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

Zhiyuan Cheng
Department of Computer Science and Engineering
Texas A&M University
College Station, TX, USA
zcheng@cse.tamu.edu

James Caverlee
Department of Computer Science and Engineering
Texas A&M University
College Station, TX, USA
caverlee@cse.tamu.edu

Kyumin Lee
Department of Computer Science and Engineering
Texas A&M University
College Station, TX, USA
kyumin@cse.tamu.edu

ABSTRACT

We propose and evaluate a probabilistic framework for estimating a Twitter user's city-level location based purely on the content of the user's tweets, even in the absence of any other geospatial cues. By augmenting the massive human-powered sensing capabilities of Twitter and related microblogging services with content-derived location information, this framework can overcome the sparsity of geo-enabled features in these services and enable new location-based personalized information services, the targeting of regional advertisements, and so on. Three of the key features of the proposed approach are: (i) its reliance purely on tweet content, meaning no need for user IP information, private login information, or external knowledge bases; (ii) a classification component for automatically identifying words in tweets with a strong local geo-scope; and (iii) a lattice-based neighborhood smoothing model for refining a user's location estimate. The system estimates k possible locations for each user in descending order of confidence. On average we find that the location estimates converge quickly (needing just 100s of tweets), placing 51% of Twitter users within 100 miles of their actual location.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database applications–Data mining; J.4 [**Computer Application**]: Social and Behavioral Sciences

General Terms: Algorithms, Experimentation

Keywords: Twitter, location-based estimation, spatial data mining, text mining

1. INTRODUCTION

The rise of microblogging services like Twitter has spawned great interest in these systems as human-powered sensing networks. Since its creation in 2006, Twitter has experienced an exponential explosion in its user base, reaching approximately 75 million users as of 2010 [4]. These users actively publish short messages ("tweets") of 140 characters or less to an audience of their subscribers ("followers"). With such a large geographically diverse user base, Twitter has essentially published terabytes of real-time "sensor" data in the form of these status updates.

Mining this people-centric sensor data promises new personalized information services, including local news summarized from tweets of nearby Twitter users [21], the targeting of regional advertisements, spreading business information to local customers [3], and novel location-based applications (e.g., Twitter-based earthquake detection, which can be faster than through traditional official channels [18]).

Unfortunately, Twitter users have been slow to adopt geospatial features: in a random sample of over 1 million Twitter users, only 26% have listed a user location as granular as a city name (e.g., Los Angeles, CA); the rest are overly general (e.g., California), missing altogether, or nonsensical (e.g., Wonderland). In addition, Twitter began supporting per-tweet geo-tagging in August 2009. Unlike user location (which is a single location associated with a user and listed in each Twitter user's profile), this per-tweet geo-tagging promises extremely fine-tuned Twitter user tracking by associating each tweet with a latitude and longitude. Our sample shows, however, that fewer than 0.42% of all tweets actually use this functionality. Together, the lack of user adoption of geo-based features per user or per tweet signals that the promise of Twitter as a location-based sensing system may have only limited reach and impact.

To overcome this location sparsity problem, we propose in this paper to predict a user's location based purely on the content of the user's tweets, even in the absence of any other geospatial cues. Our intuition is that a user's tweets may encode some location-specific content – either specific place names or certain words or phrases more likely to be associated with certain locations than others (e.g., "howdy" for people from Texas). In this way, we can fill-the-gap for the 74% of Twitter users lacking city-level granular location information. By augmenting the massive human-powered sensing capabilities of Twitter and related microblogging services with content-derived location information, this framework can overcome the sparsity of geo-enabled features in these services and bring augmented scope and breadth to emerging location-based personalized information services.

Effectively geo-locating a Twitter user based purely on the content of their tweets is a difficult task, however:

- First, Twitter status updates are inherently noisy, mix-

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.

ing a variety of daily interests (e.g., food, sports, daily chatting with friends). Are there clear location signals embedded in this mix of topics and interests that can be identified for locating a user?

- Second, Twitter users often rely on shorthand and non-standard vocabulary for informal communication, meaning that traditional gazetteer terms and proper place names (e.g., Eiffel Tower) may not be present in the content of the tweets at all, making the task of determining which terms are location-sensitive non-trivial.
- Third, even if we could isolate the location-sensitive attributes of a user's tweets, a user may have interests that span multiple locations beyond their immediate home location, meaning that the content of their tweets may be skewed toward words and phrase more consistent with outside locations. For example, New Yorkers may post about NBA games in Los Angeles or the earthquake in Haiti.
- Fourth, a user may have more than one associated location, e.g., due to travel, meaning that content-based location estimation may have difficulty in precisely identifying a user's location.

As a consequence, it is challenging to estimate the real location for a Twitter user based on an analysis of the user's tweets. With these issues in mind, in this paper, we propose and evaluate a probabilistic framework for estimating a Twitter user's city-level location based purely on the content of the user's tweets. The proposed approach relies on three key features: (i) its data input of pure tweet content, without any external data from users or web-based knowledge bases; (ii) a classifier which identifies words in tweets with a local geographic scope; and (iii) a lattice-based neighborhood smoothing model for refining the estimated results. The system provides k estimated cities for each user with a descending order of possibility. On average, 51% of randomly sampled Twitter users are placed within 100 miles of their actual location (based on an analysis of just 100s of tweets). We find that increasing amounts of data (in the form of wider coverage of Twitter users and their associated tweets) results in more precise location estimation, giving us confidence in the robustness and continued refinement of the approach.

The rest of this paper is organized as follows: Related work is in Section 2. Section 3 formalizes the problem of predicting a Twitter user's geo-location and briefly describes the sampled Twitter dataset used in the experiments. In Section 4, our estimation algorithm and corresponding refinements are introduced. We present the experimental results in Section 5. Finally, conclusions and future work are discussed in Section 6.

2. RELATED WORK

Studying the geographical scope of online content has attracted attention by researchers in the last decade, including studies of blogs [11, 15], webpages [7], search engine query logs [8], and even web users [13]. Prior work relevant to this paper can be categorized roughly into three groups based on the techniques used in geo-locating: content analysis with terms in a gazetteer, content analysis with probabilistic language models, and inference via social relations.

