Where Is This Tweet From? Inferring Home Locations of Twitter Users

Jalal Mahmud, Jeffrey Nichols, Clemens Drews
IBM Research Almaden
650 Harry Rd, San Jose, CA 95120
{jumahmud, jwnichols, cdrews}@us.ibm.com

Abstract

We present a new algorithm for inferring the home locations of Twitter users at different granularities, such as city, state, or time zone, using the content of their tweets and their tweeting behavior. Unlike existing approaches, our algorithm uses an ensemble of statistical and heuristic classifiers to predict locations. We find that a hierarchical classification approach can improve prediction accuracy. Experimental evidence suggests that our algorithm works well in practice and outperforms the best existing algorithms for predicting the location of Twitter users.

Introduction

Recent years have seen a rapid growth in micro-blogging and the rise of popular micro-blogging services such as Twitter. This has spurred numerous research efforts to mine this data for various applications, such as event detection (Sakaki et al. 2010) and news recommendation (Phelan et al. 2009). Many such applications could benefit from information about the location of users, but unfortunately location information is currently very sparse. Less than 1% of tweets are geo-tagged[1] and information available from the location field in users' profiles is unreliable at best.

In this paper, we aim to overcome this location sparseness problem by developing algorithms to predict the home, or primary, locations of Twitter users from the content of their tweets and their tweeting behavior. Our goal is to predict location at the city-level, though we also examine the possibility of predicting at other levels of granularity, such as state and time zone. The benefit of developing these algorithms is two-fold. First, the output can be used to create location-based visualizations and applications on top of Twitter. Second, our examinations of the discriminative features used by our algorithms suggest strategies for users to employ if they wish to micro-blog publically but not inadvertently reveal their location.

[1] http://thenextweb.com/2010/01/15/Twitter-geofail-023-tweets-geotagged/

Our research is motivated by previous work on location inference from tweets by Cheng et al. (2010), Eisenstein et al. (2010), and Hecht et al. (2011). Of these, only Cheng et al. attempts to predict the location of users at the city-level. Their result, which is the best of the three, is able to predict a user's city within 100 miles with 51% accuracy. We improve on that result in this work.

We make the following contributions:

- An algorithm for predicting locations of Twitter users from tweet contents, tweeting behavior (volume of tweets per time unit), and external location knowledge (e.g., dictionary containing names of cities and states). Our algorithm uses an ensemble of several classifiers.
- An algorithm for predicting locations hierarchically using time zone or state as the first level and city at the second level.
- An evaluation demonstrating that our algorithm outperforms the best existing algorithms for location prediction from tweets.

Dataset

From July 2011 to Aug 2011, we collected tweets from the top 100 cities in US by population[2]. First, we obtained a bounding box in terms of latitude and longitude for each city using Google's geo-coding API[3]. We recorded tweets using the geo-tag filter option of Twitter's streaming API[4] for each of those bounding boxes until we received tweets from 100 unique users in each location. The city corresponding to the bounding box where the user was discovered was assumed to be the ground truth home location for that user. We then invoked the Twitter REST API[5] to collect each user's 200 most recent tweets (less if that user had fewer than 200 total tweets). Some users were discovered to have private profiles and we eliminated them from our dataset. Our final data set contains 1,524,522 tweets

[2] http://en.wikipedia.org/wiki/List_of_United_States_cities_by_population
[3] http://code.google.com/apis/maps/documentation/geocoding/
[4] http://dev.twitter.com/pages/streaming_api
[5] http://dev.twitter.com/docs/api

generated by 9551 users. 100599 tweets (6.6%) were generated by Foursquare and contained URLs that could be accessed to retrieve exact location descriptions. 289650 tweets (19%) contained references to cities or states mentioned in the USGS gazetteer[6]. However, this number also includes ambiguous matches (e.g., the word "black" being matched as a town in Alabama) and the Foursquare tweets which also often contain references to cities or states. We divided the entire dataset into training (90%) and testing (10%) for 10-fold cross-validation.

Location Classification Approaches

Here we describe each of our location classifiers in detail.

