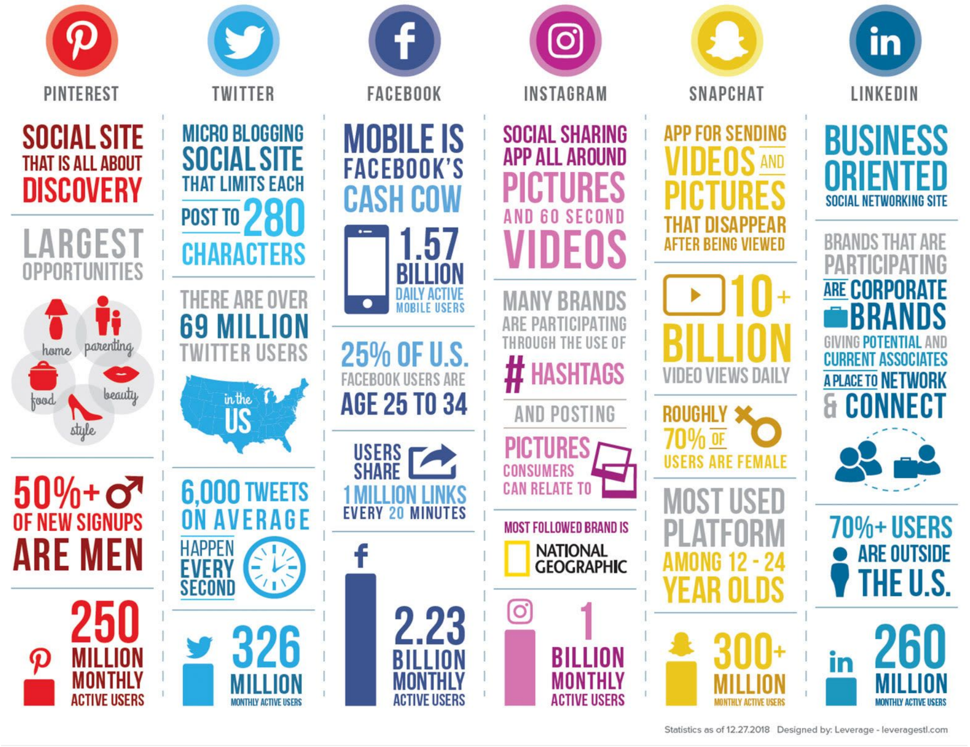
Geolocation Extraction for Twitter

- Varad Pimpalkhute, affiliated with Indian Institute of Information Technology, Nagpur.
- Project Advisor: Dr. Arjun Magge.
- Supervisor: Dr. Graciela Gonzalez-Hernandez.

Social Media Mining

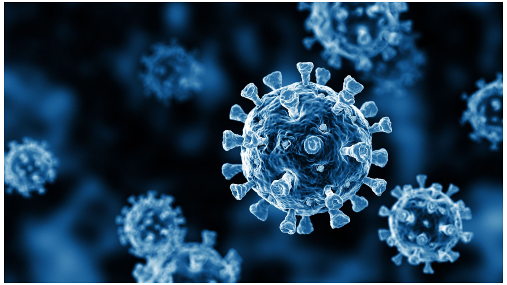


Challenges Faced in Social Media Mining

- Non Standard Text.
- Lack of consistent geolocation information.
- Large Data Volumes.
- API Limitations
- Text Mining

Why do we need geolocation of a social media user?

- Tracking Infectious Diseases.
- Gathering analysis on how well an advertisement is received in a particular region.
- Keeping tabs on spread of diseases such as COVID-19.



Task : Given an input tweet and user screen name, identify the user's current location of residence.

Let’s Take an Example

User Screen Name :

Input User Tweet :

Challenges Faced in extraction of locations

- Sparse availability of dataset for training/experimenting the user locations.
- Assume that the user has not moved from his/her previously current location.
- Many of the users might have deleted their account handles, thus rendering it difficult for us to extract the metadata of the user.
- In the profile tag, the user might input wrong/incorrect location name.
- Locations in user metadata amount to only 1% of all users.

