Tweecalization: Efficient and Intelligent Location Mining in Twitter Using Semi-Supervised Learning

Satyen Abrol, Latifur Khan and Bhavani Thuraisingham
Department of Computer Science
University of Texas at Dallas
Richardson, Texas, USA
{abrol, lkhan, bhavani.thuraisingham}@utdallas.edu

***Abstract*—Geosocial Networking is the new hotness, with social networks providing services and capabilities to the users to associate location to their profiles. But, because of privacy and security reasons, most of the people on social networking sites like Twitter are unwilling to provide locations in their profiles. This creates a need for an algorithm that predicts the location of the user based on the implicit attributes associated with him. In this paper, we develop a tool, Tweecalization that predicts the location of the user purely on the basis of his social network, using the strong theoretical framework of semi-supervised learning. In particular we employ the label propagation algorithm. On the city locations returned by the algorithm, the system performs agglomerative clustering based on geospatial proximity and their individual scores to return cluster of locations with higher confidence. We perform extensive experiments to show the validity of our system in terms of both accuracy and running time. Experimental results show that Tweecalization outperforms the content based geo-tagging approach and the Tweethood algorithm [4] in both accuracy and running time.**

Twitter; Location Mining; Label Propagation; Social Computing

I. INTRODUCTION

In 2005, Social Media was considered to be a fad, with a short life span. As of 2012, social networking is poised to be the biggest change since the industrial revolution. It has already become the number one activity on the internet. It took 13 years for television to reach 50 million viewers, and Facebook added 100 million users in just over 9 months. If Facebook was a country, it would be 3rd largest in the world after China and India [2].

Privacy concerns with social networking services have become a controversial and much publicized topic since the creation and increasing popularity of social networking sites such as Bebo, MySpace and the currently most used social networking site, Facebook. Issues relating to stalking, identity theft, sexual predators and employment are consistently on the rise. Also in regards to these social networking sites the ethics regarding data storage, the management and sharing of such data is still a big concern. A security issue occurs when a hacker gains unauthorized access to a site's protected coding or written language. Privacy issues don't necessarily have to involve security breaches. The potential harm to an individual user really boils down to how much a user engages in a social networking site, as well as the amount of the private information they're willing to share. And location is one of the most important attribute.

Trustworthiness is another reason which makes location discovery so important. It is well known that the "Arab Spring" in Tunisia, Egypt and elsewhere in the Mid-East heavily relied on the Internet, social media and technologies like Twitter, TwitPic, Facebook and YouTube in the early stages to accelerate social protest. The Department of State has effectively used social networking sites to gauge the sentiments within societies. Maintaining a social media presence in deployed locations also allows commanders to understand potential threats and emerging trends within the regions. The online community can provide a good indicator of prevailing moods and emerging issues. Many of the vocal opposition groups will likely use social media to air grievances publicly. In such cases and others similar to these, it becomes very important for organizations like the US State Department to be able to verify the location of the users posting these messages.

Finally, let us discuss the impact of social media in marketing and garnering feedback from consumers. First Social media gives marketers a voice and a way to communicate with peers, customers and potential consumers. It personalizes the "brand" and helps you to spread your message in a relaxed and conversational way. The second major contribution of social media towards business is for getting the feedback from users. Social media gives you the ability to get the kind of quick feedback inbound marketers require staying agile. Large corporations like Walmart and Starbucks are leveraging social networks beyond your typical posts and updates to get feedback on the quality of their products and services, especially ones that have been recently launched on Twitter [2].

Having said all this, it is important to understand that user's location is not easily accessible due to security and privacy concerns, thus impeding the growth of location based services in the present world scenario. By conducting experiments to find locations of 1 million users, we found that only 14.3% specify their locations explicitly.

That leaves us with the option to mine the location of the user, which is not an easy task in itself. As we shall later show, traditional text based location extraction techniques [7, 8, and 9] do not perform well in the domain of social

COLLABORATECOM 2012, October 14-17, Pittsburgh, United States

DOI 10.4108/icst.collaboratecom.2012.250520

networks. This is primarily because of the presence of multiple locations mentioned in the text and lack of relationship between the actual location of the user and the location mentioned in the text. Next, the high running time associated with such mining techniques makes them unusable in practical scenarios where real-time computation of location is necessary.

Figure 1. Twitter message of a user whose actual location is Japan.

In this paper we demonstrate the development of Tweecalization, a tool that identifies the location of the user on social networking sites and outperforms the typical gazetteer based approach in both accuracy and running time.

Since only a small fraction of users explicitly provide a location (labeled data), the problem of determining location of users (unlabeled data) based on the social network is a classic example of a scenario where the semi-supervised learning algorithm fits in. We demonstrate how Tweecalization applies label propagation on the graph constructed from the social network of the user to identify his location. On the city locations returned by the algorithm, the system performs agglomerative clustering to return cluster of locations with higher confidence. While doing this, we address several challenges, including defining the edge in an intelligent manner based on trustworthiness and explain how the use of this metric allows for detecting migration in users, the algorithm also identifies spammers and celebrities, and does clustering of locations.

In our work, we make several contributions. First, Tweecalization uses label propagation, a semi-supervised learning algorithm, for determining location of user from his social network. Second, it uses agglomerative clustering to ensure that the locations are returned with some minimum confidence. Finally, Tweecalization outperforms the content based location mining approach and our previously proposed approach, Tweethood, in both accuracy and running time.

The research paper is organized as follows. Section 2 surveys the related work in this domain and points out the novelty in our approach. Section 3 discusses the challenges faced in identifying the location of the user. Section 4 studies and establishes the relation between geospatial proximity and friendship, explains why migration is too important a phenomenon to be ignored and describes the label propagation algorithm. Section 6 discusses the experiments and the observations on various approaches and their time complexities. We conclude in section 7, by giving a few pointers for future work.

II. Related Work

Location identification and geo tagging of documents is a well-known topic in data mining and a lot of prior work exists on it. Social Networking on the other hand is still a very new field of computer science and little or no previous work has been done towards identifying the location of users based on their social activity. In this section we do a brief survey of the previous works on geo-tagging documents based on the content.

Most of the research can be broadly classified into two categories. One, involving the concepts of Natural Language Processing [18, 19, 20] and the other using data mining approach [21, 22]. Most of the work done using NLP techniques consists of input text that is structured and well edited. Li [7] combines these two methodologies and uses a 5-step approach, first short listing the keywords appearing in the gazetteer and then applying NLP techniques to remove non geo terms. A precision of 93.8% is reported using this approach.