Several studies try to estimate the location of web content utilizing content analysis based on geo-related terms in a specialized external knowledge base (a gazetteer). Amitay et al. [7], Fink et al. [11], and Zong et al. [22] extracted addresses, postal code, and other information listed in a geographical gazetteer from web content to identify the associated geographical scope of web pages and blogs.

Serdyukov et al. [19] generate probabilistic language models based on the tags that photos are labeled with by Flickr users. Based on these models and Bayesian inference, they show how to estimate the location for a photo. In terms of the intention, their method is similar to our work. However, they use a GeoNames database to decide whether a user-submitted tag is a geo-related tag, which can overlook the spatial usefulness of words that may have a strong geo-scope (e.g., earthquake, casino, and so on). Separately, the work of Crandall et al. [10] proposes an approach combining textual and visual features to place images on a map. They have restrictions in their task that their system focuses on which of ten landmarks in a given city is the scope of an image.

In the area of privacy inference, a few researchers have been studying how a user's private information may be inferred through an analysis of the user's social relations. Backstrom et al. [9], Lindamood et al. [16], and Hearthely et al. [12] all share a similar assumption that users related in social networks usually share common attributes. These methods are orthogonal to our effort and could be used to augment the content-based approach taken in this paper by identifying common locations among a Twitter user's social network.

Recent work on detecting earthquakes with real-time Twitter data makes use of location information for tracking the flow of information across time and space [18]. Sakaki et al. consider each Twitter user as a sensor and apply Kalman filtering and particle filtering to estimate the center of the bursty earthquake. Their algorithm requires prior knowledge of where and when the earthquake is reported, emphasizing tracking instead of geo-locating users. As a result, this and related methods could benefit from our efforts to assign locations to users for whom we have no location information.

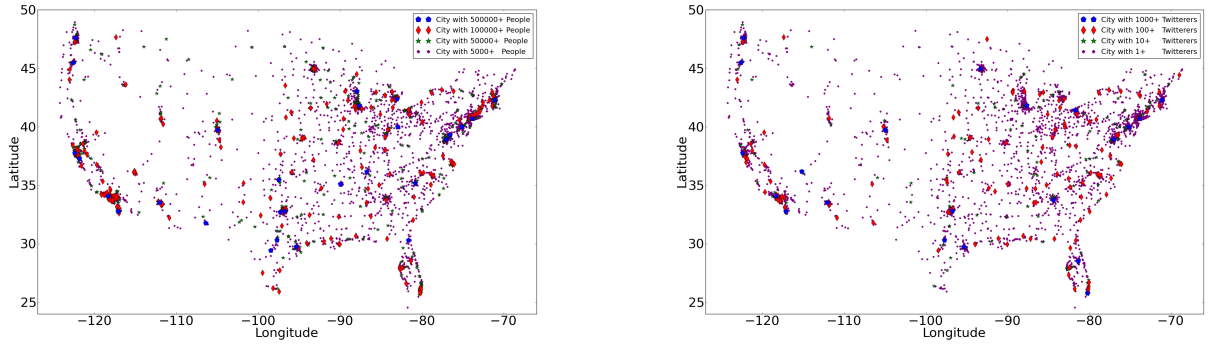
3. PRELIMINARIES

In this section, we briefly explain our dataset, formalize the research problem and describe the experimental setup.

3.1 Location Sparsity on Twitter

To derive a representative sample of Twitter users, we employed two complementary crawling strategies: crawling through Twitter's public timeline API and crawling by breadth-first search through social edges to crawl each user's friends (following) and followers. The first strategy can be considered as random sampling from active Twitter users (whose tweets are selected for the public timeline), while the second strategy extracts a directed acyclic sub-graph of the whole Twitter social graph, including less active Twitter users. We combine the two strategies to avoid bias in either one. Using the open-source library twitter4j [5] to access Twitter's open API [6] from September 2009 to January 2010, we collected a base dataset of 1,074,375 user profiles and 29,479,600 status updates.

Each user profile includes the capacity to list the user's name, location, a web link, and a brief biography. We find that 72.05% of the profiles collected do list a non-empty location, including locations like "Hollywood, CA", "England", and "UT: 40.708046,-73.789259". However, we find



(a) Population Distribution of the Continental United States

(b) User Distribution of Sampled Twitter Dataset

Figure 1: Comparison Between the Actual US Population and the Sample Twitter User Population

that most of these user-submitted locations are overly general with a wide geographic scope (e.g., California, worldwide), missing altogether, or nonsensical (e.g., Wonderland, "CALI to FORNIA"). Specifically, we examine all locations listed in the 1,074,375 user profiles and find that just 223,418 (21% of the total) list a location as granular as a city name and that only 61,335 (5%) list a location as granular as a latitude/longitude coordinate. This absence of granular location information for the majority of Twitter users (74%) indicates the great potential in estimating or recommending location for a Twitter user.

For the rest of the paper, we focus our study of Twitter user location estimation on users within the continental United States. Toward this purpose, we filter all listed locations that have a valid city-level label in the form of "cityName", "cityName, stateName", and "cityName, stateAbbreviation", where we consider all valid cities listed in the Census 2000 U.S. Gazetteer [1] from the U.S. Census Bureau. Even when considering these data forms, there can still be ambiguity for cities listed using just "cityName", e.g., there are three cities named Anderson, four cities named Arlington, and six cities called Madison. For these ambiguous cases, we only consider cities listed in the form "cityName, stateName", and "cityName, stateAbbreviation". After applying this filter, we find that there are 130,689 users (with 4,124,960 status updates), accounting for 12% of all sampled Twitter users. This sample of Twitter users is representative of the actual population of the United States as can be seen in Figure 1(a), and Figure 1(b).

3.2 Problem Statement

Given the lack of granular location information for Twitter users, our goal is to estimate the location of a user based purely on the content of their tweets. Having a reasonable estimate of a user's location can enable content personalization (e.g., targeting advertisements based on the user's geographical scope, pushing related news stories, etc.), targeted public health web mining (e.g., a Google Flu Trends-like system that analyzes tweets for regional health monitoring), and local emergency detection (e.g., detecting emergencies by monitoring tweets about earthquakes, fires, etc.). By focusing on the content of a user's Twitter stream, such an approach can avoid the need for private user information, IP address, or other sensitive data. With these goals in mind, we focus on city-level location estimation for a Twitter user, where the problem can be formalized as:

Location Estimation Problem: Given a set of tweets $S_{tweets}(u)$ posted by a Twitter 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 $l_{act}(u)$.

