EBUG: Entity-Based User Geolocation on Twitter

Ayşe Irmak Erçevik
TOBB University of Economics and Technology

Mehmet Deniz Türkmen
TOBB University of Economics and Technology

1 INTRODUCTION

Currently social media platforms host excessive number of users. People do not only read what other people writes, but also provide content to these platforms. Although people's main reason is socializing, they do not realize they give important clues about themselves with their online activity. As a results, social media has become the main data source for the researchers who are interested in the social structures.

One of the intriguing data that can be extracted from social media data is location of users. For instance, some platforms allow users to share their locations. However, whether provided location information is true is not confirmed by the platforms and most users do not even prefer to share their locations. As a result, extracting user locations from the textual user content (e.g., tweets) has been one of the common approaches.

In this study, we propose a geolocation method based on named entities that are present in user tweets. We try to predict the US states where Twitter users reside in. We present a base method, compare it with a common approach in literature, and also examine some design decisions about our method.

2 PROPOSED METHOD

2.1 Method Description

In this study, we propose leveraging named entities in text data to make an inference about home states of Twitter users. Our mechanism of user geolocation consists of 4 sequential main steps:

2.1.1 ***Named Entity Extraction:*** Named entities are basically words or word groups that point to some real-world objects, such as a person, organization, location, etc. Accordingly, named entities used in a tweet can provide clues about the location of the user. For instance, if a user tweets about the mayor of a city, it is highly possible that that user resides in that city. Or, if a user mentions an organization, let's say a school, that user probably studies in it. Moreover, sometimes users might directly mention the state they live in. From that point of view, given a set of tweets belonging to a user, named entities that are present in the tweets are detected using a named entity recognition model.

2.1.2 ***Entity Expansion:*** Since named entities are real-word objects, they are related to real-world places mostly. At the second step, we try to extract places related to entities. As in the previous example, if a user mentions the name of a mayor, we need to obtain which administrative division he/she is the mayor of. Similarly, if a detected entity is the name of a school, we need to find where the school is. From that point of view, each named entity detected at the previous step are queried through Wikipedia. Wikipedia provides a short description for each entity queried. These description texts contain places related to entities. For instance if you query 'Michelle Wu' , Wikipedia returns a description saying she is the mayor of Boston, Massachusetts.

2.1.3 ***Geoparsing:*** Geoparsing is defined as the process of retrieving place related expressions in a text data. For instance, let's think of a sentence like "Statue of liberty is amazing". When this sentence is fed to a geoparser, it detects "Statue of Liberty" since it expresses a location and returns detailed information for Statue of Liberty such as coordinates, city, state, country, etc.. In this step, we feed description texts obtained from Wikipedia to a geoparser model, and we store state information for each retrieved results.

2.1.4 ***Voting:*** After the geoparsing step, we have lots of candidate states extracted from Wikipedia descriptions. A final decision for the prediction result is made by a voting process that is carried out between the extracted state names. The state name having the majority of the votes is returned as the prediction result.

2.2 Distinction of Our Method

Text-based user geolocation is one of the hardest problems in social media mining, because textual data contains very few clues about the location of the author. Although neural networks are famous with their advanced feature extraction abilities, they have difficulty with finding location indicative words in a full text since these words are sparse. Accordingly, we don't use a classifier to extract features and make a prediction at the same time, instead we use a named entity recognizer to only extract named entities, since named entity recognition is a simpler NLP problem relatively and there are successful open-access named entity models already. Then, we use extracted named entities as features. In this way, we aim to increase prediction performance by splitting author geolocation problem into simpler and smaller parts.

Named entities don't have to be location related. Named entities are categorized based on the type of real-world objects they point to. In particular, person and organization entities are two common entity types. We propose a method called entity expansion to utilize all types of named entities and find related locations for each named entity.

A group of studies define a set of location indicative words (LIWs) based on the information gain of words in text data (Han et al. [2]). They use extracted LIWs to represent texts or users. Although that kind of approach is a solution to the feature extraction problem mentioned above, it is data dependent since it is a statistical method. On the other hand, our method does not depend the characteristic of the data, can be applied on any dataset.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

Table 1: Inference examples for 3 types of named entity

Named Entity	Entity Type	Wikipedia Description	Candidate State
UCLA	Organization	Public university in Los Angeles, California	California
Queens	Location	Borough in New York City	New York
Michelle Wu	Person	Mayor of Boston	Massachusetts

Lastly, our method differs from classification-based methods in being able not to produce any results, specifically when no named entity or candidate state is detected. That is, our method don't conclude anything if the given text data doesn't contain any clue about the author location. On the other hand, a classifier returns a prediction no matter what input data it is fed. This property will not make a difference in terms of accuracy, but it will lead to a higher precision score for our method.

