Tweet Based Geolocation

Aashish Mahajan
1207662725
amahaja8@asu.edu

Aravinda Kumar Reddy Yempada
1208601637
ayempada@asu.edu

Kranthi Sai Davuluri
1208677310
kdavulur@asu.edu

Abstract

Geographical location is very essential to understand community behavior. One particular use case is to recommend product to specific group of users. We tried to investigate and improve on the task of tweet based geolocation prediction on data sets generated using twitter API. It has been recently seen that geographical properties i.e. location, place, reference for a particular word have begun to be exploited for the purpose of geolocating documents based solely on the content, often in the context of social media and online content. One common approach that could strike our mind for this task is that geolocating texts is rooted in information retrieval. It has been seen in previous studies that location indicative words like, gazetteer terms, dialectal words in a text are indicative of tweets origin location. We have a set of training data that is labeled with location coordinate i.e. latitude/longitude coordinates. Several approaches were proposed to aggregate the data into particular location or cities. One common approach is to use a grid that could be overlaid on Earth and then mapping tweet to specific cell in grid. Given a new tweet, its location is chosen by using several techniques like tf-idfs, Naïve Bayes classifier etc. In this paper, we extend this model by using MiniBatchKmeans, to aggregate the data into clusters. In this paper, we evaluate a range of feature selection methods to obtain "location indicative words". We then evaluate the impact of hashtags and user references in the tweet, non-geotagged tweets, language, and user-declared metadata on geolocation prediction.

1 Introduction

In recent times we have seen a huge increase in non-geotagged data from social media online platforms like Twitter, Facebook, and Tumblr. As per our study around 4.5 to 5 % of Geo Tagged data is available. So, this implies that our project of predicting the location of the tweet from the user generated tweets is useful. Our first problem is to understand the important features from the user generated tweet.

Geographical location of a user is vital to geospatial applications like local search and event detection. It is really important to know a tweet location in order to accomplish these tasks effectively. For an instance, if such services are put to use for a disaster response task, it must know where to direct resources in order to effectively coordinate aid. Advertisers could benefit from tailoring advertisements to a user's location. Similarly, search results localization hinges on knowledge of a user's location. Other applications of this task are, tracing the sources of historical documents; location attribution while summarizing large documents; tailoring of ads while browsing.

Although many social media services allow a user to declare their location, such metadata is known to be unstructured and ad hoc. In this project on text-based tweet geolocation prediction, we have developed mechanism to predict the latitude and longitude based on Geo tagged user tweet.

1.1 Location Indicative words

To simplify the statement, we could say that clues to the geographic location of a document may come from a variety of word features, e.g. toponyms(*Arizona*), geographic features(*Desert/Semi-arid*), culturally local features (*soccer*), and stylistic or dialectical differences (*message vs. mssg vs. msg*). These are called 'Location Indicative words" [2] or Unique words.

1.2 Hashtags

Users often use hashtags in the social media exchange on Twitter. Let it be in a form of status, post, share, pool, trending issues, media exchange etc. For example,

#breaking neenah police respond to hostage situation at eagle nation cycles, shots fired http://bit.ly/1ofive0

Here clearly we could identify the hashtag **#breaking** used to show the critical update about news that is just aired. An update about the incident. It gives the world an idea about what people are talking about but for a social media scientist it would have a longitude and latitude associated with the hashtag. That could be simplified and located using those. For a scenario of super bowl game in phoenix if people are doing #superbowl #phoenix #game a collaboration of hashtag but they would hold a longitude and location. It would help us identify and localize the location of event/incident. These locations could be used for various purpose i.e. could be used to direct the vendor and sales in that particular region, businesses could see and monitor what consumers are talking about in their areas, and they can capitalize on those trends by using the appropriate hashtags or keywords to enter into the conversation. This could be used to provide and create special custom pricing offer based on the features that go in align with the hashtag which gives us the information for longitude and latitude that make geolocation easy.

2 Related Work

There has been tremendous amount of work in this field. Let it be for research purpose or for business purpose. For example, in the spatial data mining community, geographical references (e.g., gazetteer terms) in text have also been exploited to infer geolocation. Intuitively, if a place is frequently mentioned by a user in their tweets, they are likely tweeting from that region. Methods building on this intuition range from naive gazetteer matching and rule-based approaches to machine learning-based methods.
Also according to recent work and studies and research, many robust machine learning methods have been thought over and applied to geolocation system with an aim to improve the efficiency and performance, with the primary approach being to estimate locations based on the textual content of tweets. For instance, Bo Han et al. [2] exploit words known to be primarily used in particular regions, along with smoothing techniques, to improve a simple generative geolocation model when applied to data from the continental United States.

An application named geofeedia which is used to geo-locate social media postings in real-time, is a social media monitoring tool that organizations can use to search by location in real-time.

Users enter an address or draw a custom perimeter on a digital map around the area they want to search, and geofeedia pulls up all the content that been posted on social media by users within that area. It covers a number of social media platforms, including Twitter, Instagram, Facebook, YouTube, and Flickr. The organizations can see what consumers are talking about, trending topics in their areas, and then they can capitalize on those trends by using the appropriate hashtags or keywords to enter into the conversation and capture a market that has not been tamed since long.