As we have noted, location estimation based on tweet content is a difficult and challenging problem. Twitter status updates are inherently noisy, often relying on shorthand and non-standard vocabulary. It is not obvious that there are clear location cues embedded in a user's tweets at all. A user may have interests which span multiple locations and a user may have more than one natural location.

3.3 Evaluation Setup and Metrics

Toward developing a content-based user location estimator, we next describe our evaluation setup and introduce four metrics to help us evaluate the quality of a proposed estimator.

Test Data: In order to be fair in our evaluation of the quality of location estimation, we build a test set that is separate from the 130,689 users previously identified (and that will be used for building our models for predicting user location). In particular, we extract a set of active users with 1000+ tweets who have listed their location in the form of latitude/longitude coordinates. Since these types of user-submitted locations are typically generated by smartphones, we assume these locations are correct and can be used as ground truth. We filter out spammers, promoters, and other automated-script style Twitter accounts using features derived from Lee et al.'s work [14] on Twitter bot detection, so that the test set will consist of primarily "regular" Twitter users for whom location estimation would be most valuable. Finally, we arrive at 5,190 test users and more than 5 million of their tweets. These test users are distributed across the continental United States similar to the distributions seen in Figure 1(a), and Figure 1(b).

Metrics: To evaluate the quality of a location estimator, we compare the estimated location of a user versus the actual city location (which we know based on the city corresponding to their latitude/longitude coordinates). The first metric we consider is the **Error Distance** which quantifies the dis-

tance in miles between the actual location of the user $l_{act}(u)$ and the estimated location $l_{est}(u)$. The **Error Distance** for user u is defined as:

$$ErrDist(u) = d(l_{act}(u), l_{est}(u))$$

To evaluate the overall performance of a content-based user location estimator, we further define the **Average Error Distance** across all test users U:

$$AvgErrDist(U) = \frac{\sum_{u \in U} ErrDist(u)}{|U|}$$

A low **Average Error Distance** means that the system can geo-locate users close to their real location on average, but it does not give strong insight into the distribution of location estimation errors. Hence, the next metric – **Accuracy** – considers the percentage of users with their error distance categorized in the range of 0-100 miles:

$$Accuracy(U) = \frac{|\{u | u \in U \wedge ErrDist(u) \leq 100\}|}{|U|}$$

Further, since the location estimator predicts k cities for each user in decreasing order of confidence, we define the **Accuracy with K Estimations (Accuracy@k)** which applies the same Accuracy metric, but over the city in the top-k with the least error distance to the actual location. In this way, the metric shows the capacity of an estimator to identify a good candidate city, even if the first prediction is in error.

4. CONTENT-BASED LOCATION ESTIMATION: OVERVIEW AND APPROACH

In this section, we begin with an overview of our baseline approach for content-based location estimation and then present two key optimizations for improving and refining the quality of location estimates.

Baseline Location Estimation: First, we can directly observe the actual distribution across cities for each word in the sampled dataset. Based on maximum likelihood estimation, the probabilistic distribution over cities for word w can be formalized as $p(i|w)$ which identifies for each word w the likelihood that it was issued by a user located in city i. For example, for the word "rockets", we can see its city distribution in Figure 2 based on the tweets in the sampled dataset (with a large peak near Houston, home of NASA and the NBA basketball team Rockets).

Of course users from cities other than Houston may tweet the word "rockets", so reliance on a single word or a single tweet will necessarily reveal very little information about the true location of a user. By aggregating across all words in tweets posted by a particular user, however, our intuition is that the location of the user will become clear. Given the set of words $S_{words}(u)$ extracted from a user's tweets $S_{tweets}(u)$, we propose to estimate the probability of the user being located in city i as:

$$p(i|S_{words}(u)) = \sum_{w \in S_{words}(u)} p(i|w) * p(w)$$

where we use $p(w)$ to denote the probability of the word w in the whole dataset. Letting $count(w)$ be the number of occurrences of the word w, and t be the total number of tokens

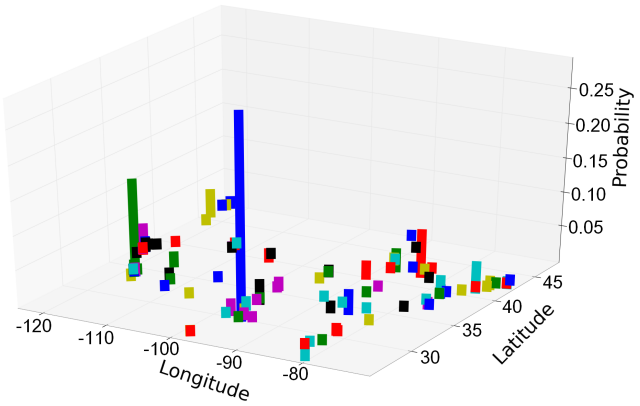


Figure 2: City estimates for the term "rockets"

in the corpus, we replace $p(w)$ with $\frac{count(w)}{t}$ in calculating the value of $p(w)$. Such an approach will produce a per-user city probability across all cities. The city with the highest probability can be taken as the user's estimated location. This location estimator is formalized in Algorithm 1.

Algorithm 1 Content-Based User Location Estimation

Input:
tweets: List of n tweets from a Twitter user u
cityList: Cities in continental US with 5k+ people
distributions: Probabilistic distributions for words
k: Number of estimations for each user
Output:
estimatedCities: Top K estimations

```
words = preProcess(tweets)
for city in cityList do
    prob[city] ← 0
    for word in words do
        prob[city] += distributions[word][city] * word.count
    end for
end for
estimatedCities = sort(prob, cityList, k)
return estimatedCities
```

Initial Results: Using this baseline approach, we estimated the location of all users in our test set using per-city word distributions estimated from the 130,689 users shown in Figure 1(b). For each user, we parsed their location and status updates (4,124,960 in all). In parsing the tweets, we eliminate all occurrences of a standard list of 319 stop words, as well as screen names (which start with @), hyperlinks, and punctuation in the tweets. Instead of using stemming, we use the Jaccard Coefficient to check whether a newly encountered word is a variation of a previously encountered word. The Jaccard Coefficient is particularly helpful in handling informal content like in tweets, e.g., by treating "awesoome" and "awesooome" as the word "awesome". In generating the word distributions, we only consider words that occur at least 50 times in order to build comparatively accurate models. Thus, 25,987 per-city word distributions are generated from a base set of 481,209 distinct words.