USER METADATA

Some of the important location related tags are:

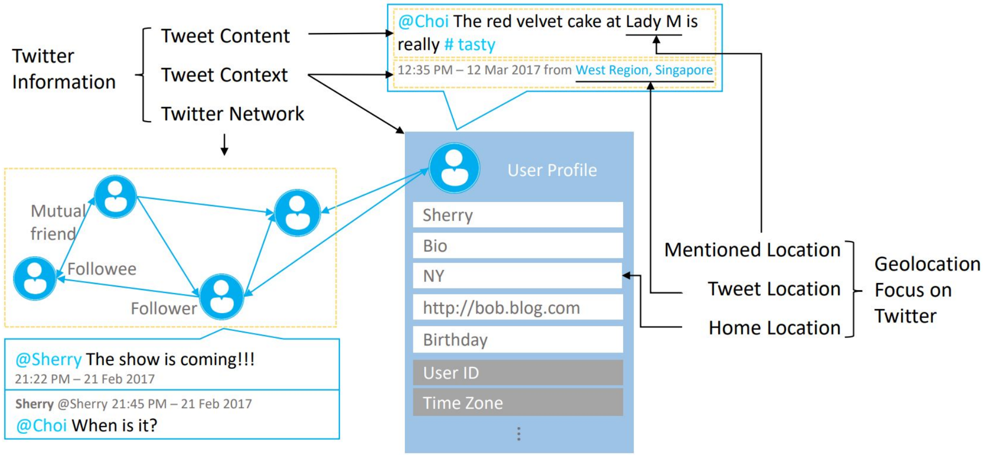
- Geo tag
- Place tag
- Profile tag

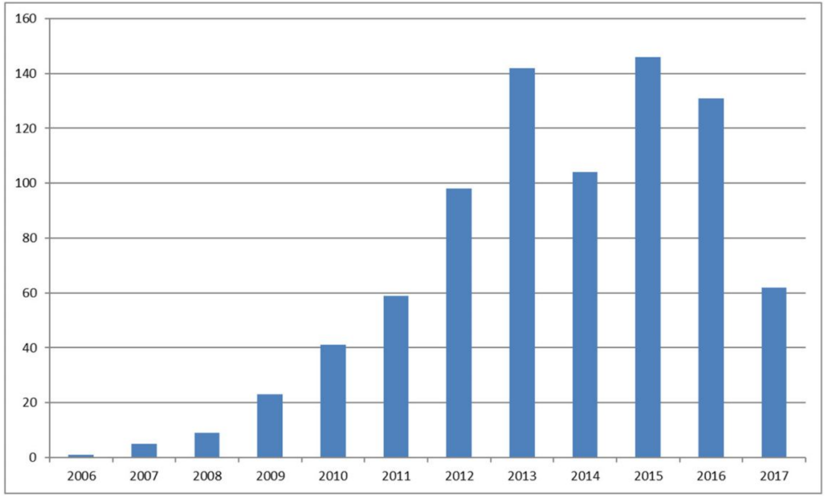

```
{
    "id":887658871220830208,"id_str":"887658871220830208","name":"Coco \ud83d\udc95","screen_name":"coco_chenelle_",
    "location":"South Africa","url":"https:\/\/www.instagram.com\/coco.chenelle_","description":"\ud83d\udccdRandburg",
    "translator_type":"none","protected":false,"verified":false,"followers_count":4025,"friends_count":2789,
    "listed_count":0,"favourites_count":7171,"statuses_count":6648,"created_at":"Wed Jul 19 13:02:21 +0000 2017",
    "utc_offset":null,"time_zone":null,"geo_enabled":true,"lang":null,"contributors_enabled":false,
    "is_translator":false, "profile_background_color":"F5F8FA","profile_background_image_url":"",
    "profile_background_image_url_https":"", "profile_background_tile":false,"profile_link_color":"1DA1F2",
    "profile_sidebar_border_color":"C0DEED", "profile_sidebar_fill_color":"DDEEF6","profile_text_color":"333333",
    "profile_use_background_image":true, "profile_image_url":"http:\/\/pbs.twimg.com\/profile_images\/131630663404667289
    "profile_image_url_https":"https:\/\/pbs.twimg.com\/profile_images\/1316306634046672897\/hPYqfULA_normal.jpg",
    "profile_banner_url":"https:\/\/pbs.twimg.com\/profile_banners\/887658871220830208\/1606087684","default_profile":tr
    "default_profile_image":false,"following":null,"follow_request_sent":null,"notifications":null},"geo":null,
    "coordinates":null,"place":null,"contributors":null,"is_quote_status":false,"quote_count":0,"reply_count":0,
    "retweet_count":0,"favorite_count":0
}
```


