You Are Where You Tweet: A Content-Based Approach to Geo-locating Twitter Users

CIKM'10

Zhiyuan Cheng, James Caverlee and Kyumin Lee
Texas A&M University

Presented by Yi Zhu

November 16, 2010

Outline

Examples

shirazluv Lilian 郭
Going out at 12.30 to meet my couzin in Mongkok. Kind of lazy men.
28 minutes ago

- Mongkok
- Hong Kong
- Object: Locating a Twitter user based on the content of tweets.

Motivation

- Location sparsity problem of Twitter
  - 26% users have listed a user location as granular as a city name.
  - Twitter begin to support per-tweet geo-tagging since August 2009. However, fewer than 0.42% tweets are tagged.

Motivation

- Personalized information services
  - Local news providing
  - Regional advertisements
  - Location-based application (earthquake detection)
- Avoid the need for sensitive data (private user information, IP address)

Challenges

- Tweets status updates are noisy. Mixing a variety of daily interests.
- Twitter users often rely on shorthand and non-standard vocabulary for informal communication.
- A user may span multiple locations beyond their immediate home location.
- A user may have more than one associated locations.

Problem Defined

- Given tweets of Twitter users, our goal is to estimate the city-level location of a user based purely on the content of their tweets.

Problem Defined

- Formally, the location estimation problem is defined as follows:
  - Given a set of tweets $S_{tweets}(u)$ posted by user u;
  - Estimate a user's probability of being located in city i: $p(i|S_{tweets}(u))$, such that the city with maximum probability $l_{est}(u)$ is the user's actual location $l_{act}(u)$.

Data Crawling

- API: twitter4j (open-source library for java).
- Two crawling strategies:
  - Crawling through Twitter's public timeline API. (Active Twitter Users)
  - Crawling by breadth-first search through social edges to crawl each user's friends. (Sub Social Graph of Twitter)

Dataset Description

- From Sep 2009 to Jan 2010
- Users: $1,074,375$
- Tweets: $29,479,600$
- 75.05% users list location, but overly general (California) or nonsensical (Wonderland).
- 21% users list a location as granular as a city name.
- 5% users list latitude/longitude coordinate.

Dataset Filter

- Filter all listed locations that have a valid city-level label.
- Users: $130,689$
- Tweets: $4,124,960$
- Test Set:
  - Extract users with 1000+ tweets and latitude/longitude coordinates. (Generated by smartphone)
  - Users: $5,190$
  - Tweets: more than 5 million

Evaluation Metrics

- Error Distance for user u
  - $ErrDist(u) = d(l_{act}(u), l_{est}(u))$
- Average Error Distance for all users U:
  - $AvgErrDist(U) = \frac{\sum_{u \in U} ErrDist(u)}{|U|}$
- Accuracy:
  - $Accuracy(U) = \frac{|\{u | u \in U \wedge ErrDist(u) \leqslant 100\}|}{|U|}$

Baseline Location Estimation

- $p(i|S_{words}(u)) = \sum_{w \in S_{words}(u)} p(i|w) \times p(w)$.
- $S_{words}(u)$ is the set of words extracted from user u.
- $p(w)$ is the probability of the word w in the whole dataset, $p(w) = \frac{count(w)}{t}$
- $p(i|w)$ the likelihood that each word w is issued by a user located in city i.

Baseline Location Estimation Result

- Accuracy: 10.12%
- AvgErrDist: 1773 miles
- Problem:
    - Local Words: isolate the words which can distinguish location of the user.
    - Tweet Sparsity: location sparsity of words in tweets.

Spatial variation model

- Given a word, decide if it is local or non-local.
- Spatial variation model (Backstrom et al., WWW'08)
  - Analysis of geographic distribution of terms in search engine query logs.
  - $Cd^{-\alpha}$ is the approximately probability of the query issued from a place with a distance d from the center.
  - C is a constant to specify the frequency of the center.
  - α control the speed of the frequency falls.