Disappointingly, only 10.12% of the 5,119 users in the

test set are geo-located within 100 miles to their real locations and the **AvgErrDist** is 1,773 miles, meaning that such a baseline content-based location estimator provides little value. On inspection, we discovered two key problems: (i) most words are distributed consistently with the population across different cities, meaning that most words provide very little power at distinguishing the location of a user; and (ii) most cities, especially with a small population, have a sparse set of words in their tweets, meaning that the per-city word distributions for these cities are underspecified leading to large estimation errors.

In the rest of this section, we address these two problems in turn in hopes of developing a more valuable and refined location estimator. Concretely, we pursue two directions:

- *Identifying Local Words in Tweets:* Is there a subset of words which have a more compact geographical scope compared to other words in the dataset? And can these "local" words be discovered from the content of tweets? By removing noise words and non-local words, we may be able to isolate words that can distinguish users located in one city versus another.
- *Overcoming Tweet Sparsity:* In what way can we overcome the location sparsity of words in tweets? By exploring approaches for smoothing the distributions of words, can we improve the quality of user location estimation by assigning non-zero probability for words to be issued from cities in which we have no word observations?

4.1 Identifying Local Words in Tweets

Our first challenge is to filter the set of words considered by the location estimation algorithm (Algorithm 1) to consider primarily words that are essentially "local". By considering all words in the location estimator, we saw how the performance suffers due to the inclusion of noise words that do not convey a strong sense of location (e.g., "august", "peace", "world"). By observation and intuition, some words or phrases have a more compact geographical scope. For example, "howdy" which is a typical greeting word in Texas may give the estimator a hint that the user is in or near Texas.

Toward the goal of improving user location estimation, we characterize the task of identifying local words as a decision problem. Given a word, we must decide if it is local or non-local. Since tweets are essentially informal communication, we find that relying on formally defined location names in a gazetteer is neither scalable nor provides sufficient coverage. That is, Twitter's 140 character length restriction means that users may not write the full address or location name (e.g., "t-center" instead of "Houston Toyota Center", home of the NBA Rockets team. Concretely, we propose to determine local words using a model-driven approach based on the observed geographical distribution of the words in tweets.

4.1.1 Determining Spatial Focus and Dispersion

Intuitively, a local word is one with a high local focus and a fast dispersion, that is it is very frequent at some central point (like say in Houston) and then drops off in use rapidly as we move away from the central point. Non-local words, on the other hand, may have many multiple central points with no clear dispersion (e.g., words like basketball). How do we assess the spatial focus and dispersion of words in tweets?

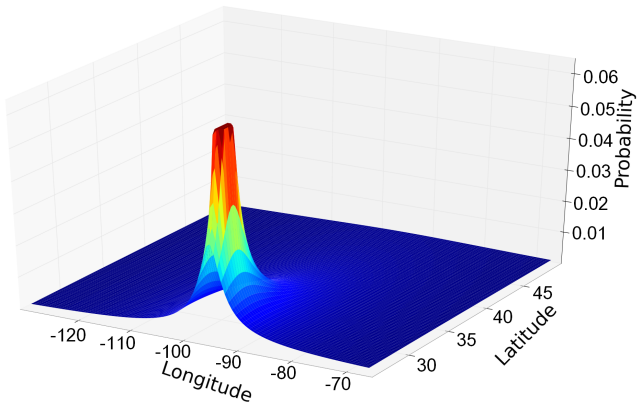


Figure 3: Optimized Model for the Word "rockets"

Recently Backstrom et al. introduced a model of spatial variation for analyzing the geographic distribution of terms in search engine query logs [8]. The authors propose a generative probabilistic model in which each query term has a geographic focus on a map (based on an analysis of the IP-address-derived locations of users issuing the query term). Around this center, the frequency shrinks as the distance from the center increases. Two parameters are assigned for each model, a constant C which identifies the frequency in the center, and an exponent α which controls the speed of how fast the frequency falls as the point goes further away from the center. The formula for the model is $Cd^{-\alpha}$ which means that the probability of the query issued from a place with a distance d from the center is approximately $Cd^{-\alpha}$. In the model, a larger α identifies a more compact geo-scope of a word, while a smaller α displays a more global popular distribution.

In the context of tweets, we can similarly determine the focus (C) and dispersion (α) for each tweet word by deriving the optimal parameters that fit the observed data. These parameters C and α are strong criteria for assessing a word's focus and dispersion, and hence, determining whether a word is local or not. For a word w, given a center, the central frequency C, and the exponent α, we compute the maximum-likelihood value like so: for each city, suppose all users tweet the word w from the city a total of n times, then we multiply the overall probability by $(Cd_i^{-\alpha})^n$; if no users in the city tweet the word w, we multiply the overall probability by $1 - Cd_i^{-\alpha}$. In the formula, d_i is the distance between city i and the center of word w. We add logarithms of probabilities instead of multiplying probabilities in order to avoid underflow. For example, let S be the set of occurrences for word w (indexed by cities which issued the word w), and let d_i be the distance between a city i and the model's center. Then:

$$f(C, \alpha) = \sum_{i \in S} \log C d_i^{-\alpha} + \sum_{i \notin S} \log \left(1 - C d_i^{-\alpha}\right)$$

is the likelihood value for the given center, C and α. Backstrom et al. also prove that $f(C, \alpha)$ has exactly one local maximum over its parameter space which means that when a center is chosen, we can iterate C and α to find the largest $f(C, \alpha)$ value (and hence, the optimized C and α). Instead of using a brute-force algorithm to find the optimized set

Table 1: Example Local Words

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

of parameters, we divide the map of the continental United States into lattices with a size of two by two square degrees. For the center in each lattice, we use golden section search [17] to find the optimized central frequency and the shrinking factor α. Then we zoom into the lattice which has the largest likelihood value, and use a finer-grained mesh on the area around the best chosen center. We repeat this zoom-and-optimize procedure to identify the optimal C, and α. Note that the implementation with golden section search can generate an optimal model for a word within a minute on a single modern machine and is scalable to handle web-scale data. To illustrate, Figure 3 shows the optimized model for the word "rockets" centered around Houston.

