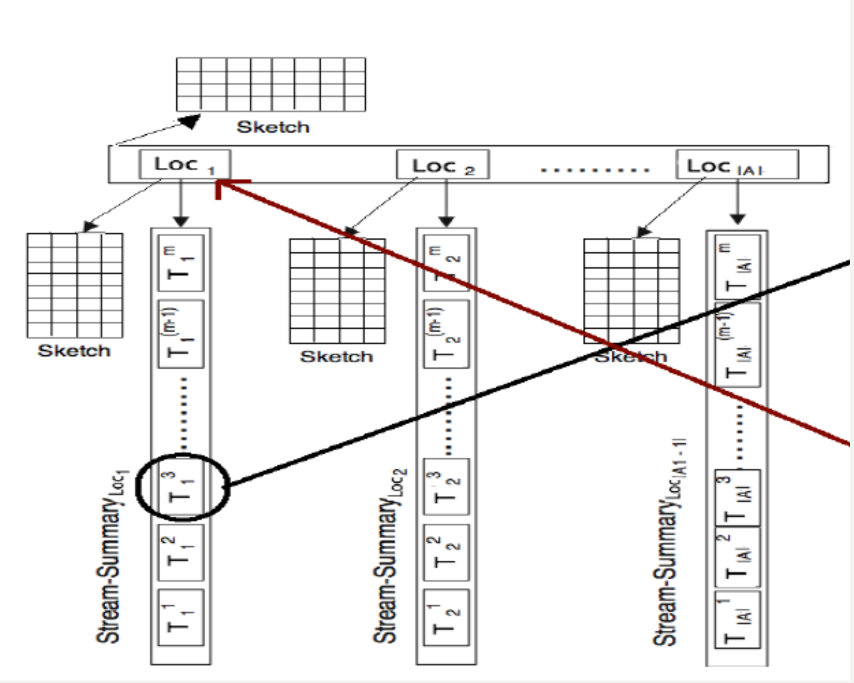
Algorithm 2 Remove (l_i, t_x, ts)

$F(l_i) \leftarrow F(l_i) - 1$
if l_i is θ-frequent **then**
  if $TS(StreamSummary_{l_i}) \leq ts$ **then**
    $F_{l_i}(t_x) \leftarrow F_{l_i}(t_x) - 1$
    if t_x turned ϕ-infrequent for l_i **then**
      $StreamSummary_{l_i} = StreamSummary_{l_i} \setminus \{t_x\}$
      Decrease $Count_{t_x}$
  if l_i turned θ-infrequent **then**
    for all $t_y \in StreamSummary_{l_i}$ **do**
      Decrease $Count_{t_y}$
    Delete $StreamSummary_{l_i}$
$F(t_x) \leftarrow F(t_x) - 1$
if $t_x \in$ *Topic-StreamSummary-Table* **then**
  if $TS(StreamSummary_{t_x}) \leq ts$ **then**
    $F_{t_x}(l_i) \leftarrow F_{t_x}(l_i) - 1$
    if l_i turned ψ-infrequent for t_x **then**
      $StreamSummary_{t_x} = StreamSummary_{t_x} \setminus l_i$

Operation: insert

Algorithm 1 Insert (l_i, t_x, ts)

$F(l_i) \leftarrow F(l_i) + 1$
if l_i turned θ-frequent **then**
    Create $StreamSummary_{l_i}$ with timestamp ts for location l_i
if l_i is θ-frequent **then**
    $F_{l_i}(t_x) \leftarrow F_{l_i}(t_x) + 1$
    if t_x turned ϕ-frequent for l_i **then**
        $StreamSummary_{l_i} = StreamSummary_{l_i} \cup \{t_x\}$
        Increase $Count_{t_x}$
for all l_j turned θ-infrequent **do**
    for all $t_y \in StreamSummary_{l_j}$ **do**
        Decrease $Count_{t_y}$
    Delete $StreamSummary_{l_j}$
for all t_y turned ϕ-infrequent for location l_i **do**
    $StreamSummary_{l_i} = StreamSummary_{l_i} \setminus \{t_y\}$
    Decrease $Count_{t_y}$