Mehler [9] develops a model for estimating and evaluating spatial significance of entities using NLP techniques. Liu et al. [10] do a similar geo-analysis of the impact of location of the source on the viewpoint presented in news articles. Sheng in [11] discusses the need for reordering the search results (like food, sports, etc.) based on user references obtained by analyzing user's location.

Lieberman [17] describes the construction of a spatio-textual search engine using the gazetteer and NLP tools, a system for extracting, querying and visualizing textual references to geographic locations in unstructured text documents. They use an elaborate technique for removing the stop words, using a hybrid model of *Part-of-Speech (POS)* and *Named-Entity Recognition* (NER) tagger. POS tagger helps to identify the nouns and the NER tagger annotates them as person, organization, or location. But this system does not work well for text where name of a person is ambiguous with a location. E.g. Jordan might mean Michael Jordan, the basketball player or it might mean the country Jordan. In that case the NER tagger might remove Jordan considering it to be name of a person. For removing geo-geo ambiguity they use the *pair strength* algorithm. Pairs of feature records are compared to determine whether or not they give evidence to each other, based on the familiarity of each location, frequency of each location, as well as their document and geodesic distances.

Amitay [8] present a way of determining the page focus of web pages using the gazetteer approach after pruning the data. They are able to correctly tag individual name place occurrences 80% of the time and are able to recognize the correct focus of the pages 91% of the time. But they have a low accuracy for the geo/non-geo disambiguation.

The task of identifying location mentioned in documents or messages is very different from identifying the location of user from the messages posted by him/her. That is, even if page focus of the messages is identified correctly, that may not be the correct location of the user. E.g. people express their opinions on political issues around the world all the time. The revolutionary wave of demonstrations and protests occurring in the Arab world, led to many messages having references to the Arab countries. Or the recent economic crisis in Europe may result in tweets mentioning Greece. In addition to this, the time complexity of text based geo-tagging messages is very large making it unsuitable for real time applications like

opinion mining. Thus, as we will show in our experiment section, the geo-tagging of user messages is an inaccurate method to identify location of the user.

In our previous approach, Tweethood [4], we propose a method based on *k*-nearest neighbor algorithm and predict location of the user based on the location of his closest friends. This methodology relies on closest training examples in the feature space for identifying the location and is hence based on the fully supervised learning model. We modify the algorithm to determine the locations of unlabeled users by going further deep in the graph. But that leads to a running time that is exponential in terms of the number of vertices. Additionally, Tweethood is an un-intelligent location mining approach which does not differentiate between various users and hence has no way of taking user migration into account. Tweecalization on the other hand introduces a trustworthiness measure which defines the similarity between two users in an astute way which captures social phenomenon of migration to correctly identify the most current location of user.

Tweecalization makes an important contribution in the field of identifying the location of a user based on his social network based on a strong theoretical framework of label propagation. We demonstrate how the problem of identification of location of a user can be mapped to a semi supervised learning problem. We conduct a variety of experiments to show the validity of our approach and how it outperforms the traditional gazetteer based text mining approach and our previous approach, Tweethood, in both accuracy and running time.

III. CHALLENGES

As discussed previously, a lot of efforts are being made on the part of the social networking companies to incorporate location information in the communication. Twitter recently acquired Mixer Labs, a maker of geo-location Web Services, to boost up its location based services campaign and compete with the geo savvy mobile social networking sites like Foursquare and Gowalla. Nowadays, on logging into your Twitter account, you are given the option to add location (city level) to your messages.

But still, these efforts are not paying dividends simply because of several security and privacy reasons. And there is no incentive for users. We conducted an experiment and found that out of 1 million users on Twitter; only 14.3% actually share their location explicitly. Hence explicitly mentioned locations are rare and untrustworthy in certain cases where the user has mal-intent. That leaves us with the question: can the location be mined from implicit information associated with the users like the content of messages posted by them and nature of their social media network?

A. Location Mining from Social Network of User (LMSU)

This approach makes use of the social network of the user to identify his home location. Here, the social network of the user comprises of *followers* (people following the user) and *following* (people he is following). This approach gives us an insight into a user's close friends and the celebrities he is following. Intuitively, most of a person's friends are from same country and also, a person is more likely to follow celebrities that are from his own country. In other words, an American's friends are mostly Americans and he has a higher probability of following US President than Asian users.

There are certain technical challenges that need to be addressed before we can mine the location from the social network. First, only a small percentage of the users with public profiles are willing to share their location on Twitter for privacy and security reasons. Second, it is necessary to identify spammers and celebrities since they cannot be dealt in the same way as other users because of the differences in the properties associated with their social graphs. Third, we need to come up with an objective function that captures 'friendship' in the best manner for constructing the graphs for application of proposed algorithms.

Additionally, in certain cases this approach may face certain outliers. At the country level, it is not always safe to assume that a person always follows celebrities from his own country. Queen Rania of Jordan advocates for global education and thus has followers around the world. In such cases, judging the location of a user based on the celebrities he is following can lead to inaccurate results.

B. Location Mining from Text (LMT)

Twitter, being a popular social media site, is a way by which users generally express their opinions, with frequent references to locations including cities, countries etc. It is also intuitive in such cases to draw a relation between such locations mentioned in the *tweets* and the place of residence of the user. In other words a message from a user supporting the Longhorns (Football team for University of Texas at Austin) is most likely from a person living in Austin, Texas, USA than from someone in Australia.

Let us discuss some of the technical challenges posed by this approach. As previously mentioned, the identification of location of a user from the messages is a very different task from identification of the locations in web pages or other media. Twitter messages consist of text that is unstructured and more often than not have grammatical and spelling errors. The existing NER and POS taggers require the text to be structured and hence do not perform well on such data. Therefore, it becomes more difficult to identify the location from them. The other major issue that one faces in identification of a location concept is that unlike other sources of information like web pages, news articles etc., Twitter messages consist of multiple concept classes, i.e. several locations may be mentioned in the messages collected from a single user. In such a case identification of a single location that acts as page's focus can be a difficult task.

Even if the algorithm is able to identify possible location concepts, we still need to disambiguate them correctly. There are two types of ambiguities that exist: Geo/Non-Geo and Geo/Geo ambiguities. Geo/Non-Geo ambiguity is the case of a place name having another, non-geographic meaning, e.g. Paris might be the capital of France or might refer to the socialite and celebrity Paris Hilton. Geo/Geo ambiguity arises from two concepts having the same name but different geographic locations, e.g. Paris is the capital of France and is also a city in Texas.