4.1.2 *Training and Evaluating The Model*

Given the model parameters C (focus) and α (dispersion) for every word, we could directly label as local words all tweet words with a sufficiently high focus and fast dispersion by considering some arbitrary thresholds. However, we find that such a direct application may lead to many errors (and ultimately poor user location estimation). For example, some models may lack sufficient supporting data resulting in a clearly incorrect geographic scope. Hence, we augment our model of local words with coordinates of the geo-center, since the geographical centers of local words should be located in the continental United States, and the count of the word occurrences, since a higher number of occurrences of a word will give us more confidence in the accuracy of the generated model of the word.

Using these features, we train a local word classifier using the Weka toolkit [20] – which implements several standard classification algorithms like Naive Bayes, SVM, AdaBoost, etc. – over a hand-labeled set of standard English words taken from the 3esl dictionary [2]. Of the 19,178 words in the core dictionary, 11,004 occur in the sampled Twitter dataset. Using 10-fold cross-validation and the SimpleCart classifier, we find that the classifier has a *Precision* of 98.8% and *Recall* and *F-Measure* both as 98.8%, indicating that the quality of local word prediction is good. After learning the classification model over these known English words, we apply the classifier to the rest of the 14,983 tweet words (many of which are non-standard words and not in any dictionary), resulting in 3,183 words being classified as local words.

To illustrate the geographical scope of the local words discovered by the classifier, five local word models are listed in Table 1. The word "automobile" is located around two hundred miles south of Detroit which is the traditional auto manufacturing center of the US. The word "casino" is located in the center of Las Vegas, two miles east of the North Las Vegas Airport. "tortilla" is centered a hundred miles south of the border between Texas and Mexico. The word "canyon" is located almost at the center of the Grand Canyon. The center for the word "redsox" is located 50 miles east of Boston, home of the baseball team.

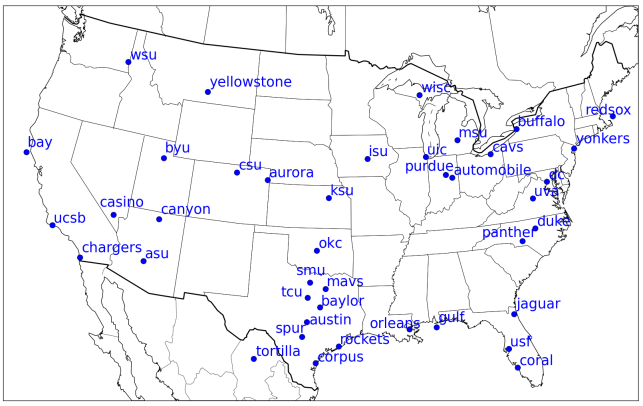


Figure 4: Geographical Centers of Local Words Discovered in Sampled Twitter Dataset

In order to visualize the geographical centers of the local favored words, a few examples are shown on the map of the continental United States in Figure 4. Based on these and the other discovered local words, we will evaluate if and how user location estimation improves in the experimental study in Section 5.

4.2 Overcoming Tweet Sparsity

The second challenge for improving our content-based user location estimator is to overcome the sparsity of words across locations in our sampled Twitter dataset. Due to this sparseness, there are a large number of "tiny" word distributions (i.e., words issued from only a few cities) The problem is even more severe when considering cities with a small population. As an example, consider the distribution for the word "rockets" over the map of the continental United States displayed in Figure 2. We notice that for a specific word, the probability for the word to be issued in a city can be zero since there are no tweets including the word in our sampled dataset. In order to handle this sparsity, we consider three approaches for smoothing the probability distributions: Laplace smoothing, data-driven geographic smoothing, and model-driven smoothing.

4.2.1 *Laplace Smoothing*

A simple method of smoothing the per-city word distributions is Laplace smoothing (add-one smoothing) which is defined as:

$$p(i|w) = \frac{1 + count(w,i)}{V + N(w)}$$

where $count(w,i)$ denotes the term count of word w in city i; V stands for the size of the vocabulary and N(w) stands for the total count of w in all the cities. Briefly speaking, Laplace smoothing assumes every seen or unseen city issued word w once more than it did in the dataset.

Although simple to implement, Laplace smoothing does not take the geographic distribution of a word into consideration. That is, a city near Houston with zero occurrences of the word "rockets" is treated the same as a city far from Houston with zero occurrences. Intuitively, the peak

for "rockets" in Houston (recall Figure 2) should impact the probability mass at nearby cities.

4.2.2 Data-Driven Geographic Smoothing

To take this geographic nearness into consideration, we consider two techniques for smoothing the per-city word distributions by considering neighbors of a city at different granularities. In the first case, we smooth the distribution by considering the overall prevalence of a word within a state; in the second, we consider a lattice-based neighborhood approach for smoothing at a more refined city-level scale.

State-Level Smoothing: For state-level smoothing, we aggregate the probabilities of a word w in the cities in a specific state s (e.g., Texas), and consider the average of the summation as the probability of the word w occurring in the state. Letting S_c denote the set of cities in the state s, the state probability can be formulated as:

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

Furthermore, the probability of the word w to be located in city i can be a combination of the city probability and the state probability:

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

where i stands for a city in the state s, and $1 - \lambda$ is the amount of smoothing. Thus, a small value of λ indicates a large amount of state-level smoothing.

Lattice-based Neighborhood Smoothing: Naturally, state-level smoothing is a fairly coarse technique for smoothing word probabilities. For some words, the region of a state exaggerates the real geographical scope of a word; meanwhile, the impact of a word issued from a city may have higher influence over its neighborhood in another state than the influence over a distant place in the same state. With this assumption, we apply lattice-based neighborhood smoothing.

Firstly, we divide the map of the continental United States into lattices of 1 x 1 square degrees. Letting w denote a specific word, lat a lattice, and S_c be the set of cities in lat, the per-lattice probability of a word w can be formalized as:

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

In addition, we consider lattices around (the nearest lattice in all eight directions) lat as the neighbors of the lattice lat. Introducing μ as the parameter of neighborhood smoothing, the lattice probability is updated as:

$$p'(lat|w) = \mu * p(lat|w) + (1.0 - \mu) * \sum_{lat_i \in S_{neighbors}} p(lat_i|w)$$

In order to utilize the smoothed lattice-based probability, another parameter λ is introduced to aggregate the real probability of w issued from the city i, and the probability of the smoothed lattice probability. Finally the lattice-based per-city word probability can be formalized as:

$$p'(i|w) = \lambda * p(i|w) + (1.0 - \lambda) * p'(lat|w)$$

where i is a city within the lattice lat.