Tools used for extracting user metadata:

- Tweepy
- Twitter API

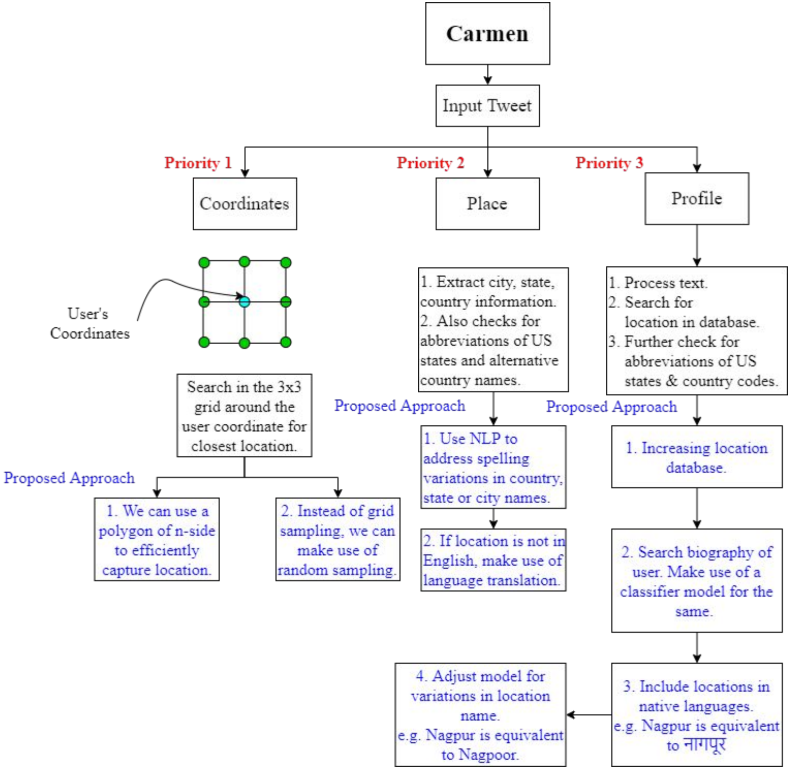
Related Work



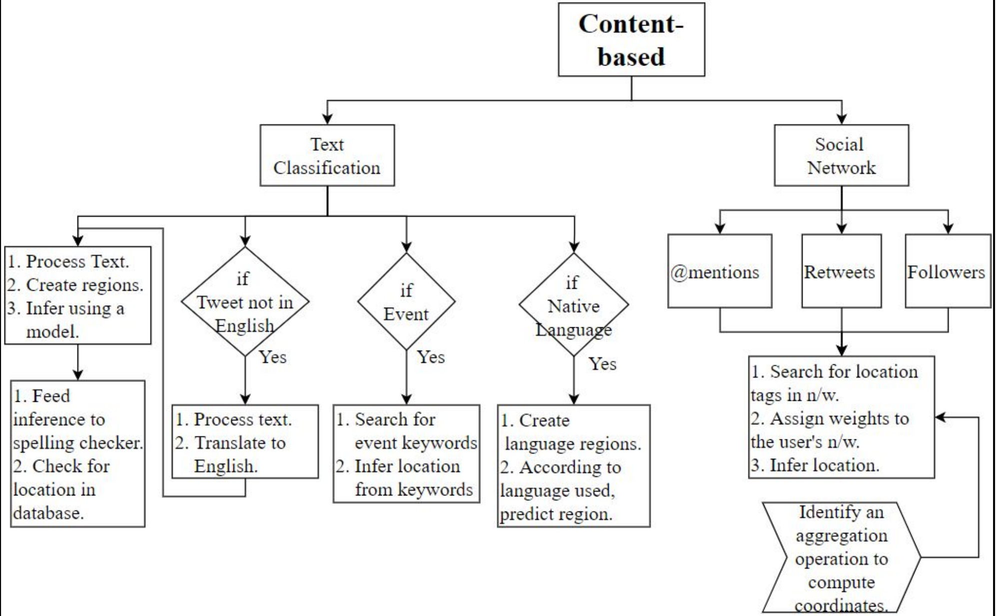


Exemplar methods in literature

Carmen



Pigeo



Goal

1. Improve on Carmen and Pigeo. Finally, integrate both the methodologies in one system.
2. Generating a dataset for testing our model by scrapping user data from Twitter using tools like Tweepy.
3. Addressing the issue of classification on a hierarchical level.

Scope of Today's PPT

Improve on Carmen and Pigeo. Finally, integrate both the methodologies in one system.

1. Improve on existing location database.
2. Make use of population heuristics.
3. Content based classification.
4. Aggregation on user tweets for meta-based location prediction.

Methodology

1. Improving Location Database.

- ➢ Carmen's old location database limited in scope.
- ➢ Carmen has its own system for indexing locations.
- ➢ Need for a unified service. Thus, we make use of GeoNames Service.
- ➢ Features of GeoNames:
    - Contains 25 million geographical locations.
    - Also, includes 13 million alternate names.
    - At a high level, can be categorized in four parts:
        - Country level locations
        - State level locations
        - County level locations, &
        - City/Town level locations.

geonameid	integer id of record in geonames database
asciiname	name of geographical point in plain ascii characters
alternatenames	comma separated, ascii names automatically transliterated
latitude	latitude in decimal degrees
longitude	longitude in decimal degrees
feature class	is the location a landmark, a city, a river, etc.
country code	ISO-3166 2-letter country code
admin1 code	code for the first administrative division
admin2 code	code for the second administrative division
population	bigint population of the location

In order to make our search even better, we also add a Radius parameter to the database.

Radius: An approximate radius of the location (in kms)

Tool used: **GADM**