In addition to this, as evident, the gazetteer based approach may prove to be inaccurate in cases where the user talks about news making incidents in other parts of the world. E.g. Haiti was a popular geo-reference in *tweets* after the earthquake. In another case, someone who talks about going to Venice for a vacation is not necessarily Italian.

IV. Location Mining From Social Network Of User (LMSU)

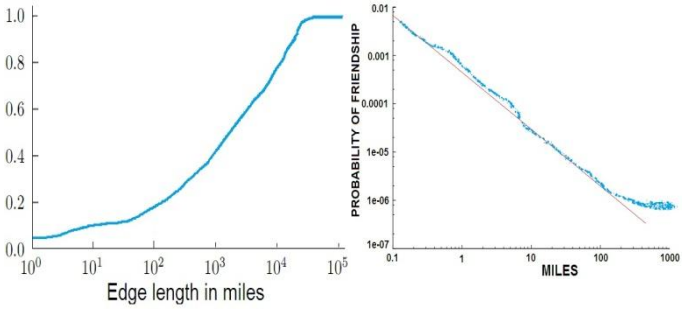
A. Geospatial Proximity and Friendship

Before we describe the label propagation algorithm, we would like to study and establish a relationship between geospatial proximity and friendship, particularly in Twitter. We hypothesize that there is a direct relation between geographical proximity and probability of friendship on Twitter. In other words, even though we live in the internet age, where distances actually don't matter and people can communicate with people across the globe, users tend to bring people from their offline friends into their online world. The relationship between friendship and geographic proximity in Online Social Networks (OSNs) has been studied in detail previously also by Backstrom for Facebook [23], Liben-Nowell for LiveJournal [15]. We perform our own set of experiments to understand the nature of friendships on Twitter, and study the effect of geographical proximity on friendship.

We formulate 10 million friendship pairs in which locations of both users are known, i.e. we considered only labeled data. It is important to understand our initial definition of *friendship* on Twitter, that A and B are friends if A *follows* B or B *follows* A. We divide the edge distance for the pairs into buckets of 10 miles. We determine the Cumulative Distribution Function, to observe the probability as a continuous curve. Fig 2(a) shows the results of our findings. It is interesting to note that only 10% of the pairs have the users within 100 miles and 75% of the users are at least 1000 miles from each other. That is, the results are contrary to the hypothesis we proposed and to the findings of Backstrom for Facebook, Liben-Nowell for LiveJournal. We delve deeper to analyze the social aspect of relationships on Twitter and find it to be very different from other OSNs like Facebook, LiveJournal, etc. The relationships on Twitter have a direction to them, meaning A *following* B, unlike Facebook or LinkedIn does not guarantee B also *follows* A. In addition to this, the presence of celebrities (followed by large number of users) and spammers (following large number of users) adds unique features to the graph.

These observations make us redefine the concept of *friendship* on Twitter and we make it somewhat stricter. According to our new definition, two users, A and B are friends if and only if A is *following* B and B also *follows* A back.

We form 10^{12} user pairs and identify the geographical distance between them. And then we divide the dataset into buckets of 0.1 mile and see as to what percentage of them actually have an edge (are friends). Fig 2(b) shows the probability of friendship versus the distance (in miles) distribution. The results for Twitter are very similar to that studied for LiveJournal and Facebook. The curve follows the power law having a curve of the form $a(x+b)^{-c}$ with exponent of -0.87and for distances greater than 1000 miles becomes a straight horizontal line.



(a) (b)

Figure 2. (a) shows the Cumulative Distribution Function, to observe the probability as a continuous curve and (b) shows the probability vs distance for 10^{12} users.

By doing this, we ensure that the other user is neither a celebrity (since celebrities don't follow fans back) nor a spammer (because no one wants to follow a spammer!). And a two way edge also means that the user A knows B and thus B is not some random user following A. And finally, the chances of A being interested in messages of B and vice versa without them being friends are pretty slim.

B. Twitter Users and Migration

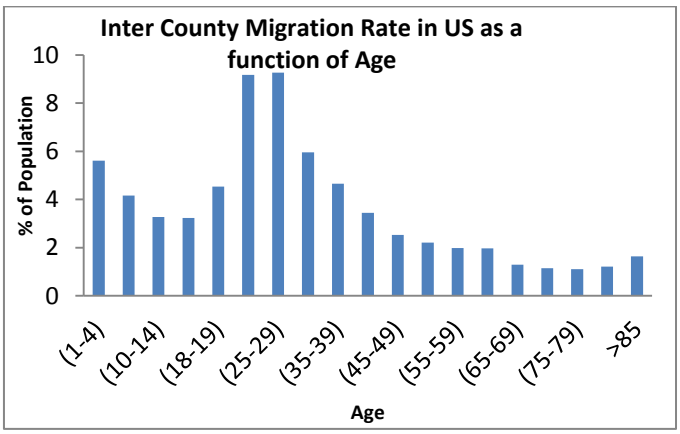


Figure 3. The inter county migration rate in the US as a function of age

Next, we try to link the migration effect to the users on Twitter using data from the U.S. census for the year 2008-09 and the Twitter age demographics studied by Edision Research [1]. First, we study the migration rate as a function of age, dividing the age groups in buckets of 5 years. There are two key observations that we make from the graph in Fig 3. First, we see a distinguishably high migration rate of over 9% for users in the age groups from 20 to 24 years and 25 to 29 years. This is consistent with our intuition, that after completion of high school, people have a tendency to move out of their hometowns for college or jobs. The second observation is that the migration rate decreases strictly with

age. This is also intuitive, since as we grow older there are increased chances of employment stability and people with families prefer to settle down.

The second part in linking is the study of demographics. Fig 4 shows the graph for the age distribution for Twitter users as surveyed by Edision Research [1]. The interesting observation is that 25-34 year olds make up a third of the Twitter population. Based on these two observations we can infer that Twitter users have a high tendency to migrate.

That leaves us with some important questions as to how do we know from a bunch of locations which one is the most current location of the user? Is there a way to define friendship to implicitly capture its temporal nature?