4.2.3 Model-Based Smoothing

The final approach to smoothing takes into account the word models developed in the previous section for identifying C and α. Applying this model directly, where each word is distributed according to $Cd^{-\alpha}$, we can estimate a per-city word distribution as:

$$p'(i|w) = C(w)d_i^{-\alpha(w)}$$

where $C(w)$ and $\alpha(w)$ are taken to be the optimized parameters derived from the real data distribution of words across cities. This model-based smoothing ignores local perturbations in the observed word frequencies, in favor of a more elegant word model (recall Figure 3). Compared to the data-driven geographic-based smoothing, model-based smoothing has the advantage of "compactness", by encoding each word's distribution according to just two parameters and a center, without the need for the actual city word frequencies.

5. EXPERIMENTAL RESULTS

In this section, we detail an experimental study of location estimation with local tweet identification and smoothing. The goal of the experiments is to understand: (i) if the classification of words based on their spatial distribution significantly helps improve the performance of location estimation by filtering out non-local words; (ii) how the different smoothing techniques help overcome the problem of data sparseness; and (iii) how the amount of information available about a particular user (via tweets) impacts the quality of estimation.

5.1 Location Estimation: Impact of Refinements

Recall that in our initial application of the baseline location estimator, we found that only 10.12% of the 5,119 users in the test set could be geo-located within 100 miles of their actual locations and that the AvgErrDist across all 5,119 users was 1,773 miles. To test the impact of the two refinements – local word identification and smoothing – we update Algorithm 1 to filter out all non-local words and to update the per-city word probabilities with the smoothing approaches described in the previous section.

For each user u in the test set, the system estimates k (10 in the experiments) possible cities in descending order of confidence. Table 2 reports the Accuracy, Average Error Distance, and Accuracy@k for the original baseline user location estimation approach (*Baseline*), an approach that augments the baseline with local word filtering but no smoothing (*+ Local Filtering*), and then four approaches that augment local word filtering with smoothing – *LF+Laplace*, *LF+State-level*, *LF+Neighborhood*, and *LF+Model-based*. Recall that Accuracy measures the fraction of users whose locations have been estimated to within 100 miles of their actual location.

First, note the strong positive impact of local word filtering. With local word filtering alone, we reach an Accuracy of 0.498 which is almost five times as high as the Accuracy we get with the baseline approach that uses all words in the sampled Twitter dataset. The gap indicates the strength of the noise introduced by non-local words, which significantly affects the quality of user location estimation. Also consider that this result means that nearly 50% of the users in our test set can be placed in their actual city purely based on

Table 2: Impact of Refinements on User Location Estimation

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

an analysis of the content of their tweets. Across all users in the test set, filtering local words reduces the Average Error Distance from 1,773 miles to 539 miles. While this result is encouraging, it also shows that there are large estimation errors for many of our test users in contrast to the 50% we can place within 100 miles of their actual location. Our hypothesis is that some users are inherently difficult to locate based on their tweets. For example, some users may intentionally misrepresent their home location, say by a New Yorker listing a location in Iran as part of sympathy for the recent Green movement. Other users may tweet purely about global topics and not reveal any latent local biases in their choice of words. In our continuing work, we are examining these large error cases.

Continuing our examination of Table 2, we also observe the positive impact of smoothing. Though less strong than local word filtering, we see that Laplace, State-level, and Neighborhood smoothing result in better user location estimation than either the baseline or the baseline plus local word filtering approach. As we had surmised, the Neighborhood smoothing provides the best overall results, placing 51% of users within 100 miles of their actual location, with an Average Error Distance over all users of 535 miles.

Comparing State-level smoothing to Neighborhood smoothing, we find similar results with respect to the baseline, but slightly better results for the Neighborhood approach. We attribute the slightly worse performance of state-level smoothing to the regional errors introduced by smoothing toward the entire state instead of a local region. For example, state-level smoothing will favor the impact of words emitted by a city that is distant but within the same state relative to a words emitted by a city that is nearby but in a different state.

As a negative result, we can see the poor performance of model-based smoothing, which nearly undoes the positive impact of local word filtering altogether. This indicates that the model-based approach overly smooths out local perturbations in the actual data distribution, which can be useful for leveraging small local variations to locate users.

To further examine the differences among the several tested approaches, we show in Figure 5 the error distance in miles versus the fraction of users for whom the estimator can place within a particular error distance. The figure compares the original baseline user location estimation approach (*Baseline*), the baseline approach plus local word filtering (*+ Local Filtering*), and then the best performing smoothing approach (*LF+Neighborhood*) and the worst performing smoothing approach (*LF+Model-based*). The x-axis identifies the error distance in miles in log-scale and the y-axis quantifies the fraction of users located within a specific error distance. We can clearly see the strong impact of local word filtering and the minor improvement of smoothing across all