There are several papers which predict the location based on the user profile, his friendship network and his daily activities. Particularly, the research paper by Jurgens et al. [5] describes several approaches done using the user details and their social network. As for our project we found that only [2] has work related to what we are doing in this project.

3 Proposal Method

3.1 Preprocessing Step

Preprocessing is an important step in our project. Tweets usually contain jargon and non-alphanumeric characters and also the location inference words are very sparse. As the dataset size is huge we have to remove the tweets with no useful location information.

We have extracted the tweets and the location information from the given json files and stored them in a csv file. As a first step, we have removed all the stop words from the data. The set of stop words are taken from NLTK package. We have removed all the common words as they do not provide any advantage to the model. We have used the common words list from [3]. We then removed all the non-alphanumeric characters from the data. We have evaluated our model on two sets of data, one with hashtags and other without hashtags. For the latter dataset, we have removed all the hashtags from the data.

After the initial preprocessing in python, we have loaded our data into a PostgreSQL database for further analysis. We used custom queries to find the frequencies of each word and removed all the words with frequency less than 10 in the entire dataset. We assumed that these rare words would not have any impact in classifying the data on such a huge dataset. Also, removing these words would help us reduce the amount of time needed to train the model.

The initial dataset has 14528735 tweets and the final dataset after preprocessing has 11856488. The tweets example can be seen in Table 1.

3.2 Clustering

Using the idea from [2] (use a grid that could be overlaid on Earth and then mapping tweet to specific cell in grid), we have grouped the tweets into 2000 clusters based on the latitude and longitude. Initially, we started working on grids, where each tweet is mapped to fixed location based on grid, but this method seems unreliable, so we used K means clustering instead. Particularly we have used MiniBatchKmeans from scikit due to its better performance. This method of K Means is very efficient and converges in few iterations. This reduced our work from

30 hours without converging to 30 minutes with converging and still finding optimal clusters. We ran the algorithm several times to find the best K. We have evaluated K based on the average error of each cluster point from the cluster centroid. Cluster centroid is very important for us since our prediction would eventually be cluster centroid.

Tweet
cek
bence dee ben cok begendim #sandy
golazo
niall blue
hallo hallo wwf
nasl olursun ya
featuring effing
masjid 4ampong
cierto pase imbeciles pero creo que deberian poner peligro sus vidas por
cobro martes jajaajjaja #perfect #needit #wantit #gottahaveit

Table 1

After building the model, we used cPickle to store and load the model. Then for each tweet, we appended cluster classes and cluster centroid locations. We also used the same technique to the hashtags, where each hashtag is appended with cluster class and cluster centroid locations.

The data that comes out from this task is the input for our classifier and it look likes below.

Tweet	Original Latitude	Original Longitude	Class	Centroid Latitude	Centroid Longitude
cek	-6.26501	106.8813	1070	-6.284	106.842
bence dee ben cok begendim #sandy	38.72332	35.51189	484	38.697	35.549
golazo	43.24832	-2.93357	422	43.299	-2.883
niall blue	51.39518	0.0362	1281	51.512	0.003
hallo hallo wwf	-6.20484	106.8173	946	-6.192	106.822
nasl olursun ya	40.70165	29.88766	909	40.789	29.897
featuring effing	41.57627	-73.469	1697	41.386	-73.482
masjid kampung	1.310011	103.9276	81	1.34	103.93
cierto pase imbeciles pero creo que deberian poner peligro sus vidas por	27.86196	-15.4391	1834	27.853	-15.439
cobro martes jajaajjaja #perfect #needit #wantit #gottahaveit	19.38345	-99.1809	1154	19.418	-99.153

3.3 Classification

3.1.1 Feature Extraction

We have used the Vector Space Model using the bag of words model for our project since it is the best method for text classification. Given the text, we have built the term document matrix and used term frequency - Inverse Document Frequency (tf-idf) as the weight. The best terms of the model are evaluated using the chi-square statistic described in [2]. We have then normalized the weights. These important terms are taken as features for our model.

3.2.2 Naïve Bayes Classifier

We have used the Naïve Bayes classifier as our final classifier. Our experiments showed that it has better performance than other classifiers such as Decision Trees, SVM. We have used the MulitnomialNB class of scikit learn package to train the model and predict the test data. We used train_test_split to split the data into train data and test data randomly to avoid any bias. We have also used 5-fold cross validation to check if the results are consistent and the model is not overfitting.

3.4 Datasets Mechanism

We have built three sets of data, extracted features and ran the Naïve Bayes classifier on them.

3.4.1 Unique Words

In this method, we have used only the words that we got after the preprocessing. We have not used the hashtags.

3.4.2 Unique Words + Hashtags

In this method, we have used the words and the hashtags that we got after the preprocessing. The idea is to evaluate the importance of hashtags in predicting the location of user.