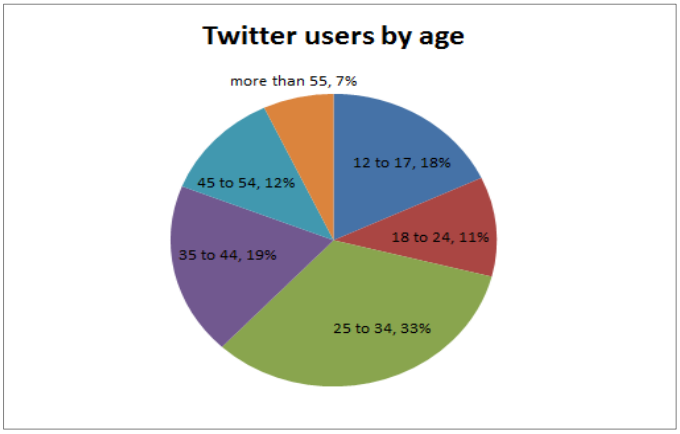


Figure 4. The distribution of Twitter users according to age

C. *Label Propagtion with Variable Depth*

Definition 1 (location concept) A location concept C of a user U is the location of the user in the format {City} X/ {State} Y/ {Country} Z. And for each location depending on the level of detail, either of X, Y or/and Z can be null.

Definition 2 (keyword) A keyword, K, is a word that has been tagged as a proper noun and is a candidate for a location.

Graph related approaches are the methods that rely on the social graph of the user while deciding on the location of the user. As observed earlier, the location data of users on social networks is a rather scarce resource and only available to a small portion of the users. This creates a need for a methodology that makes use of both labeled and unlabeled data for training. In this case, the location concept serves the purpose of class label. Therefore, our problem is a classic example for the application of semi-supervised learning algorithms.

In this section we propose a semi-supervised learning method for label propagation based on the algorithm proposed by Zhu and Ghahramani surveyed in [13] with strong theoretical foundation, where labeled data act like sources that push out labels through unlabeled data.

Before we begin explaining the algorithm, we briefly describe the theoretical framework that lies beneath the label propagation and how it is different from the *k*-nearest neighbor approach. The labeled propagation algorithm is based on transductive learning. In this environment, the data set is divided into two sets. One is the training set, consisting of the labeled data. And on the basis of this labeled data we try to predict the class for the second set, called the test or validation data consisting of unlabeled data. On the other hand the *k*-nearest neighbor (*k*-NN) approach is based on the inductive learning, in which, based on the training set, we try to determine a prediction function that tries to determine the class for the test set correctly. The major disadvantage with *k*-NN approach is that in certain cases, predicting the model based on the test set becomes a difficult task. For example, in our case if we try to determine the number of neighbors we need to consider for optimal accuracy, based on some users from (training data), this approach may not always produce the best results for other users. Hence, finding a value of *k* that works best for all instances of users seems a rather impossible task.

Chapelle [16] propose something called the "semi-supervised smoothness assumption". It states that if two points x_1 and x_2 in a high-density region are close, then so should be the corresponding outputs y_1 and y_2. This assumption implies that if two points are linked by a path of high density (e.g., if they belong to the same cluster), then their outputs are likely to be close. If, on the other hand, they are separated by a low-density region, then their outputs need not be close.

We divide the dataset into two parts. The first part consists of the labeled data $(U_1, L_1)...$ (U_l, L_l) of the form (user, location) where $\{L_1...L_l\} \varepsilon \{C_1...C_p\}$ (C_k is a location concept as discussed in the previously). And the second part of dataset has the unlabelled data $(U_{l+1}, L_{l+1})...$ (U_{l+u}, L_{l+u}). The pair (U_{l+u}, L_{l+u}) corresponds to the user whose location is to be determined.

Algorithm 1: Label Propagation (User, depth)

Input: User and the depth of the graph
Output: Location vector of the User
1: Compute the friends of User for maximum depth
2: Calculate similarity weight matrix W
3: Calculate the diagonal matrix D
4: Initialize $L^{(0)}$
5: Until L(t) converges
6: $L^{(t)} = D^{-1} . W. L^{(t-1)}$
7: $L_l^{(t)} = L_l^{(t-1)}$
8: Return $L_l^{(\infty)}[n+1]$

Algorithm 1. Label Propagation algorithm to determine location of a user

First, we need to construct a weight matrix W of dimensions $(n+1) \times (n+1)$ where W_{ij} is the measure of similarity between the two users U_i and U_j.

D. *Trustworthiness and Similarity Measure*

Just like any other machine learning technique, in label propagation also, the single most important thing is the way we define similarity (or distance) between two data points or,

in this case, users. All the existing graph based techniques, including [23] and [4] either build a probabilistic model or simply look at the location of the friends to predict the location. In other words, these techniques are un-intelligent and have the common flaw that not all friends are equally credible when suggesting locations for the primary user. We introduce the notion of trustworthiness for two specific reasons. First, we want to differentiate between various friends when propagating the labels to the central user and second, to implicitly take into account the social phenomenon of migration and thus provide for a simple yet intelligent way of defining similarity between users.

Trustworthiness (TW) is defined as the fraction of friends which have the same label as the user himself. So, if a user, John Smith, mentions his location to be Dallas, Texas and 15 out of his 20 friends are from Dallas, we say that the trustworthiness of John is 15/20=0.75. It is worthwhile to note here that users, who have lived all their lives at a single city, will have a large percentage of their friends from the same city and hence will have a high trustworthiness value. On the other hand, someone who has lived at several places will have a social graph consisting of people from all over and hence such a user should have little say when propagating labels to users with unknown locations. For locations without a location TW is zero.

Friendship Similarity amongst two people is a subjective term and we can implement it in several ways including number of common friends, semantic relatedness between the activities (verbs) of the two users collected from the messages posted by each one of them, etc. Based on the experiments we conducted, we adopted the number common friends as the optimum choice because of the low time complexity and better accuracy. We first calculate the common friends between users U_i and U_j and assign it as CF.

$$CF_{ij} = \text{Common\_Friends}\ (U_i, U_j) \quad (1)$$

The similarity between two users (SIM_{ij}) is a function of Trustworthiness and Friendship Similarity and can be represented as

$$SIM_{ij} = \alpha \times \text{Max}\{TW(U_i), TW(U_j)\} + (1-\alpha) \times CF_{ij} \quad (2)$$

where TW is the individual trustworthiness of the two users and α is an arbitrary constant whose value is between 0 and 1. Typically, α is chosen to be around 0.7 for trustworthiness measure to have the decisive say in the final similarity measure.