2.3 Dataset

For this project we used Geo-Tagged Microblog Corpus dataset [3] and made changes to the dataset according to the problem definition.

2.3.1 ***Dataset Description.*** Geo-Tagged Microblog Corpus dataset contains 377,616 English Tweets from 9,475 users in the continental US over one week. Each message is tagged with a location information as (latitude, longitude) coordinate pair which is obtained via users' GPS-enabled devices. Even though it is observed that users tweet from different coordinates, most of the messages from a single person tend to come from nearby locations. The dataset covers most of the continental US, however, in line with our problem, we were interested in each of the 48 contiguous United States and District of Columbia.

2.3.2 ***Preprocess.*** For modeling purposes, we wanted to use only a single geographic location for each user. Therefore, at first, we derived state name from the geolocation (coordinate pair) tag of tweets by using Nominatim API[1] . Then, for each user, we determined the location where the user tweeted most frequently and removed tweets which were sent by same user but from different locations. Finally, we removed redundant user information from the dataset, leaving only the data of users tweeting from US states For convenience in our models, we grouped all data on the user information by using the "Group by" operation

2.3.3 ***Processed Dataset.*** After preprocess, final dataset consists of 9172 rows and 3 columns. Each row displays (1) User-id, (2) State Name (Name of the state where the user frequently tweeted) and (3) List of Tweets (List of tweets that user tweeted from the state) information. For our models, we split the final dataset as %75 train and %25 test.

2.4 Implementation

2.4.1 ***Named Entity Recognition (NER).*** : We utilize dslim/bert-base-NER[2], pre-trained NER model to automatically obtain named entity occurences. This model categorizes entities into 4 types, which are "location", "organization", "person" and "miscellaneous".

2.4.2 ***Entity Expansion.*** : Wikipedia provides a REST API service[3], which allows people to send query requests and returns results from Wikipedia database. During entity expansion, we send a summary request for the entered

2.4.3 ***Geo-parsing.*** : We employ "mordecai" Python library[4] for geoparsing.

2.4.4 ***Social Network.*** : Since Twitter constitutes a social network, we can represent relations between users by using a graph data structure. We build a social graph for our dataset to use in one of the experiments in Section 4. We use "networkx" Python library[5] for graph representation. To build the graph, we go over user's tweet and find mentions with "tweet-preprocessor" Python library[6]. Then if one of two users is detected to mention the other at least in one of its tweets, these two users are considered to be connected on the social graph.

3 BASELINE

In this project, we created our baseline model based on the IGR model which is mention in the Text-Based Twitter User Geolocation Prediction Bo Han [1]. Our basic approach is to detect the Location Indicative Words in the tweets of users, to produce a representation (feature) vector for each user and to estimate which state the users live in by giving these vectors as input to the Multinominal Naive Bayes classifier and SVM.Baseline model implementation consists of 4 main stages; (1) Preprocess, (2) Feature Selection, (3)Creation of Users' Representation Vectors, (4)Classification.

3.0.1 ***Preprocess.*** :At first, in the data set reserved for training, tweets belonging to each user were collected in a list structure. By using CountVectorizer[7] method which is found in scikit-learn library, FeatureExtraction.text module, we (1)convert all characters to lowercase before tokenizing, (2)remove English stop words and punctuation, (3)tokenize each tweet (4) build a vocabulary with top 10000 token that have a document frequency lower than 0,95 and higher than 0,2. (5) create a sparse matrix of token counts for each tweet with a length of (10000). (6) convert each string value in the target column (State Name) to an integer value and the resulting integer representations placed in the State Index column.

3.0.2 ***Feature Selection.*** :To the Location Indicative Words in the dataset, we decide to use Information Gain which is the one of the Information Theory-Based Methods. We calculate the Mutual Information (MI) of the tokens that are found the vocabulary we created. For this process we prefer to MutualInfoClassif[8] method from

[1] https://nominatim.org/release-docs/latest/api/Overview/
[2] https://huggingface.co/dslim/bert-base-NER
[3] https://en.wikipedia.org/api/rest$_v$1/
[4] https://pypi.org/project/mordecai/
[5] https://networkx.org/
[6] https://pypi.org/project/tweet-preprocessor/
[7] https://scikit-learn.org/stable/modules/generated/sklearn.feature$_e$$xtraction.text.CountVectorizer.$
[8] https://scikit-learn.org/stable/modules/generated/sklearn.feature$_s$$election.mutual_info_classif.html$