Extracting Radius using GADM.

Libraries used: Geopandas, fiona.

```
def extract_radius(data):
    data = data.to_crs(epsg=3035) # Convert to make the map planar
    # Compute area in km^2
    area_k = np.round(data['geometry'].area/1000000, 2)
    area_m = np.round(area_k*0.386102, 2)
    radius = np.round(np.sqrt(area_k/np.pi), 2)
    data['Area_km2'] = area_k # in km^2
    data['Area_m2'] = area_m # in miles^2
    data['Radius'] = radius # in miles
    return data
```

Illustration of a location info in the database.


```
{"id": 5809844, "unzip": "", "city": "Seattle", "population": 684451,
    "latitude": 47.60621, "longitude": -122.33207, "county": "King County",
    "countycode": "033", "state": "Washington", "statecode": "WA",
    "country": "United States", "countrycode": "US", "radius": 26700,
    "postal": "", "parent_id": "5799783", "aliases": ["siyaattl", "sijetl",
    "ciyaattttil", "syatl", "sietl", "siiaittl", "siitthl", "siat’l",
    "seyaatele", "xi ya tu ", "ciyattil", "siyatala", "seattlum", "sietla",
    "siehtl", "si’aitala", "seatl", "siet‘l", "sietlas", "shiatoru",
    "sietli", "siatul", "sytl", "siyaatil", "chiiae‘tethil", "syy’ttl",
    "siaeteul", "siietl", "siet’li", "siyatal", "siatl", "sea", "seattle",
    "xi ya tu", "syttl", "siy’aattl", "siaenttl"]}
```

Population Heuristics

```
--------------------------------------------------------------------
Alias 1: obshtina byala
Location(country='Republic of Bulgaria', state='Varna', county='Obshtina
    Byala', id=732718, population=3287)
Alias 2: obshtina byala
Location(country='Republic of Bulgaria', state='Oblast Ruse',
    county='Obshtina Byala', id=732719, population=11958)
--------------------------------------------------------------------
```

Content-based Classification

Location can be detected in a tweet based on the user's:

- Dialect
- Mention of landmarks
- Regional issues, etc.

Thus, we exploit this information in the content to determine a user's approximate location.