Next, we use Gaussian distribution function to calculate the weight W_{ij}. If the number of events is very large, then the Gaussian distribution function may be used to describe physical events. The Gaussian distribution is a continuous function which approximates the exact binomial distribution of events. Since the number of common friends can vary a lot, we use the Gaussian distribution. The Gaussian distribution shown is normalized so that the sum over all values of CF gives a probability of 1.

$$W_{ij} = e^{\frac{SIM^2}{2\sigma^2}} \quad (3)$$

But, there are certain special cases we need to take care of. Spammers and celebrities tend to be misleading while predicting the location of a user. The algorithm has zero tolerance towards spammers. A spammer is typically identified by the high ratio of the number of users he is following and the number of users following him back. We define the Spam Ratio (Ω_{ij}) of two users U_i and U_j as

$$\Omega_{ij} = \max\left\{\frac{\text{Following}(U_i)}{\text{Followers}(U_i)}, \frac{\text{Following}(U_j)}{\text{Followers}(U_j)}\right\} \quad (4)$$

And if Ω_{ij} is found to be greater than a threshold $N_{spammer}$, either of the two users is a spammer and set W_{ij} as 0, to isolate the spammer.

Finally, we would like to control the influence of celebrities in deciding the location of the user because of previously discussed problems. But, it is also important to note here that in certain cases the celebrities he is following are our best bet in guessing the user's location. If Followers(U_j) is greater than the threshold $N_{celebrity}$ than U_j is identified as a celebrity and the existing similarity it has with any user U_i gets abbreviated by a factor β, which is a function of number of followers of U_j and increases monotonically with the number of followers.

$$W_{ij} = \beta\ (U_j) \times W_{ij} \quad (5)$$

It is important to note here that the similarity weight matrix W is symmetric in nature for all i and j except if U_i is a celebrity. In such a case the weight W_{ij} will be much lesser then the calculated value, as mentioned before.

Another data structure that we define is the (n+1) × (n+1) diagonal matrix D, used for normalization

$$D_{ii} = \sum_{j=1}^{n+1} W_{ij} \quad (6)$$

And finally we define the Location Label matrix L of dimensions (n+1) × p, where p is the number of distinct location concepts. Initialize $L^{(0)}$ as

$$L_{ij}^{(0)} = \begin{cases} 1\ ; \text{if at j, } L_i = \text{concept class of } U_i \\ 0\ ; \text{otherwise} \end{cases} \quad (7)$$

Thus, initially, the bottom *u* rows consist of only zeroes. After all the matrices have been initialized we begin to iterate. Thus at step t of the iteration,

$$L^{(t)} = D^{-1}.\ W.\ L^{(t-1)} \quad (8)$$

$$L_l^{(t)} = L_l^{(t-1)} \qquad \text{// Clamp the labeled data} \quad (9)$$

At each step of the iteration, all unlabelled users receive a location contribution from their respective neighbors, proportional to the normalized similarity weight of the existing edge between the two. In this algorithm, we ensure

that the labeled vertices are clamped to the users and do not change. It can be easily shown here that as the number of iterations, t, becomes large, L converges to a definite value (α approaches zero).

$$\alpha = L^{(t)} - L^{(t-1)} = (D^{-1}W)^{(t)} L^{(0)} - (D^{-1}W)^{(t-1)} L^{(0)} \quad (10)$$

$$\alpha = (D^{-1}W)^{(t-1)} L^{(0)} [D^{-1}W - I] \quad (11)$$

Because the matrix $D^{-1}W$ is a square matrix each of whose rows consists of non-negative real numbers, with each row summing to 1, it follows that as $t \to \infty$, $(D^{-1}W)^{(t-1)} \to 0$, and hence L converges to a fixed value. The worst case running time of the algorithm is $O(n^3)$.

Now we discuss the impact of increasing the depth on accuracy and running time of the algorithm. By increasing the depth we include the friends of friends of the user also in our set of data points. The direct advantage of this is that we have more labeled data points in our set thereby having a positive impact on the accuracy. Next, inclusion of more data points (users) leads to discovery of implicit 'friendship' relationships between users that may not be specified otherwise. The only disadvantage that is associated with increasing the depth is the increase in the running time of the algorithm.

E. Agglomerative Hierarchical Clustering

The label propagation algorithm returns a vector of size p, each corresponding to a particular location concept propagated to it by its neighbors. At this point we introduce something called the Location Confidence Threshold (LCT). The idea behind LCT is to ensure that when the algorithm reports the possible locations, it does so with some minimum level of confidence.

$$\text{LCT}(u, \text{maxDepth}) = 1 - \beta(u)^{\text{maxDepth}} \quad (12)$$

As evident, the LCT increases with the increasing value of *maxDepth* that we specify as an input, because of the addition of more labeled data. β is a constant whose value lies between 0 and 1 and depends on the social graph of the user. For example, higher the number of labeled immediate friends of the user, lower is the value of β.

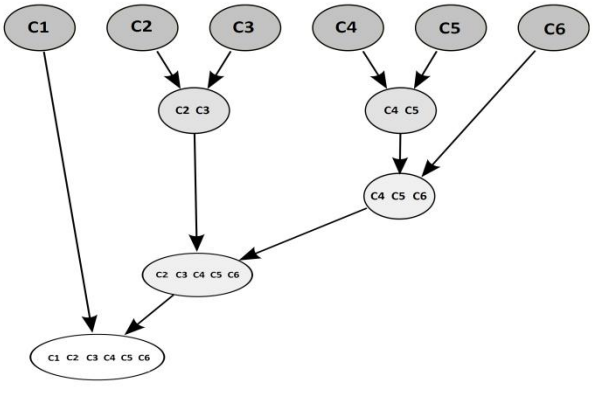


Figure 5. Illustration to show the agglomerative hierachial clustering

Till this point we have given little emphasis to the geospatial proximity of the different concepts. That is, we were treating the concepts purely as labels, with no mutual relatedness. Since the concepts are actual geographical cities, we agglomerate the closely located cities and suburbs in an effort to improve the confidence and thus the accuracy of the system. Fig 5 shows the agglomerative hierarchical clustering algorithm. Consider we have *p* concepts $C_1,\ldots,Cp$ each associated with its respective probability.