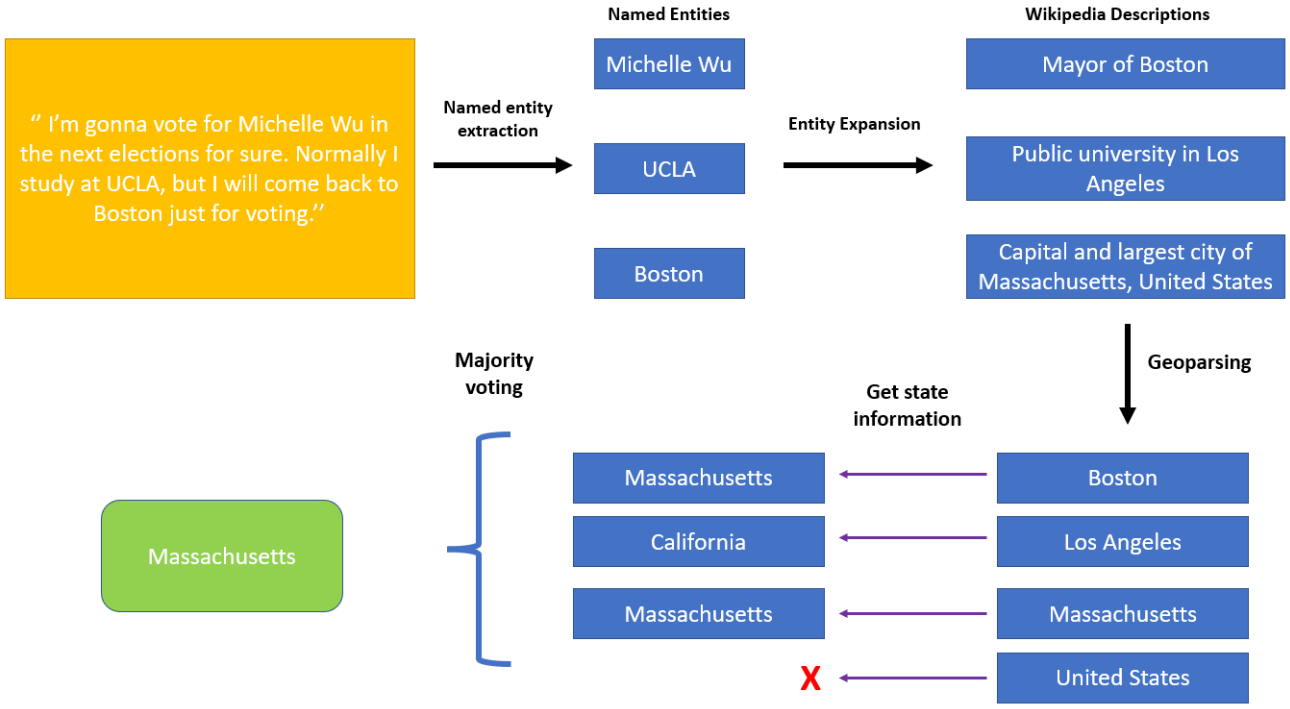

Figure 1: Steps in the proposed geolocation method

scikit-learn library,FeatureSelection module. After we calculate the MI of each token in vocabulary, we sorted tokens in descending order of MI values and create the Location Indicative Word Dictionary by choosing those with MI values greater than 0.0001.

3.0.3 ***Creation of Users' Representation Vectors.*** : For each tweet, by looking at the sparse token count matrix of the tweet, those whose value is different from 0 and found in the Location Indicative Word Dictionary are determined. The index value that these tokens correspond to in the binary representation vector is changed to 1. After creating the binary representation vector of each tweet, we performed a Bitwise OR operation between the binary vectors of tweets belonging to the same user and generated a binary representation vector for each user

3.0.4 ***Classification.*** : Training is performed by giving the representation vector and State Index value of each user in the training dataset as input to the Multinominal Naive Bayes[9] and SVM[10] classifiers. We apply preprocessing steps to test set, and give the information of the users in the test set to the classifiers to measure the performance of the models.

4 EVALUATION & EXPERIMENTAL RESULTS

In this section, we compare the proposed method to the baseline methods. Also we examine some design decisions about our method.

4.1 Evaluation Metrics

We use accuracy and precision metrics to measure prediction performances. We do not calculate neither accuracy nor precision label-specific. That is, accuracy is simply the rate of the number of correct state predictions to the number of users in test data.

As we explain in Section 2.2, one of the nice properties of our method is that it does not return a prediction result if the textual content does not contain any clues, which are named entities for our model. This property will not make a difference in terms of accuracy score, because not submitting a result is the same as false answer. However, non-submitted predictions are ignored by precision metric. In particular, precision is the rate of the number of correct predictions to the number of submitted predictions. Accordingly, we also use precision metric to emphasize this property of our method.