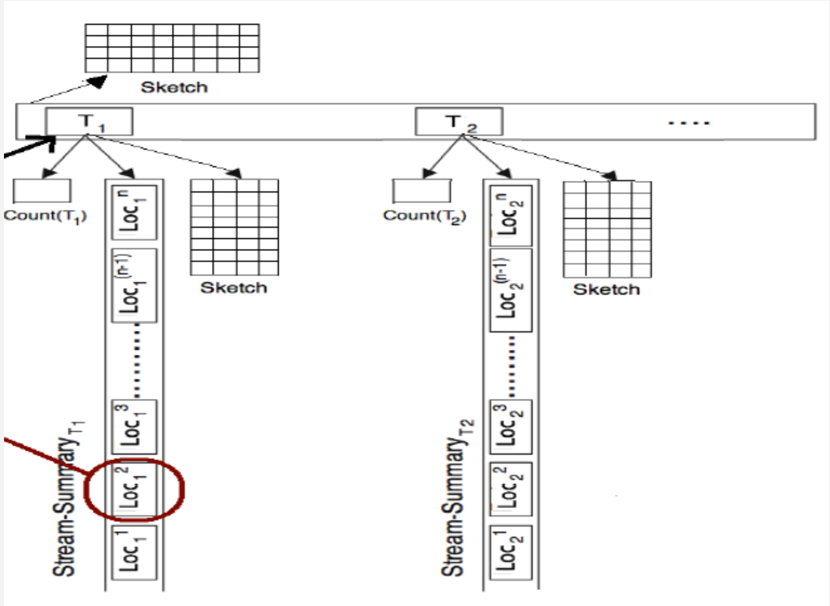


- $F(l_i) > [\theta N]$, $F(l_i, t_x) > [\phi F(l_i)]$ and $F(l_i, t_x) > [\psi F(t_x)]$

Operation: insert

Algorithm 1 Insert (l_i, t_x, ts)

```
F(t_x) ← F(t_x) + 1
if t_x ∈ Topic-StreamSummary-Table then
    F_{t_x}(l_i) ← F_{t_x}(l_i) + 1
    if l_i turned ψ-frequent for t_x then
        StreamSummary_{t_x} = StreamSummary_{t_x} ∪ {l_i}
    for all l_j turned ψ-infrequent for t_x do
        StreamSummary_{t_x} = StreamSummary_{t_x} \ {l_j}
```



- $F(l_i) > \lceil \theta N \rceil$, $F(l_i, t_x) > \lceil \phi F(l_i) \rceil$ and $F(l_i, t_x) > \lceil \psi F(t_x) \rceil$

Operation: remove

Algorithm 2 Remove (l_i, t_x, ts)

$F(l_i) \leftarrow F(l_i) - 1$
if l_i is θ-frequent **then**
  if $TS(StreamSummary_{l_i}) \leq ts$ **then**
    $F_{l_i}(t_x) \leftarrow F_{l_i}(t_x) - 1$
    if t_x turned ϕ-infrequent for l_i **then**
      $StreamSummary_{l_i} = StreamSummary_{l_i} \setminus \{t_x\}$
      Decrease $Count_{t_x}$
  if l_i turned θ-infrequent **then**
    for all $t_y \in StreamSummary_{l_i}$ **do**
      Decrease $Count_{t_y}$
    Delete $StreamSummary_{l_i}$

Location-StreamSummary-Table

$F(t_x) \leftarrow F(t_x) - 1$
if $t_x \in$ *Topic-StreamSummary-Table* **then**
  if $TS(StreamSummary_{t_x}) \leq ts$ **then**
    $F_{t_x}(l_i) \leftarrow F_{t_x}(l_i) - 1$
    if l_i turned ψ-infrequent for t_x **then**
      $StreamSummary_{t_x} = StreamSummary_{t_x} \setminus l_i$