Initially, we have all concepts present individually as $\{C_1\}$, $\{C_2\},\ldots.$, $\{C_p\}$. If any concept has a value greater than the LCT, then the program returns that concept as the location and terminates. Otherwise, at the next step we construct a matrix in which the number in the *i*-th row *j*-th column is an objective function Θ of the distances and cumulative scores between the *i*-th and *j*-th concepts.

$$\Theta_{ij} = \frac{e^{\frac{S}{T}}}{\text{dist}(i,j)} \quad (13)$$

where $S = S_i + S_j$, the combined score of concept clusters C_i and C_j, dist(i,j) is the geographic distance between the two clusters and T is a constant with $0<T<1$.

At the first step of agglomeration, we combine two concepts with highest value of the objective function, Θ and check if the new concept cluster has a combined score greater than the LCT. If not, then we continue the process, constructing the matrix again, but this time some of the concepts are replaced by concept clusters. And we proceed to choose the two concepts clusters that have the maximum value of the objective function Θ. The mean geographic distance between a concept cluster A_i and a concept cluster B_j can be defined as

$$d_{AB} = \frac{1}{|A||B|} \sum_{x \varepsilon A} \sum_{x \varepsilon B} d(x, y) \quad (14)$$

Thus at each step of the agglomeration, we choose the two concept clusters with maximum value of the objective function Θ. If the score of the combined bag of concepts crosses the LCT, we return the bag of concepts as the possible location vector and terminate.

To understand how agglomerative cluster basically works, consider a scenario in which the label propagation returns an array of location concepts including (Los Angeles, 0.34), (Long Beach, 0.05), (Irvine, 0.17), and a lot of other concepts. Suppose the LCT for the algorithm to return a cluster of concepts is 0.5. Then, if we simply combine location concepts based on just proximity, then initially Los Angeles and Long Beach will get combined (Long Beach is closer to Los Angeles than Irvine) , but since their combined score is not sufficient, in the next iteration Irvine also gets added to the cluster. And the final cluster that is returned is {Los Angeles, Long Beach, Irvine} with a combined score of 0.56. On the other hand if we use agglomerative clustering with an objective function mentioned previously. In the first step Los Angeles and Irvine are combined to yield a location cluster of {Los Angeles, Irvine}, which has a combined score greater than the LCT and is hence returned as the output. Thus, by

using agglomerative clustering we end up being more specific by returning two concepts instead of three, at the loss of small confidence.

V. Location Mining From Text (LMT)

In this section we discuss the content based approach which uses the gazetteer for mining the location from the messages posted by the user.

To determine the location from mining the messages, we devise a score based identification and disambiguation method Location_Identification. Before running the actual algorithm, we perform pre-processing of data, which involves removal of all those words from the messages that are not references to geographic locations. For this, we use the CRF Tagger, which is an open source Part of Speech (POS) tagger for English with an accuracy of close to 97% and a tagging speed of 500 sentences per second [5]. The CRF tagger identifies all the proper nouns from the text and terms them as keywords $\{K_1, K_2,\ldots,K_n\}$. In the next step, the TIGER (**T**opologically **I**ntegrated **G**eographic **E**ncoding and **R**eferencing system) [6] dataset is searched for identifying the city names from amongst them. The TIGER dataset is an open source gazetteer consisting of topological records and shape files with coordinates for cities, counties, zip codes, street segments, etc., for the entire US.

Algorithm 2: Location_Identification (User_Messages)

Input: UM: *All Messages of User*
Output: Vector (C, S): Concepts and Score vector

1: for each keyword, K_i //Phase 1
2: for each $C_j \in K_i$ //C_j - Location Concept
3: for each $T_f \in C_j$
4: type = Type (T_f)
5: If (T_f occurs in UM) then $S_{Cj}=S_{Cj} + S_{type}$
6: for each K_i //Phase 2
7: for each $C_j \in K_i$
8: for $T_f \in C_j$,$T_s \in C_L$
9: If ($T_f = T_s$) and ($C_j \neq C_L$) then
10: type = Type (T_f)
11: $S_{Sj} = S_{Cj} + S_{type}$
12: return (C, S)

Algorithm 2. Gazetteer based approach to identify and disambiguate location

Algorithm 2 describes the gazetteer based algorithm. We search the TIGER gazetteer for the concepts $\{C_1, C_2\ldots C_n\}$ pertaining to each keyword. Now our goal for each keyword would be to pick out the right concept amongst the list, in other words disambiguate the location. For this, we use a weight based disambiguation method. In phase 1, we assign the weight to each concept based on the occurrence of its terms in the text. Specific concepts are assigned a greater weight as compared to the more general ones. In phase 2, we check for correlation between concepts, in which one concept subsumes the other. In that case the more specific concept gets the boosting from the more general concept. If a more specific concept C_i is part of another C_j then the weight of C_j is added to that of C_i.

For example, suppose city carries 15 points, state 10 and a country name carries 5 points. For the keyword "Dallas", consider the concept of {City} Dallas/ {State} Texas/ {Country} USA. The concept gets 15 points because Dallas is a city name, and it gets an additional 10 points if Texas is also mentioned in the text. In phase 2, we consider the relation between two keywords. Considering the previous example, if {Dallas, Texas} are the keywords appearing in the text, then amongst the various concepts listed for "Dallas" would be {City} Dallas/{State} Texas/{Country} USA and one of the concepts for "Texas" would be {State} Texas/ {Country} USA. Now, in phase 2 we check for such correlated concepts, in which one concept subsumes the other. In that case the more specific concept gets the boosting from the more general concept. Here, the above mentioned Texas concept boosts up the more specific Dallas concept. After the two phases our complete we re-order the concepts in descending order of their weights. Next, each concept is assigned a probability depending on their individual weights.

VI. Experiments

In this section, we evaluate the quality of the algorithms mentioned in the previous sections and describe how Tweecalization outperforms the other approaches.

A. Data

For the experiments, we randomly choose 1000 users from different countries and cities who explicitly mention their location and treat it as ground truth. It is important to note here, for uniformity, we ensure that each has at least 10 friends so that k-closest friends approach used in Tweethood can be applied. Fig. 6 shows the friend distribution for the dataset of 1000 users. We see that almost 45% of the users have 20 to 100 people as their friends.