3.4.3 Inverse Cluster Frequency

We have taken a cue from [2] and tried to build features similar to Inverse City Frequency in the paper. As we do not have user city name in the training data. We have used cluster as analogous to user and built huge documents consisting of all texts of each user in that cluster. This resulted in 2000 such documents since we have 2000 clusters. We have applied the bag of words model on this documents and called the weights as Inverse Cluster Frequency weights.

4 Results

We have evaluated the model on three datasets described above. After predicting the class labels we have used the centroids of that class as the location of the test example. We have measured the accuracy based on the deviation of latitude and longitude of the predicted value from the true value. We have given the accuracy on different values of Latitude and Longitude deviations. For example, **Latitude=1.5^0, Longitude=3^0** in the below table means that the predicted value is at most 1.5^0(~69 miles) away in north-south direction and at most 3^0 (0-69miles) away in east-west direction.

We have evaluated on different sizes of tweets such as 0.5 Million, 1 Million and 1.5 Million. The processing power has limited the size of our dataset and we could evaluate our method only on 1.5 million tweets. The reason is due to the nature of text in tweets due to which the vocabulary is very high. Considering the bag of words model, the resuting term document matrix is humungous and did not fit on our memory for bigger datasets. However, we have got some concrete findings based on the below evaluations.

4.1 Unique Words

Below are the results of unique words dataset, where each value is accuracy in %.

	5,00,000	**1 Million**	**1.5 Million**
Latitude=1.5^0, Longitude=3^0	22.1813	22.8476	23.0964
Latitude=1.5^0, Longitude=5^0	24.7113	25.4306	25.678
Latitude=2.5^0, Longitude=3^0	25.0753	25.806	26.0908
Latitude=5^0, Longitude=7^0	35.91	36.709	37.074
Latitude=7.5^0, Longitude=10^0	40.452	41.2266	41.624

4.2 Unique Words + Hashtags

Below are the results of unique words + hashtags dataset, where each value is accuracy in %.

	5,00,000	**1 Million**	**1.5 Million**
Latitude=1.5^0, Longitude=3^0	24.448	25.86	26.538
Latitude=1.5^0, Longitude=5^0	27.0686	28.5526	29.2557

Latitude=2.5^0, Longitude=3^0	27.4006	28.926	29.59866
Latitude=5^0, Longitude=7^0	38.37533	40.109	40.8508
Latitude=7.5^0, Longitude=10^0	42.8113	44.745	45.454

4.3 Inverse Cluster Frequency

Below are the results of Inverse Cluster Frequency dataset, where each value is accuracy in %.

	5,00,000	**1 Million**	**1.5 Million**
Latitude=1.5^0, Longitude=3^0	18.486	19.11366	19.2442
Latitude=1.5^0, Longitude=5^0	20.9633	21.61733	21.6022
Latitude=2.5^0, Longitude=3^0	21.28	21.9753	22.0606
Latitude=5^0, Longitude=7^0	34.5993	35.2466	35.10008
Latitude=7.5^0, Longitude=10^0	42.6793	43.334	43.2885

4.4 Cross Validation results

We have run 5-fold cross validation and measured the accuracy for deviations '**Latitude=5^0, Longitude=7^0**' using 0.5 million and 1 million tweets.

5 fold cross validation results in %					
500000	36.113646	36.266757	36.336997	36.330686	36.478095
1 Million	37.0717	37.063077	36.963782	36.994153	37.017848

5 Findings

- Based on the results we can say that the accuracy is dependent on the size of the dataset. It clearly increases with the number of tweets.

- The results of 'Unique words + Hashtags' are at least 2% better than the results of just 'Unique words'. Thus, we can conclude that hashtags has some importance in predicting the location of the user.
- The accuracy of our model is low but the cross-validation scores are same across all the folds which means that our model is predicting x% of tweets at any time. Low false predictions.
- Inverse Cluster Frequency did not give better results than other two approaches which shows that users in a cluster need not be same.
- Predicting locations from just the twitter data is difficult. Having user data, his social activities would have helped us to build better models.

6 Future Work

Our approach relies very much on the clustering accuracy. We have tried several iterations to find the best K for the problem and this has taken us lot of time since we have used K-means clustering and MiniBatchKmeans. In the future, we can improve this by using k-d tree search optimization. Also, we can further work on extracting new features and evaluating them on our classifier.

Reference:

[1] http://dl.acm.org/citation.cfm?id=2391120
[2] Text-Based Twitter User Geolocation Prediction. Bo Han, Paul Cook, Timothy Baldwinhttp://dl.acm.org/citation.cfm?id=2391120
[3] https://github.com/pkLazer/password_rank/blob/master/4000-most-common-english-words-csv.csv
[4] Tuten, 2008; N´unez-Red´o, D´ıaz, Gil, Gonz´alez, & Huerta, 2011; Yin, Lampert, Cameron, Robinson, & Power, 2012)
[5] Geolocation Prediction in Twitter Using Social Networks: A Critical Analysis and Review of Current Practice. David Jurgens, Tyler Finnethy, James McCorriston, Yi Tian Xu, Derek Ruths