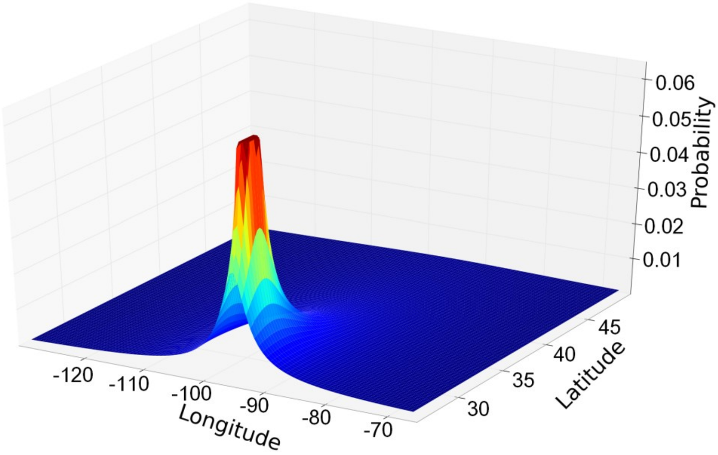
Identifying Local Words in Tweets

- C and α can be used to determine if the word is local.
- For a word w, given a center and the central frequency is C, compute the maximum-likelihood value.
- For each city i, users from i tweet word w n times:
  - $n > 0$, then multiply the overall probability by $(Cd_i^{-\alpha})^n$.
  - $n = 0$, then multiply the overall probability by $1 - Cd_i^{-\alpha}$.
  - d_i is the distance between city i and the center of word w.

Identifying Local Words in Tweets

- To avoid underflow, logarithms are added.
- Suppose S is the set of occurrences for word w, then:
- $f(C, \alpha) = \sum_{i \in S} \log Cd_i^{-\alpha} + \sum_{i \notin S} \log(1 - Cd_i^{-\alpha})$
- It has exactly one local maximum (unimodal)
    - Lattices
    - Golden section search

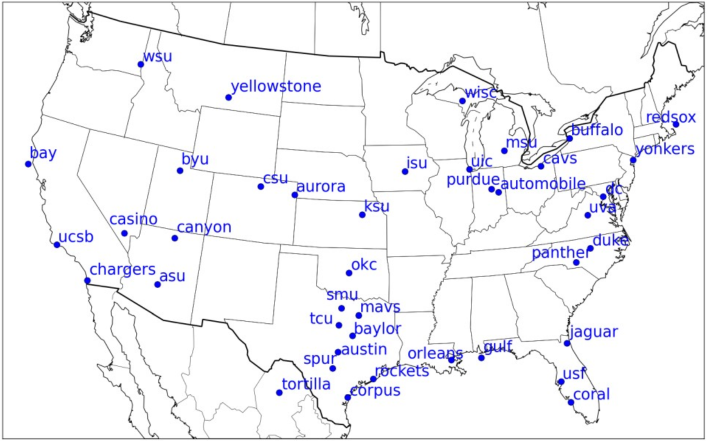
Identifying Local Words in Tweets



Identifying Local Words in Tweets

Word	Latitude	Longitude	C_0	α
automobile	40.2	-85.4	0.5018	1.8874
casino	36.2	-115.24	0.9999	1.5603
tortilla	27.9	-102.2	0.0115	1.0350
canyon	36.52	-111.32	0.2053	1.3696
redsox	42.28	-69.72	0.1387	1.4516

Identifying Local Words in Tweets



Laplace Smoothing (Add-One Smoothing)

- $p(i|w) = \frac{1+count(w,i)}{V+N(w)}$,
- $count(w, i)$: term count of word w in city i;
- V: the size of vocabulary;
- $N(w)$: total count of w in all cities.

State-Level Smoothing

- State probability:

$$p_s(s|w) = \frac{\sum_{i \in S_c} p(i|w)}{|S_c|},$$

  - S_c: set of cities in the state s.

- State-level smoothing:

$$p'(i|w) = \lambda \times p(i|w) + (1 - \lambda) \times p_s(s|w),$$

  - i: a city in the state s;
  - $1 - \lambda$: amount of smoothing.

Lattice-Based Neighborhood Smoothing

- Per-lattice probability:

$$p(lat|w) = \sum_{i \in S_c} p(i|w),$$

  - *lat*: a lattice.
  - S_c: set of cities in *lat*.

- Lattice probability:

$$p'(lat|w) = \mu * p(lat|w) + (1-\mu) * \sum_{lat_i \in S_{neighbors}} p(lat_i|w),$$

  - μ: parameter.
  - neighbors: 8 lattice around *lat*.

Lattice-Based Neighborhood Smoothing

- Lattice-based neighborhood smoothing:

$$p'(i|w) = \lambda * p(i|w) + (1 - \lambda) * p'(lat|w),$$

  - i: a city in the lattice lat;
  - λ: smoothing parameter.