Topic-StreamSummary-Table

- $F(l_i) > [\theta N]$, $F(l_i, t_x) > [\phi F(l_i)]$ and $\mathrm{F}(l_i, t_x) > [\psi F(t_x)]$

Running time and memory requirements

- Sub-linear in its space usage
- Two update operations (insert & remove): log-linear running time

Accuracy Guarantees

- A location-topic pair (l_i; t_x) is a trending correlated pair if and only if t_x is a trending topic for l_i and l_i is a trending location for t_x.
- trending = non-decreasing relative frequency
- Perfect recall guaranteed!
- I.e. At any given time *ts*, all trending correlated pairs in the time window ending at *ts* are reported by GeoScope

Case Study: Twitter

Data set:

- hashtag → topic
- city (tweet originates from) → location
- February 1st to June 18th 2011
- 63 M Tweets

Two ways to get location:

- tweet location
- user location

Effectiveness

Comparison of the solution to three baselines:

- Traditional Heavy-Hitters Approach (THHA)
- Geographical Heavy-Hitters Approach (GHHA)
- Statistically Significant Topic-Location Detection (SSTLD)

Information overload

Method	GeoScope	GHHA	SSTLD
Number of pairs	17	23	150 000

Effectiveness

Human validation

- Online questionnaire
- Human judge has to choose hashtag with most geographical significance out of two distinct hashtags h1 and h2

	THHA vs. GeoScope	GHHA vs. GeoScope	SSTLD vs. GeoScope
Fraction of *GeoScope* hashtags	0.94	0.74	0.89

Topics and locations

Topics with high geo-significance

- many hashtags with global significance (e.g. #ff, #jobs)
- only a small amount of topics is signficant as geographical trend (e.g. #egypt)

Cities with high geo-significance

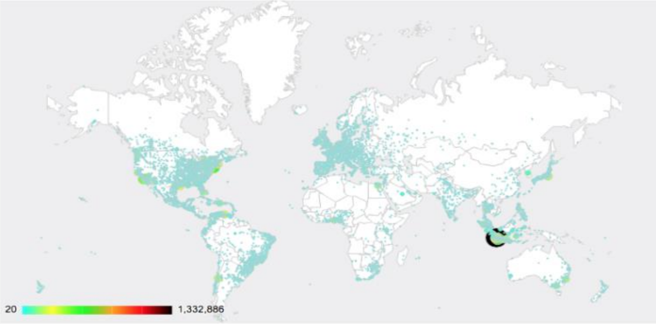
- cities which appear in a large number of correlations (e.g. Santiago)

Topics and locations

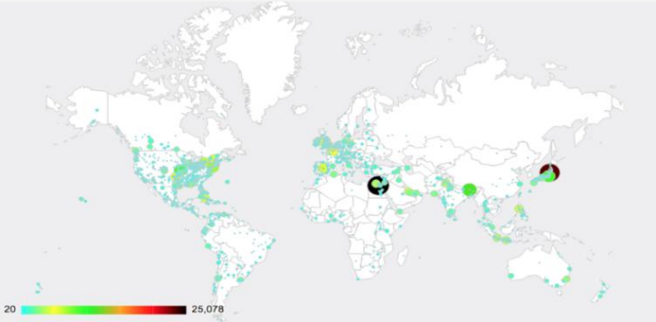
Geo-Origin vs. Geo-Focus

- geo-origin: where social content is created
- geo-focus: what location content is about

→ GeoScope is geo-focus based



Tweets *in* cities [1]



Tweets *about* cities [1]

Topics and locations

Sliding window

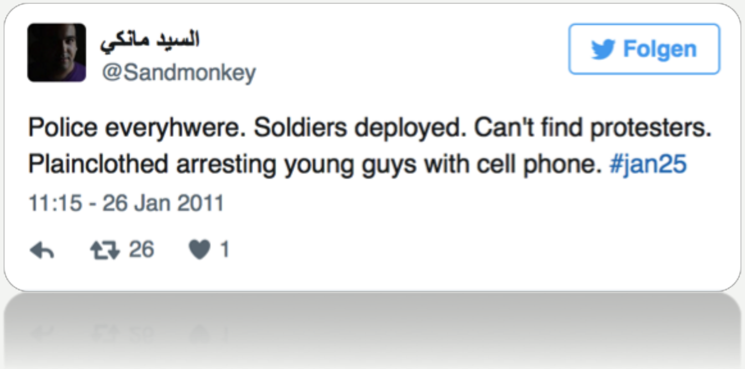
- interesting topics detected at particular points in time