Content-based Statistical Classifiers

We use three statistical location classifiers that are each trained from different terms extracted from S, the set of all users' tweets. The classifiers and their associated terms are:

- **Words:** all words contained within S
- **Hashtags:** all hashtags contained within S
- **Place Names:** all city and state location names within S, as identified via a geographical gazetteer

These classifiers can be created for any level of location granularity for which we have ground truth. Each user in our training dataset corresponds to a training example, where features are derived from his or her tweet contents. The output is a trained model with the number of classes equal to the total number of locations of that granularity in our training dataset (e.g., total number of cities). All of these classifiers use the same approaches for feature selection, training, and classification, which are described below.

Feature Selection

First, we tokenize all tweets in the training dataset, which removes punctuation and other whitespace. All URLs and most tokens containing special characters are then removed, except for tokens that represent hashtags and start with # (e.g., the token #Portland). Once the tokens have been extracted, different processes are used to extract terms for each classifier.

For the Words classifier, we use as terms all non-stop word tokens that are identified as nouns by a part-of-speech tagger. Stop words are defined by a standard list of 319 stop words, and parts of speech are classified using Open NLP (http://opennlp.sourceforge.net). We do not use adjectives, verbs, prepositions, etc. because they are often generic and may not discriminate among locations.

For the Hashtags classifier, we use as terms all tokens that start with the # symbol.

For the Place Names classifier, we generate a set of terms that appear in the tweets and match names of US cities and states from the USGS gazetteer. Not all city or state names are a single word, so we first generate bi- and tri-grams from the ordered list of tokens. We then compare all uni-, bi-, and tri-grams to the list of city and state names. Any matching names are used as terms.

Once we have the set of terms for a particular classifier, it is helpful to identify terms that are particularly discriminative (or "local") for a location (also discussed by Cheng et al. (2010)). For example, the term "Red Sox," is local to the city "Boston". We use several heuristics to select *local terms,* which become features for our statistical models. First, we compute the frequency of the selected terms for each location and the number of people in that location who have used them in their tweets. We keep the terms that are present in the tweets of at least K% people in that location, where K is an empirically selected parameter. We experimented with different values and selected K=5. Next, we compute the average and maximum conditional probabilities of locations for each term, and test if the difference between these probabilities is above a threshold, T_{diff} (empirically selected as 0.1). If this test is successful, we then further test if the maximum conditional probability is above a threshold, T_{max} (empirically selected as 0.5). This ensures that the term has high bias towards a particular location. Applying these heuristics gives us localized terms and eliminates many terms with uniform distribution across all locations.

Training and Classification

Once the features (i.e. local terms from the previous step) are selected for each classifier, we build statistical models using standard machine learning approaches. We have tried a number of classifiers from WEKA such as Naïve Bayes, Naïve Bayes Multimonial, SMO (an SVM implementation), J48, PART and Random Forest. We found that Naïve Bayes Multimonial, SMO and J48 classifiers produced reasonable classification results for our dataset, and we empirically selected Naïve Bayes Multimonial.

Content-based Heuristics Classifiers

We have also built two heuristic classifiers that predict users' locations at different granularities.

The *local place* heuristic classifier is specific to classifying city or state-level location. The heuristic is that a user would mention his or her home city or state in tweets more often than any other cities or states. For every city or state in our training corpus, we compute the frequency of its occurrences in user's tweets and use this as the matching score of that user with that city or state. The city or state with the highest matching score is predicted as the location classification for that user.

The *visit history* heuristic classifier is applicable to location classification at all granularities. The heuristic is that a user would visit places in his home location more often than places in other locations. In order to retrieve a user's visit history, we look for URLs generated by the Four-

[6] http://www.census.gov/geo/www/gazetteer/places2k.html

square location check-in service in their tweets, retrieve venue location information from those URLs (e.g., city, state) using the Foursquare API, and build a frequency-based statistic for the visited locations at the desired level of granularity. Links that cannot be resolved to a venue are discarded. The location with the highest frequency is returned as the location classification for the user.