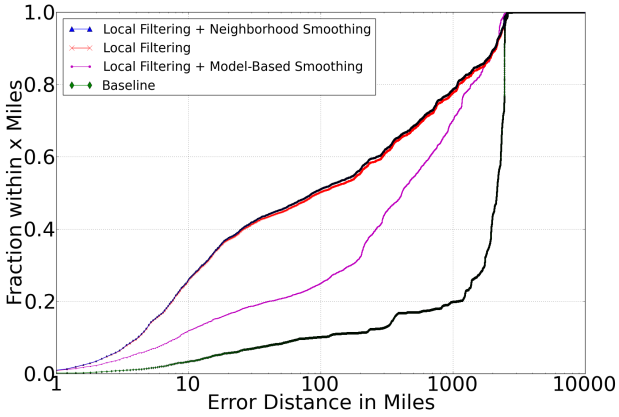


Figure 5: Comparison Across Estimators

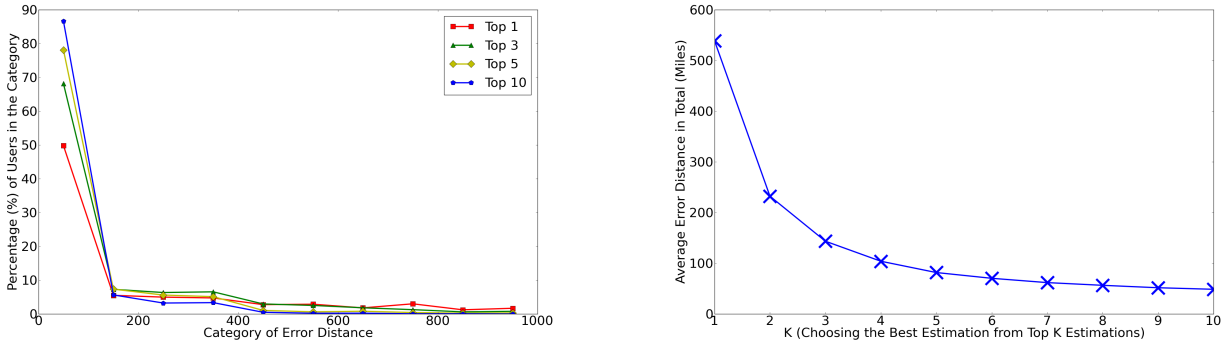
error distances. For the best performing approach, we can see that nearly 30% of users are placed within 10 miles of their actual location in addition to the 51% within 100 miles.

5.2 Capacity of the Estimator

To better understand the capacity of the location estimator to identify the correct user location, we next relax our requirement that the estimator make only a single location prediction. Instead, we are interested to see if the estimator can identify a good location somewhere in the top-k of its predicted cities. Such a relaxation allows us to appreciate if the estimator is identifying some local signals in many cases, even if the estimator does not place the best location in the top most probable position.

Returning to Table 2, we report the Accuracy@k for each of the approaches. Recall Accuracy@k measures the fraction of users located within 100 miles of their actual location, for some city in the top k predictions of the estimator. For example, for Accuracy@5 for *LF+Neighborhood* we find a result of 0.788, meaning that within the first five estimated locations, we find at least one location within 100 miles of the actual location in 79% of cases. This indicates that the content-based location estimator has high capacity for accurate location estimation, considering the top-5 cities are recommended from a pool of all cities in the US. This is a positive sign for making further refinements and ultimately to improving the top-1 city prediction.

Similarly, Figure 6(a) shows the error distance distribution for varying choices of k, where each point represents the fraction of users with an error in that range (i.e., the first point represents errors of 0-100 miles, the second point 100-200 miles, and so on). The location estimator identifies a city in the top-10 that lies within 100 miles of a user's



(a) Error Distance Distribution

(b) Average Error Distance

Figure 6: Capacity of the Location Estimator: Using the Best Estimation in the Top-k

actual city in 90% of all cases. Considering the top-1, top-3, top-5, and top-10, we can see that the location estimator performs increasingly well. Figure 6(b) continues this analysis by reporting the Average Error Distance as we consider increasing k. The original reported error of around 500 miles for the top-1 prediction drops as we increase k, down to just 82 miles when we consider the best possible city in the top-10.

5.3 Estimation Quality: Number of Tweets

An important question remains: how does the quality of estimation change with an increasing amount of user information? In all of our experiments so far, we have considered the test set in which each user has 1000+ tweets. But perhaps we can find equally good estimation results using only 10 or 100 tweets?

To illustrate the impact of an increasing amount of user data, we begin with a specific example of a test user with a location in Salt Lake City. Figure 7 illustrates the sequence of city estimations based on an increasing amount of user tweet data. With 10 tweets, Chicago has the dominant highest estimated probability. With 100 tweets, several cities in California, Salt Lake City and Milwaukee exceed Chicago. By 300 tweets, the algorithm geo-locates the user in the actual city, Salt Lake City; however there is still significant noise, with several other cities ranking close behind Salt Lake City. By 500 tweets, the probability of Salt Lake City increases dramatically, converging on Salt Lake City as the user data increases to 700 tweets and then 1000 tweets.

To quantify the impact of an increasing amount of user information, we calculate the distribution of Error Distance and the Average Error Distance across all of the test users based on the Local Word filtering location estimator relying on a range of tweets from 100 to 1000. Figure 8(a) shows the error distance distribution, where each point represents the fraction of users with an error in that range (i.e., the first point represents errors of 0-100 miles, the second point 100-200 miles, and so on). The errors are distributed similarly; even with only 100 tweets, more than 40% of users are located within 100 miles. In Figure 8(b), we can see that with only 100 tweets that the Average Error Distance is 670 miles. As more tweets are used to refine the estimation, the error drops significantly. This suggests that as users continue to tweet, they "leak" more location information which can result in more refined estimation.