Examples:

#earthquake correlated to Tokio

#Jan25, Cairo

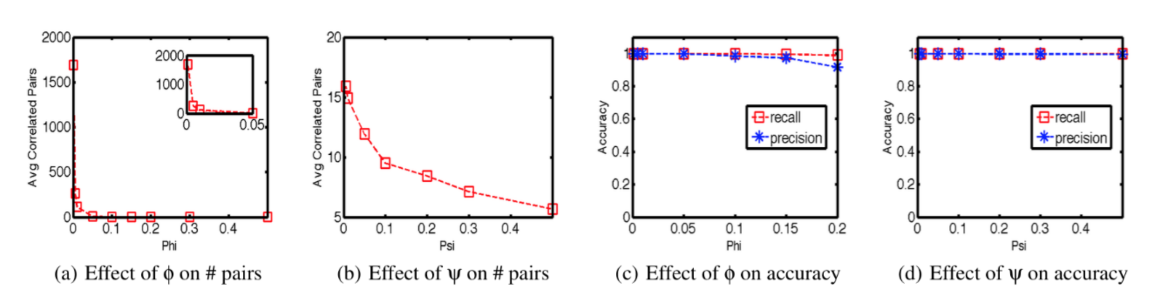
#NewCastle, Nottingham



→ crisis management, political interests, general interests

Accuracy

- Increasing Φ and ψ drastically decreases the number of correlated pairs
- proper settings are dependent on the specific application
- perfect recall rate for various settings
- increasing ψ slightly affects precision rate



Effect of Φ and ψ on accuracy [1]

Space and Time Efficiency

- comparison between GeoScope and exact method

Increasing window size:

- memory requirement remains constant (GeoScope)
- time required to report correlated pairs remains constant

Conclusion

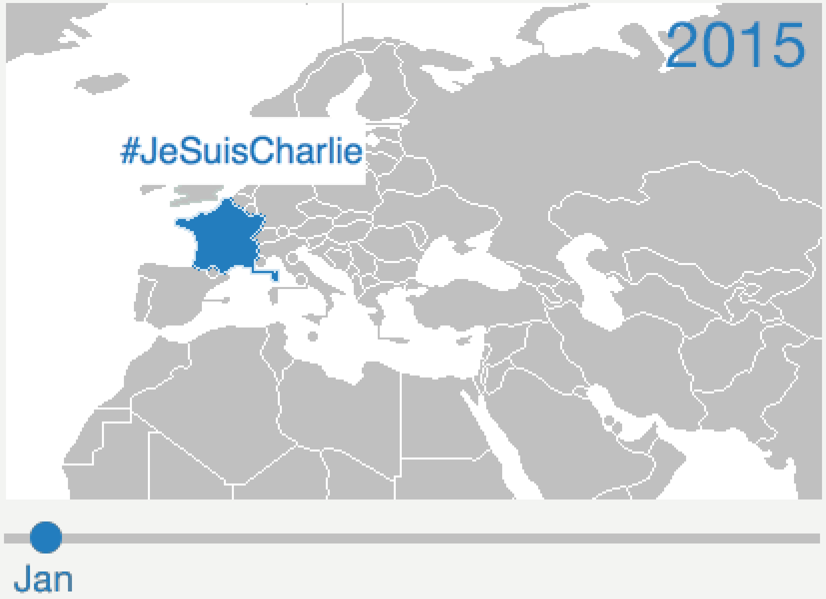
- Online detection of geo-correlated information trends
- GeoScope identifies correlated location-topic pairs along a sliding window in a social data stream
- Approximate solution (with sub-linear memory and running time) and guarantee to capture **all** trending correlations
- Tool is generic (not only Twitter analysis)
- Redefine topics (here: hashtag based) and locations (here: cities)
- Future work: compact way for hierarchical geo-trend detection