Behavior-based Time Zone Classifiers

We have constructed a time zone location classifier based on the time at which users send their tweets. To construct the classifier, we first divide the day into equal-sized time slots of a pre-specified duration. Each time slot represents a feature-dimension for the classifier. We have tried different sizes for time slots, e.g., 60, 30, 15, 5, and 1 minutes. We empirically chose 1 minute duration time slots for our classifier. For each time slot, we count the number of tweets sent during that time slot for each user in our training set. Since total tweet frequency in a day varies across users, we normalize the number of tweets in a time slot for a user by the total number of tweets for that user. Different times of day are more discriminative, and we capture this variation by weighting the feature values of each time-slot using the standard deviation for that time slot. To train the classifier, we use the Naïve Bayes classifier from WEKA.

Ensemble of Location Classifiers

We also create ensemble of our classifiers to improve accuracy. In this work, we have used a dynamically weighted ensemble method (Jiménez et al. 1998) for creating an ensemble of statistical and heuristic classifiers. We have also tried majority voting (Rokach et al. 2010) and boosting (Freund et al. 1996) but they did not yield a better result.

Here we will introduce a metric, *Classification Strength,* which we use in our dynamically weighted ensemble implementation. Let T denote the set of terms from user's tweets that would be considered for classification using a particular classifier. For statistical classifiers, the *matching location distribution* is the set of locations in our trained model containing terms from T. For the local-place classifier, this distribution contains locations from our dataset that match content in the user's tweets. For the visit-history classifier, this distribution contains locations from the user's visit history that appear in our dataset. The *Classification Strength* for a user is the inverse of the number of matching locations in the matching location distribution. Classification strength of a classifier for a particular instance expresses discriminative ability of that classifier for classifying that instance. For our implementation of dynamically weighted ensemble, classification strength of a classifier for a particular instance is used as the weight of that classifier in the ensemble (for classifying that instance) and the location with the highest rank by weighted linear combination is returned as the result. Since classification strength cannot be computed for the behavior-based time zone classifier, we use the probability value or the confidence value associated with that classifier as its weight.

Hierarchical Ensemble of Classifiers

We have also developed location predictors using a two level hierarchy. When time zone is the first level of hierarchy, we first trained an ensemble time zone classifier from our training corpus using all content-based classifiers and the behavior-based classifier. City classifiers were trained for each time zone, where each classifier was limited to predicting only the cities in its time zone and trained with only examples from that time zone. When state is used as the first level of hierarchy, the ensemble state classifier contains only our content-based classifiers and city classifiers are built only for states containing more than one city.

Experiments

We conducted many experiments to evaluate different aspects of our algorithms. To determine the accuracy of our algorithm, we use the standard accuracy metric Recall (R). Let the total number of users in our test set be n. When this is given to our location predictor, only n_1 predictions are correct. Hence, we define recall (R) as n_1/n.

Word	Hashtag	Place name	Local-place	Visit-history
0.34	0.17	0.54	0.5	0.13

Table 1. Recall Comparison among different classifiers

Table 1 shows the comparative performance of the individual location classifiers. The Place Name statistical classifier gives the best recall performance. The high recall of the place name-based classifier may be explained by the fact that many users send tweets containing names of places (cities and states in our system), and those place names tend to have bias towards users' home cities. The low recall of visit-history classifier is due to the sparseness of needed Foursquare URLs in our dataset (only 6.6% of those in our dataset).

	City	State	Time zone
Recall	0.58	0.66	0.78

Table 2. Location Prediction Performance using Ensemble

Table 2 shows the performance of our ensemble classifier for predicting location at the level of city, time zone or state. Performance is generally higher for classifiers that discriminate between fewer classes.

Table 3 shows the performance of different hierarchical classification approaches for city location estimation. Note

that the performance all hierarchical classifiers is superior to the single level ensemble for city prediction.