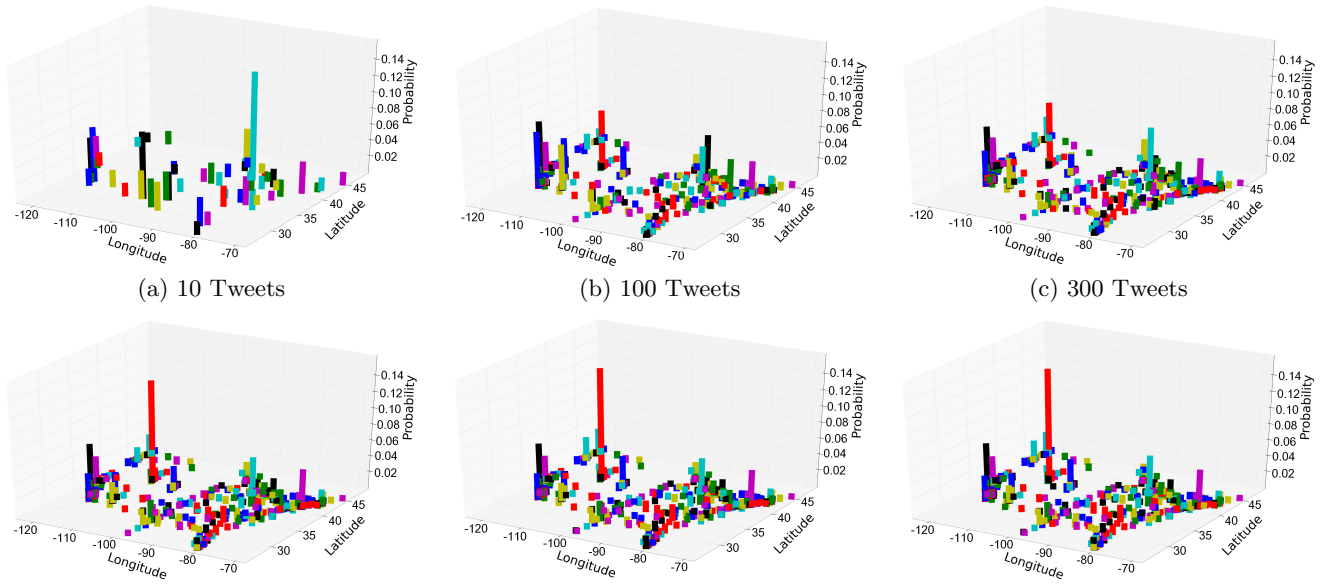
6. CONCLUSION

The promise of the massive human-powered sensing capabilities of Twitter and related microblogging services depends heavily on the presence of location information, which we have seen is largely absent from the majority of Twitter users. To overcome this location sparsity and to enable new location-based personalized information services, we have proposed and evaluated a probabilistic framework for estimating a Twitter user's city-level location based purely on the content of the user's tweets, even in the absence of any other geospatial cues. The content-based approach relies on two key refinements: (i) a classification component for automatically identifying words in tweets with a strong local geo-scope; and (ii) a lattice-based neighborhood smoothing model for refining a user's location estimate. We have seen how the location estimator can place 51% of Twitter users within 100 miles of their actual location.

As a purely data-driven approach, we anticipate continued refinement of this approach through the incorporation of more data (in the form of wider coverage of Twitter users and their associated tweets). We are also interested to combine the purely content-based approach here with social network inference based approaches for combining location evidence of social ties in the estimator. We are also interested to further explore the temporal aspect of location estimation, to develop more robust estimators that can track a user's location over time.

7. REFERENCES

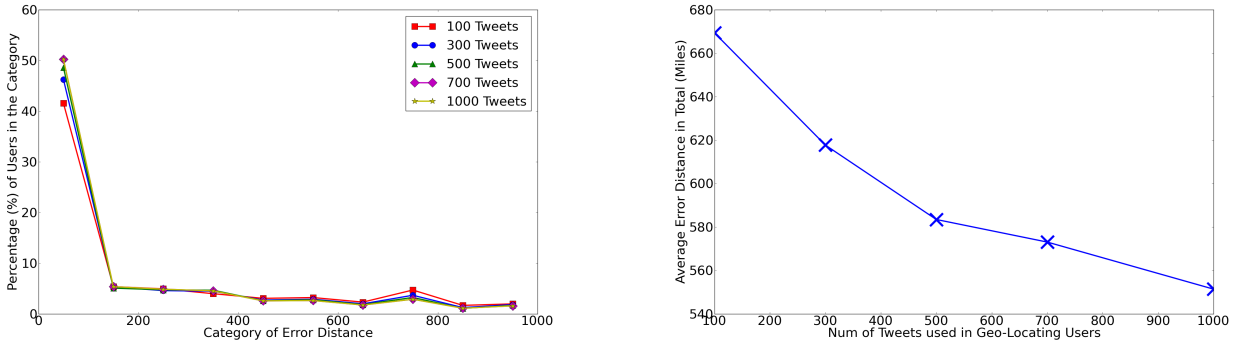
[1] Census 2000 u.s. gazetteer. http://www.census.gov/geo/www/gazetteer/places2k.html.
[2] Kevin's word list. http://wordlist.sourceforge.net.
[3] The local business owner's guide to twitter. http://domusconsultinggroup.com/wp-content/uploads/2009/06/090624-twitter-ebook.pdf.
[4] New data on twitter's users and engagement. http://themetricsystem.rjmetrics.com/2010/01/26/new-data-on-twitters-users-and-engagement/.
[5] Twitter4j open-source library. http://yusuke.homeip.net/twitter4j/en/index.html.
[6] Twitter's open api. http://apiwiki.twitter.com.
[7] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. Web-a-where: geotagging web content. In *SIGIR*, 2004.



(a) 10 Tweets (b) 100 Tweets (c) 300 Tweets

(d) 500 Tweets (e) 700 Tweets (f) 1,000 Tweets

Figure 7: Example: Location Estimation Convergence as Number of Tweets Increases



(a) Error Distance Buckets with Different # of Tweets (b) Average Error Distance with Different # of Tweets

Figure 8: Refinement of Location Estimation with Increasing Number of Tweets

[8] L. Backstrom, J. Kleinberg, R. Kumar, and J. Novak. Spatial variation in search engine queries. In *WWW*, 2008.

[9] L. Backstrom, E. Sun, and C. Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In *WWW*, 2010.

[10] D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world's photos. In *WWW*, 2009.

[11] C. Fink, C. Piatko, J. Mayfield, T. Finin, and J. Martineau. Geolocating blogs from their textual content. In *AAAI 2009 Spring Symposia on Social Semantic Web: Where Web 2.0 Meets Web 3.0*, 2009.

[12] R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. Social network classification incorporating link type. In *IEEE Intelligence and Security Informatics*, 2009.

[13] M. Hurst, M. Siegler, and N. Glance. On estimating the geographic distribution of social media. In *ICWSM*, 2007.

[14] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: Social honeypots + machine learning. In *SIGIR*, 2010.

[15] J. Lin and A. Halavais. Mapping the blogosphere in america. In *Workshop on the Weblogging Ecosystem at the 13th International World Wide Web Conference*, 2004.

[16] J. Lindamood, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. Inferring private information using social network data. In *WWW*, 2009.

[17] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. 1986.

[18] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW*, 2010.

[19] P. Serdyukov, V. Murdock, and R. van Zwol. Placing flickr photos on a map. In *SIGIR*, 2009.

[20] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*. Morgan Kaufmann, June 2005.

[21] S. Yardi and D. Boyd. Tweeting from the town square: Measuring geographic local networks. In *ICWSM*, 2010.

[22] W. Zong, D. Wu, A. Sun, E.-P. Lim, and D. H.-L. Goh. On assigning place names to geography related web pages. In *JCDL*, 2005.