Project Ideas

I. Map of trending topics

Project Ideas

II. Map of trending topics (or specific products) over time period

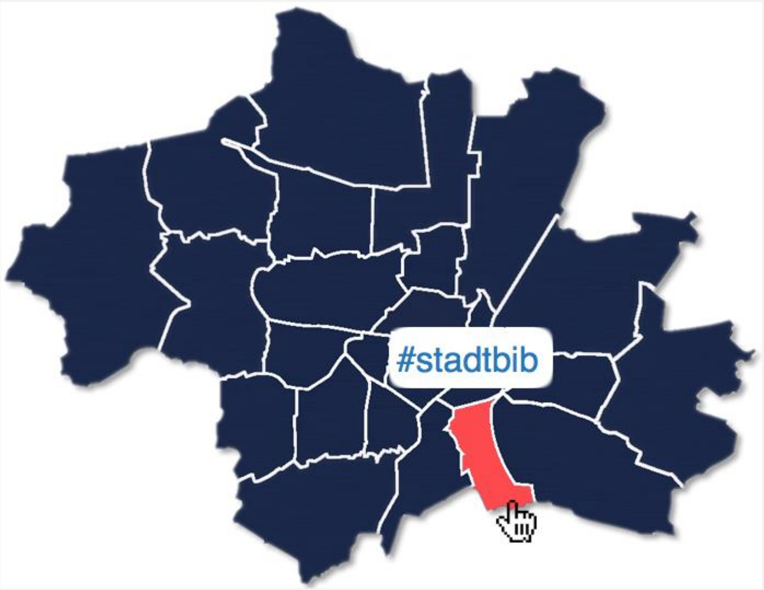


Project Ideas

III. 3D Version

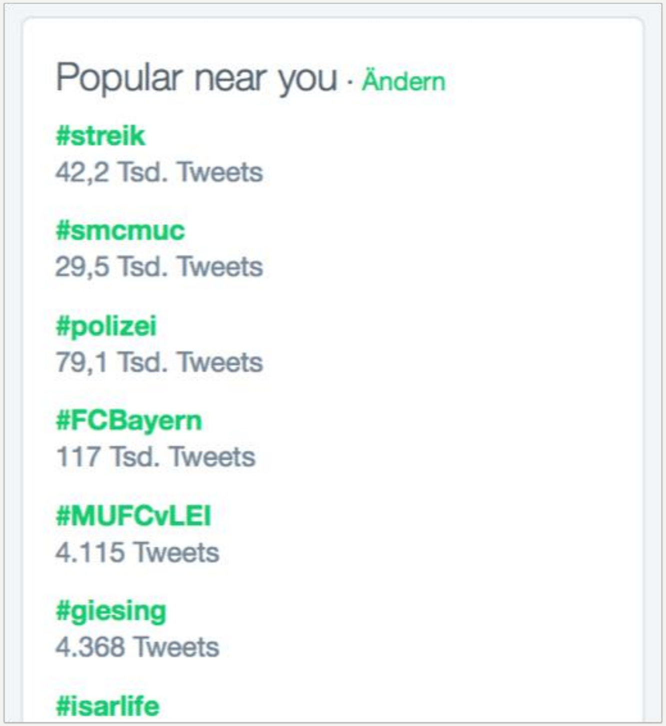
Project Ideas

IV. Map of trending topics for districts of a city



Project Ideas

V. Personal recommendations (depending on location)



Project Ideas

VI. Sarcasm in tweets (depending on location)

Sources

[1] Budak, Ceren, et al. "Geoscope: Online detection of geo-correlated information trends in social networks." Proceedings of the VLDB Endowment 7.4 (2013): 229-240.

[2] https://twitter.com/agustinespina/status/718134116755988480

[3] https://twitter.com/Sandmonkey/status/30207160700899329