	Time-zone hierarchy	State hierarchy
Recall	0.64	0.59

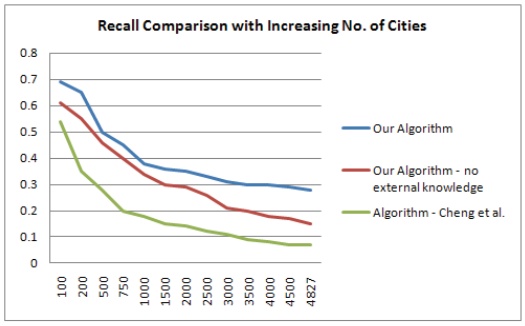
Table 3. Performance of Hierarchical City Location Estimator

We also compared the performance of city-level classification by directly comparing with the algorithm of Cheng et al. (2010) (which achieved best accuracies for city prediction) using their dataset, which we received from the authors. For comparison, we implemented their algorithm and used multiple accuracy metrics: recall for exact location match and the distance-based relaxed accuracy metric used by Cheng et al. (2010). Since Cheng et al. did not use any external knowledge (such as a dictionary), we also compared the performance when our algorithm did not use any external knowledge (i.e. we removed place-name and visit-history classifier from the ensemble). Figure 1 shows that our algorithm significantly outperforms their algorithm in all cases (two tailed p value < 0.05, 95% confidence interval).

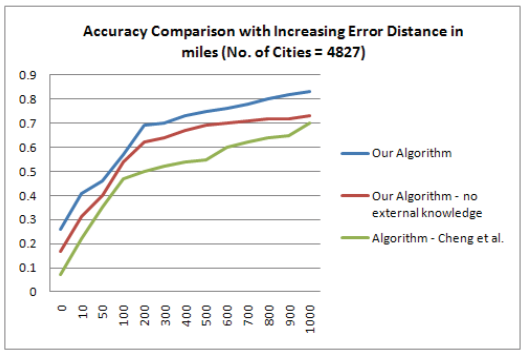
A key question is what impact the availability of explicit location references, such as place name mentions and the presence of Foursquare URLs, has on classification performance? In other words, how effectively can users mask their location if they never mention place names? To test this, we computed the performance of just the word and hashtag statistical classifiers in an ensemble. We found that locations are still predictable, but accuracy was reduced (city level location predictor was able to predict with 0.34 recall without hierarchy and 0.4 recall with time-zone hierarchy). This suggests that users may be able to partially mask their location by being careful not to mention location names in their tweets.

Conclusion

In this paper, we have presented a hierarchical ensemble algorithm for predicting the home location of Twitter users at different granularities. Our algorithm uses a variety of different features, leverages domain knowledge and combines statistical and heuristics classifications. Experimental performance demonstrates that our algorithm achieves higher performance than any previous algorithms for predicting locations of Twitter users. We are interested to explore predicting location at even smaller granularities, such as the neighborhood level. Along the same line, it would be interesting to explore the possibilities of predicting locations of each message. We also hope to integrate our algorithm into various applications to explore its usefulness in real world deployments.



(a)



(b)

Figure 1. Comparison of our hierarchical city location predictor (with time zone hierarchy) with the best available algorithm

References

Cheng, Z., Caverlee, J. and Lee, K. 2010. You Are Where You Tweet: A Content Based Approach to Geo locating Twitter Users, In *Proc.* of CIKM.

Eisenstein, J.. O'Connor, B., Smith, N.A. and Xing, E.P. 2010. A Latent Variable Model for Geographic Lexical Variation, In Proc. of EMNLP.

Freund.Y., and Shapire. R.E., 1996, Experiments with a new boosting algorithm, In Proc. of the ICML.

Jiménez, D. 1998, Dynamically weighted ensemble neural networks for classification, In Proc. of International Joint Conf. on Neural Networks.

Hecht, B., Hong, L., Suh, B. and Chi, Ed H. 2011. Tweets from Justin Bieber's Heart: The Dynamics of the "Location" Field in User Profiles, In *Proc.* of CHI.

Phelan, O., McCarthy, K., and Smyth, B. 2009. Using Twitter to Recommend Real time Topical News. In Proc. of RecSys.

Rokach, L. Pattern Classification using Ensemble Methods. 2010. Series in Machine Perception and Artificial Intelligence Vol. 75. World Scientific Publishing, ISBN:981 4271 063.

Sakaki, T., Okazaki, M., and Matsuo, Y. 2010. Earthquake shakes Twitter users: real time event detection by social sensors. In Proc. of WWW.