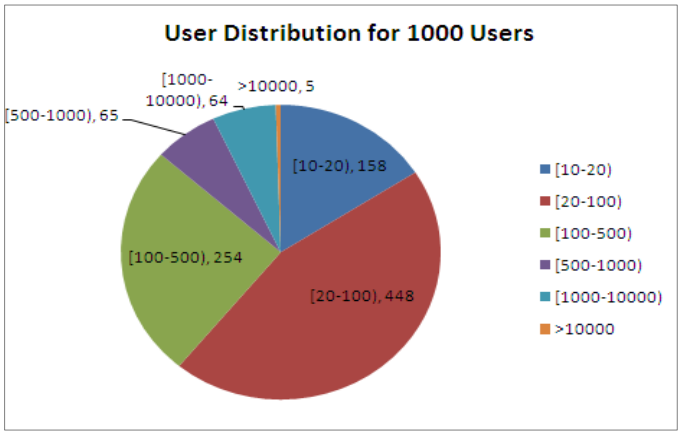


Figure 6. The user distribution for the data set.

Secondly, all processes are run offline i.e. we store all the relevant information about the user like location, friend count, friends ids, etc. on the local machine and then run the

algorithm. Hence the entire process is done offline, barring the geo-coding process, which is used to convert the explicitly mentioned locations to a standard format.

B. *Evaluation Method*

Our evaluation is designed with the following goals in mind. First, we aim to compare the accuracy of different approaches both at the city as well as the country level and show the effectiveness of Tweecalization in comparison to Tweethood and gazetteer based location mining technique. Second, we want to show the tradeoff between accuracy and time as a function of depth. Finally, we show how running time increases for difference algorithms with increasing depth. For all experiments we choose the gazetteer based approach discussed in the previous sections as the baseline.

C. *Experiment Type 1: Accuracy vs. Depth*

For these set of experiments, the Y axis represents the accuracy in percentage and the X axis shows the depth.

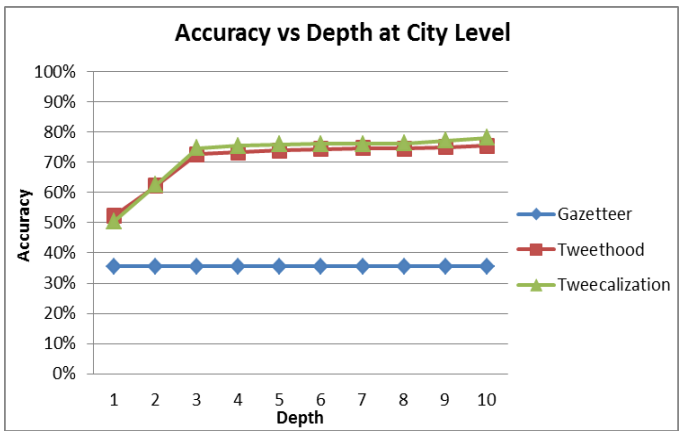


Figure 7. Accuracy vs Depth for various algorithms compared to Tweecalization

Fig. 7 shows the accuracy as a function of the depth for the city level location identification for the Agglomerative clustering on Label Propagation (Tweecalization), compared to Agglomerative clustering on Fuzzy *k*-closest Friends (Tweethood). We make two key observations, firstly, with the increasing depth the accuracy increases monotonically for both algorithms. As mentioned earlier, the reason for this is that, by increasing depth in Tweecalization, we ensure that we are adding more labeled data to our training set. Secondly, adding more data points leads to identification of new associations between nodes, that is, we can find new friendships that may not be otherwise specified by the user himself. On the other hand for Tweethood this is obvious because for *null* nodes, we are willing to go further and thus eventually find a label. The second key observation we make for this experiment is that, the accuracy doesn't increase significantly after depth=3 for both algorithms. On further analysis we find that the possibility of an implicit friendship existing between a user and node decreases with increasing depth of the graph and hence in such cases the increasing depth has little effect on the label propagation algorithm.

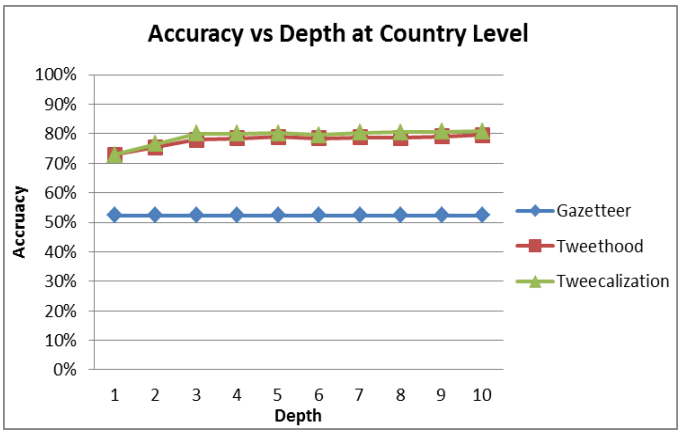


Figure 8. Accuracy vs. Depth at country level for Tweecalization

For depth less than 4, the accuracy value increases linearly with depth and is recorded to be 75.5% for Tweecalization at d=3. The baseline gazetteer based approach has a fairly low accuracy of 35.6% compared to our approach.

Next, we study the effect of increasing the depth on country level location identification for the two algorithms. Fig 8 shows the Accuracy vs. Depth comparison for different algorithms. The observations are very similar to the city level identification i.e. for depth greater than 4 the accuracy saturates. The accuracy for Tweecalization at depth=4 is reported to be 80.10% compared to 78.4% for Tweethood. And understandably, the accuracy for country level is higher than for the city level, because in certain cases the algorithm chooses the incorrect city, even though the country for both is the same.

D. *Experiment Type 2: Time Complexity*

For these set of experiments, the Y axis represents the time in seconds for various algorithms and the X axis shows the depth.

Fig. 9 shows the average running time for various algorithms for determination of the location of a single user as a function of depth. The key observations to make here are that for Tweethood, the time increases exponentially with increasing depth. Tweecalization, on the other hand, shows much better scalability because of a running time that's cubic in the size of friends. The increase in running time for Tweecalization is so insignificant in comparison to Tweethood that it appears as a straight line close to the X axis. At depth=4 the average running time recorded for Tweethood was 258.19 sec as compared to 0.624 sec for Tweecalization. The average running time for the content based approach is found to be 286.23 seconds. But for depth less than 4 both Tweethood and Tweecalization outperform the traditional gazetteer based location mining technique. This highlights the major contribution of Tweecalization, which is increased scalability with increasing depth for higher accuracy.