4.2 EBUG versus Baseline Methods

	Accuracy	Precision
EBUG	0.146	0.266
Baseline NB	**0.408**	**0.363**
Baseline SVM	0.382	0.353

Table 2: EBUG versus baseline method

Results on **Table 2** show that the performance of the model we designed based on the precision and accuracy metrics we use to compare the models is lower than the performance of our baseline

[9]https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html
[10]https://scikit-learn.org/stable/modules/svm.html

model.It is seen that the Multinomial Naive Bayes model shows the best performance among these three models.It is thought that the main reason for the performance difference between the baseline models and the EBUG to be so high is that all data was used while creating the Location Indicative Word Dictionary structure.

4.3 Contribution of Entity Types

As explained in the **Section 2.1**, we make use of all types of entity rather than using only location entities. We believe that other entities bring additional information about the author location. In this experiment, we investigate entity-type-specific contribution to prediction performance.

	Accuracy	Precision
All	**0.146**	0.266
Location	0.113	**0.341**
Organization	0.039	0.179
Person	0.014	0.075
MISC	0.015	0.119

Table 3: Impact of entity types on location prediction performance

Results on **Table 3** show that location entities are the most informative entities of all entity types. However, maximum prediction scores are obtained when all type are used together which refers that other entity types contribute additional information indeed as we expect. Location entities are followed by organization, person and misc entities.

4.4 Contribution of Neighborhood Information

Social networks are super-common structures that represent communities, individuals and relations between them. They contain lots of interesting properties which can be useful to understand underlying patterns and structures. Accordingly, including the information gathered from the neighborhood into prediction process can increase the prediction performance. In this experiment, we extend the method described in the Section2.1 by including candidate states extracted from the Twitter friends into voting process. In particular, if a user's friend mentions a real-world object that has a real-world location, this affects the prediction result of the user.

Neighborhood Utilization	Accuracy	Precision
No	0.146	0.266
Yes	**0.23**	**0.29**

Table 4: Performance impact of utilizing neighborhood entities

As it can be seen on **Table 4**, neighborhood information makes a dramatic contribution to prediction performance. This result also confirms the fact that people who are close in social media are close also geographically.

4.5 Weighting Entity Types in Voting

Till now we have showed that all entity types make a contribution to prediction more or less. However some entities may be more important than the others, e.g. location entities. In this experimental setup, we investigate how weighting entity types during voting process would change the prediction performance. Here we try a set of weight combinations. Since we have four types of entities (Location, organization, person, miscellaneous), we represent weight sets with a four-element tuple (e.g., (4,3,2,1)), where the numbers are the weights for location, organization, person, miscellaneous respectively. Till now all entity types are treated equally in voting, so weight set of (1,1,1,1) equals to that.

Weighting Combinations	Accuracy	Precision
(1,1,1,1)	0.146	0.266
(4,3,2,1)	0.145	0.268
(1,2,3,4)	0.135	0.249
(2,1,1,1)	0.145	0.268
(1,2,1,1)	0.144	0.264
(1,1,2,1)	0.141	0.260
(1,1,1,2)	0.145	0.267
(4,4,1,4)	**0.146**	**0.269**

Table 5: Impact of weighting between entity types for the voting process

We present results for a limited set of weight combination on **Table 5** since we cannot test each possible combination. After trying a couple of weight combination, we observe that weighting makes so minimal differences.

5 CONCLUSION

As a result of the experiments we made while designing our model, it was seen that location entities are the entity that gives the most information for the location of the tweet among all entity types. However, as a result of our experiments, tweet location estimation using organization, person and misc entities together with location entity is more successful than location estimation using only location entities. Moreover, it has been seen that estimating the location of the users based on the contents of the tweets of other users with which the users are related gives a better result. This also confirms that people associated with social media are located in close geographical locations.In addition to these observations, it is seen that our model is less successful than the baseline model. One of the reasons for this is thought to be that the feature selection of our baseline method covers with entity types and non-entity terms which have high MI value.

REFERENCES

[1] Timothy Baldwin Bo Han, Paul Cook. 2014. "Text-Based Twitter User Geolocation Prediction". (2014). https://www.jair.org/index.php/jair/article/view/10869/25929
[2] Bo Han, Paul Cook, and Timothy Baldwin. 2012. Geolocation prediction in social media data by finding location indicative words. In *Proceedings of COLING 2012*. 1045–1062.
[3] Noah A. Smith Eric P. Xing Jacob Eisenstein, Brendan O'Connor. 2010. "A Latent Variable Model for Geographic Lexical Variation". (2010). http://www.cs.cmu.edu/~nasmith/papers/eisenstein+oconnor+smith+xing.emnlp10.pdf