**Title:**

**SPOT: Locating Social Media Users based on Social Network Context.**

<https://github.com/manassharma/Geolocation-Of-Microbloggers/blob/master/ReferencePapers/p1681-liu.pdf>

**Highlights:**

The demo shows three different location estimation algorithms:

a friend-based,

a social closeness-based,

an energy and local social coefficient based

This project gives a web based interface to configure settings like what kind of users we need to filter, estimation models and also visualizes the location of user and the predicted location along with the location of friends whom were considered.

Play with this website: http://hpproliant.cse.unt.edu/locationdemobeta

**Friend based model ->**[**http://www.cameronmarlow.com/media/backstrom-geographical-prediction\_0.pdf**](http://www.cameronmarlow.com/media/backstrom-geographical-prediction_0.pdf)

**The paper I have added as a link is based on the following ideas:**

The data set used is address updated in facebook profile of candidates

Being close and spatial arrangement play a huge role in determining the likelihood of being friends is the main idea of this project.

The tasks include:

* Identify location of user and his friend using facebook address.
* Form a graph between friends and identify the correlation of friendship with distance.
* Plot population density and infer that people know each other well in less populated areas (i.e.) no. of friends within a mile in less populated area > highly populated area.
* Plot this and you will derive that friendship is inversely proportional to distance.
* However, people living in metropolitan area will have more friends in distant places.
* So, they have devised a model to calculate friendship based on location of the user and his friends.
* They use distance and rank (no of mutual friends who live in the area btw two friends)
* Use the test data (for which you already have the geolocation attributes captured) and compare the results.

So Friend based model uses this algorithm (considering distance and probability of being friends.

**Social Tightness Based Model:**

Consider that two users A and B have friends A1 and B1 respectively. This technique takes into consideration, the number of mutual friends A and B has to identify if they have social closeness.

**Energy and Local Social Coefficient Model:**

This model also uses social closeness metric and tries to come up with a total energy of a user being at a location.

**Confidence-based iteration method:`**

This paper heavily relies on the location info of a user’s friend. Incase there is no info about the user’s friend, they use a refining technique in which friends’ location is estimated and then iteratively improvise. But they didn’t provide the details of the algorithmused.

Since they extended the model used on Fb to twitter data, they considered people who follow each other as friends and the initial location is considered to be the one from where he tweets more. (THIS INFORMATION IS EXTRACTED FROM TWITTER API AND THE MOST FREQUENT LOCATION IS CONSIDERED THE USER’S HOME LOCATION).

**Title:**

**Location Extraction From Disaster-Related Microblogs**

<https://github.com/manassharma/Geolocation-Of-Microbloggers/blob/master/ReferencePapers/p1017.pdf>

**Highlights:**

Instead of NER libraries, previous works used TF-IDF algorithm to select important words in a tweet. Also, geolocation of tweets (with location info) and store it prediction reference.

However this paper uses NER libraries on annotated tweets.

They have evaluated a lot of NER libraries and compared the results:

* Standford NER: Uses Conditional Random Fields Technique wherein it predicts based on sequence of labels rather than unigrams.
* OpenNLP: Uses maximum entrophy model for classification. Also considers the whole document text for entities classification.
* Yahoo PlaceMaker: It can extract exact latitude , longitude info based on address etc.
* Twitter NLP: Mainly focused to work on tweets and cannot be retrainable. But extensively tries to interpret tweets.

Since these libraries do not come with native location classification support, they retrained OpenNLP& Stanford NER using twitter data and did cross verification. For Twitter NLP, they use a gazetteer as a reference which in turn hindered Twitter NLP performance.

Tweets might also be of the form #Australia or #Fashion. So results were computed with and without hashtags and compared.

Python has extensive support for the above mentioned libraries.

(Since this project deals with disaster related tweets, they have made use of CrowdFlower to annotate tags as disaster related. So there has been manual work involved in classifying the tags from various timelines.

So they have followed controlled annotation where in people would be given same subset of tweets and their classification results were validated. CrowdFlower assigned trust level to people involved in annotating so that their work can be picked with confidence. This part is to describe how they identify disaster related tweets).

Based on comparison, Standford NER outperformed other libraries after it was retrained with tweets. But finding Point of Interest seems to be a bit complicated.

**TITLE : Effective Location Identification from Microblogs**

**ICDE Conference, 2014**

https://github.com/manassharma/Geolocation-Of-Microbloggers/blob/master/ReferencePapers/icde14\_location.pdf

**GOAL** : Proposes a global location identiﬁcation method, called GLITTER which combines multiple microblogs of a user and utilizes them to identify the user’s locations.

**HIGHLIGHTS :**

* infers top-k locations of a user from his/her microblogs and top-k locations for each of his/her microblogs
* organizes points of interest (POIs) into a tree structure where the leaf nodes are POIs and non-leaf nodes are segments of POIs
* extracts candidate locations from microblogs, aggregates them and generates top-k locations of the user, reﬁnes them further to compute top-k locations of each microblog
* fuzzy matching between locations and microblogs to achieve high recall

**METHOD :**

Tree-based Location Structure (TLS):

* location hierarchy obtained from existing public database Yago (http://www2007.org/papers/paper391.pdf)
* based on the POI, the location segments from the database are used to construct the tree based structure where non-leaf node -> Range Location and leaf nodes -> Point Location
* each node is assigned a Dewey code in top-down manner

**FRAMEWORK**:

1. **Extract** substrings of each microblog which correspond to locations in the TLS
2. **Aggregate** candidate locations of every microblog and generate top-k user locations
3. **Refine** the TLS with each addition of a POI

**PERFORMANCE:**

High quality and scalability when compared to previous implementations in terms of location accuracy and specific details.

This paper describes in detail the implementation of the GLITTER framework. It seems to achieve location accuracy (of the user and of the microblog) better than other proposed methods at that time. Performance statistics also look good.

**Title**: **Content based geolocation for Placing Tweets Pertaining to Trending News on Map: muse2013**

-> Arabic to English

-> Create a user's document page

Phase 1: Apply NER

Phase 2: Locations Database->Match the NER give a score->

Perform K means clustering on the locations

If the location is near to the centroid then multiply the score

Phase 3: Iterate through the list and score based on order of occurrence

**Title:** **Location Inference Using Microblog Text and Friendships**

-> Mainly uses CRF++

3 ways:

1. Location specific estimation eg rainstorm

Train CRF such that you identify location with time(today)

1. Time (to know if u r travelling)

To filter users travelling have a ranking, where value increase slowly if there are more posts

But if posts occur shortly, then increase is not much

1. Friends circle

Retweet+recent tweets+time to measure closeness> threshold yes

**Preprocessing data sets:**

Less than 40 chars

Followers less than frnds