Approaches

There are many approaches used for determining a user's location from content such as:

1) N-Grams
2) Segmentation of geographical space (using k-tree)
3) Geometric Decomposition
4) Deep Learning Classifier Models.

We build our model on Fasttext, and Transformers. Fasttext is more lightweight and suitable for experiments, Transformers are used for improving accuracy.

Training models - Fasttext

Input: Country name, Tweet

Output: Predicted User's Location - State name.

```
# Trains basic classifier using input training data.
import fasttext
model = fasttext.train_supervised(input="training data here")
model.save_model("model.bin") # Save model binary
model.test("valid data") # Testing model
```

Results

1) An instance of content model on Germany dataset.

Architecture	LR (x 10^{-6})	F1-Score	Precision	Recall
BERT	10	0.872	0.843	0.902
BERTweet	10	0.899	0.896	0.906
DistilBERT	50	0.835	0.839	0.831
RoBERTa	6	**0.924**	0.897	**0.952**
XLNET	5	0.903	**0.922**	0.866

Results

2) Incorporating all improvements in our system.

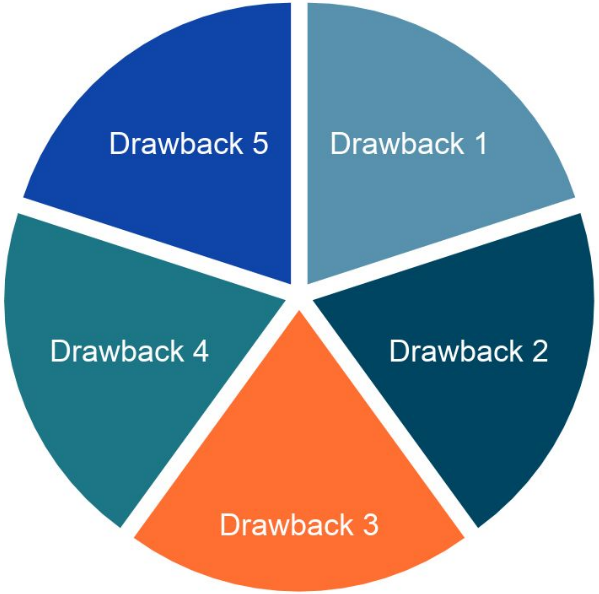
Dataset	#Tweets	#Place Resolved	#Profile Resolved
User Timeline	4069	21	704
Stream (29 Jan)	2784	12	378

System used: Carmen

Dataset	#Tweets	#Place Resolved	#Profile+ #Content Resolved
User Timeline	4069	29	3974
Stream (29 Jan)	2784	17	2443

System used: Carmen-Plus

A few drawbacks of the system



1. Resolution at city/county level is poor.
2. Hierarchical Classification is not feasible.
3. Transformers are computationally expensive.
4. Doesn’t account for spelling variations.
5. Radius is generalized, thus radius data is often not accurate.

References

[1] X. Zheng, J. Han and A. Sun, "A Survey of Location Prediction on Twitter," in IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 9, pp. 1652-1671, 1 Sept. 2018, doi: 10.1109/TKDE.2018.2807840.

[2] Han, T. (2012). Geolocation Prediction in Social Media Data by Finding Location Indicative Words. In Proceedings of COLING 2012 (pp. 1045–1062). The COLING 2012 Organizing Committee.

[3] Roller, J. (2012). Supervised Text-based Geolocation Using Language Models on an Adaptive Grid. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (pp. 1500–1510). Association for Computational Linguistics.

[4] Roesslein, J. (2020). Tweepy: Twitter for Python!. URL: https://github.com/tweepy/tweepy.

[5] Mark Dredze, Michael J Paul, Shane Bergsma, & Hieu Tran (2013). Carmen: A Twitter Geolocation System with Applications to Public Health. In AAAI Workshop on Expanding the Boundaries of Health Informatics Using AI (HIAI).

[6] Rahimi, A., Cohn, T., & Baldwin, T. (2016). pigeo: A Python Geotagging Tool. In Proceedings of ACL-2016 System Demonstrations (pp. 127–132). Association for Computational Linguistics.

Questions