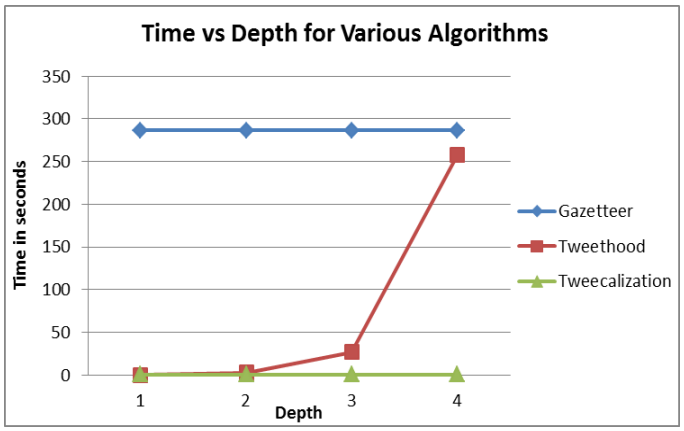


Figure 9. Time vs Depth for various algorithms compared to Tweecalization

VII. CONCLUSION AND FUTURE WORK

We present the development of a system that predicts the location of the user based on the locations of his friends. We choose the values d=4 for Tweecalization because of its optimal combination of accuracy and time complexity. Our experiments show that we are able to correctly identify the location of the user at the city level with an accuracy of 75.5% after using agglomerative clustering. The accuracy for country level identification is reported to be as high as 80.10%.

The system performs better than the traditional gazetteer based approach and Tweethood, in respect to both time and accuracy and is thus suited for the real-time applications. Even though IP address of a user can predict the location with most accuracy, only Twitter has the IP address of the users. On the contrary Tweecalization provides a simple yet effective algorithm that can be used to identify the location of user with a public profile.

The accuracy can be improved further by combining content based approach and Tweecalization. Even though time complexity is currently the biggest concern, it can be tackled by using the distributed computing or cloud computing framework. By doing this, we can geo-tag users on the fly and build real time web applications.

ACKNOWLEDGMENT

This material is based upon work supported by The Air Force Office of Scientific Research under Award No. FA-9550-09-1-0468. We thank Dr. Robert Herklotz for his support.

REFERENCES

[1] ABI Research. [Online].Available: http://www.abiresearch.com [Accessed: May. 1, 2010].

[2] Socialnomics.net,Socialnomics – Social Media Blog. [Online]. Available: http://www.socialnomics.net [Accessed: May. 1, 2010].

[3] Time in Partnership with CNN,Iran Protests: Twitter, the Medium of the Movement. [Online]. Available: http://www.time.com/time/world/article/0,8599,1905125,00.html [Accessed: May. 1, 2010].

[4] S. Abrol, L. Khan, "Agglomerative Clustering on Fuzzy k-Closest Friends with Variable Depth for Location Mining", The Second IEEE International Conference on Social Computing (SocialCom2010), Aug 20-22, 2010 Minneapolis, Minnesota.

[5] CRF Tagger,[Online]. Available: http://sourceforge.net/projects/crftagger/ [Accessed: May. 1, 2010].

[6] TIGER gazetteer [Online].Available: http://www.census.gov/geo/www/tiger/ [Accessed: May. 1, 2010].

[7] H. Li, R.K. Sihari, C. Niu, and W. Li, "Location Normalization for Information Extraction", 19th International Conference on Computational Linguistics, Aug. 2002.

[8] E. Amitay, N. Har'El, R. Sivan, and A. Soffer, "Web-a-Where:geo-tagging web content," Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 273-280.

[9] A. Mehler, Y. Bao, X. Li, Y. Wang, and S. Skiena., "Spatial analysis of news sources," IEEE Transactions on Visualization and Computer Graphics, 12(5):765--772, 2006.

[10] J. Liu and L. Birnbaum, "Localsavvy: aggregating local points of view about news issues" LocWeb, pages 33--40, 2008.

[11] C. Sheng, W. Hsu, and M.-L. Lee. Discovering geographical specific interests from web click data. In LocWeb.

[12] Foursquare And Gowalla In A Dead Heat In The Location War. [Online] . Available: http://techcrunch.com/2010/03/14/foursquare-gowalla-location-war/. [Accesed: May 15 2012].

[13] Y. Bengio, O. Dellalleau, and N. L. Roux, "Label propagation and quadratic criterion," In O. Chapelle, B. Schlkopf and A. Zien (Eds.), Semi-supervised learning. MIT Press, 2006.

[14] S. Abrol, L. Khan, "TWinner: Understanding News Queries With Geo-Content Using Twitter", 6th Workshop On Geographic Information Retrieval (GIR'10) At Zurich, Switzerland.

[15] D. Liben-Nowell Novak, R. Kumar, P. Raghavan, and A. Tomkins, "*Geographic Routing in Social Networks*," Proceedings of the National Academy of Sciences (PNAS), 102(33):11623--11628, 2005.

[16] O. B. Chapelle, Schölkopf and A. Zien: *Semi-Supervised Learning*. MIT Press, Cambridge, MA (2006).

[17] M. D. Lieberman, H. Samet, J. Sankaranarayana and J. Sperling, "STEWARD: Architecture of a spatio-textual search engine," 15th ACM International Symposium on Geographic Information Systems, pages 186-193, Seattle, WA, November 2007.

[18] S. Cucerzan and D. Yarowsky, "Language independent NER using a unified model of internal and contextual

[19] G. Eriksson, K. Franz´en, F. Olsson, L. Asker, and P. Lid´en, "Exploiting syntax when detecting protein names in text," Workshop on Natural Language Processing in Biomedical Applications, 2002.

[20] J. Leidner, G. Sinclair, and B. Webber, "Grounding spatial named entities for information extraction and question answering," Workshop on the Analysis of Geographic References, Edmonton, Alberta, Canada, May 2003. NAACL-HLT.

[21] Y. Ravin and N. Wacholder, "Extracting names from natural-language text," Technical Report RC-20338, IBM Research Division, T.J.Watson, Yorktown Heights, NY, October 1997.

[22] D. A. Smith and G. Crane, "Disambiguating geographic names in a historical digital library," The 5th European Conference on Research and Advanced Technology or Digital Libraries (ECDL'01), Lecture Notes in Computer Science, Darmstadt, September 2001. Springer.

[23] L. Backstrom, E. Sun. and C. Marlow, "Find Me If You Can: Improving Geographical Prediction with Social and Spatial Proximity," in World Wide Web Conference (WWW '10), Raleigh, NC.