Model-Based Smoothing

- $p'(i|w) = C(w)d_i^{-\alpha(w)}$,
  - $C(w), \alpha(w)$: optimized parameters for word w.

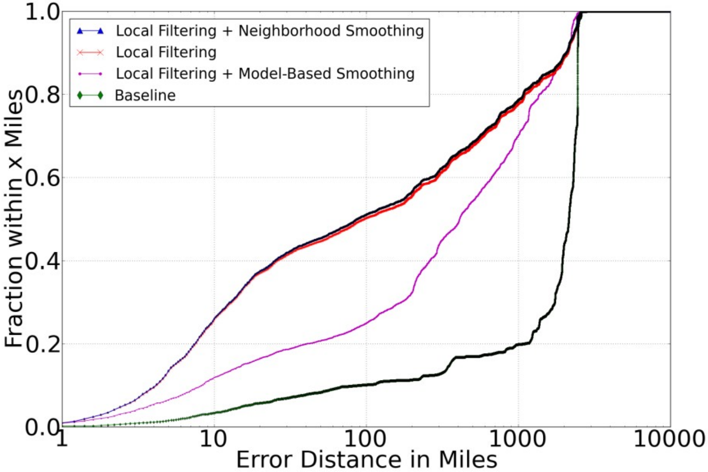
Smoothing Comparison

	Geographic Range	Parameters	Complexity
Laplace	None	None	Low
State-Level	State	λ	High
Neighborhood	Neighbor Lattices	μ, λ	Highest
Model-Based	Global	None	Lowest

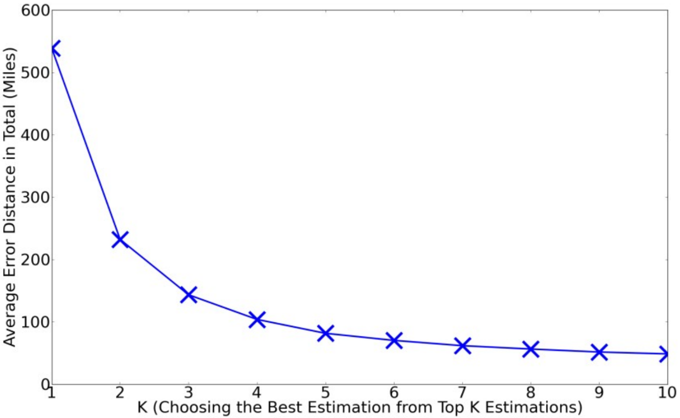
Model and Smoothing Comparison

Method	ACC	AvgErrDist (Miles)	ACC@2	ACC@3	ACC@5
Baseline	0.101	1773.146	0.375	0.425	0.476
+ Local Filtering (LF)	0.498	539.191	0.619	0.682	0.781
+ LF + Laplace	0.480	587.551	0.593	0.647	0.745
+ LF + State-Level	0.502	551.436	0.617	0.687	0.783
+ LF + Neighborhood	**0.510**	**535.564**	**0.624**	**0.694**	**0.788**
+ LF + Model-based	0.250	719.238	0.352	0.415	0.486

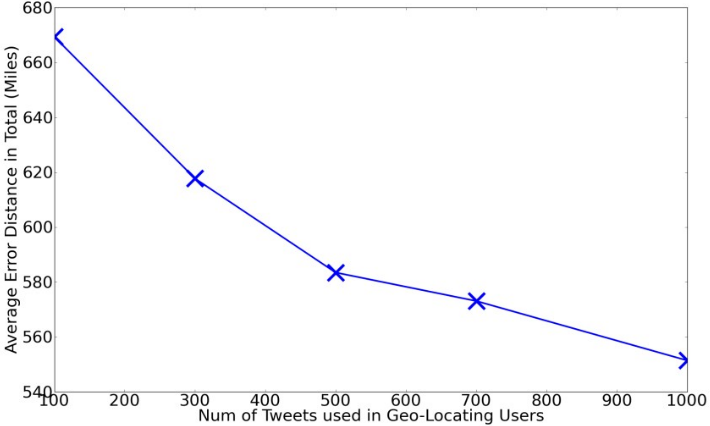
Model and Smoothing Comparison



Capacity of Estimator



Number of Tweets



Conclusion

- A probabilistic framework for estimating city-level location of Twitter users based on the content of tweets.
- Local words identifying and some smoothing can improve the estimation
- 100 tweets are enough for locating.

Thanks!

Q & A