Geolocation Extraction from Twitter

Varad Pimpalkhute[1,*], Arjun Magge[2,*], Vipin Kamble[1], Graciela Gonzalez Hernandez[2]

* Equal Contribution. https://bitbucket.org/pennhlp/carmen-plus/src

{pimpalkhutevarad, vipinkamble97}@gmail.com, {Arjun.Magge, gragon}@pennmedicine.upenn.edu

[1]Indian Institute of Information Technology, Nagpur

[2]Perelman School of Medicine, University of Pennsylvania

Geolocation Information Extraction

Aims to extract geolocation information from unstructured data:

Content Extraction

*"Barack Obama, a former U.S president, was born in **Hawaii**."*

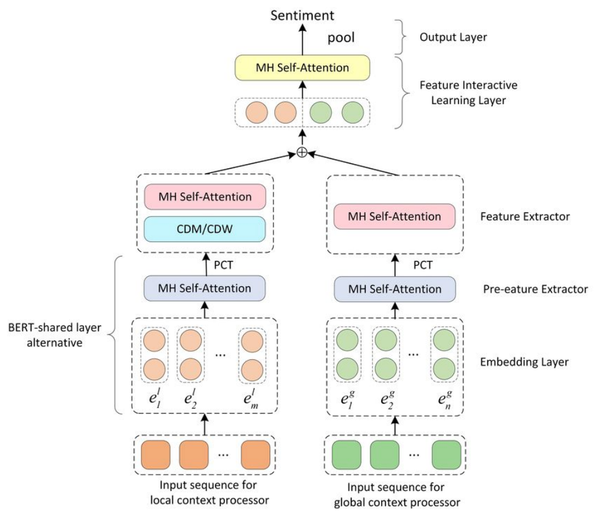
Metadata Extraction

Tweets on Twitter have a lot of metadata present at the time when the tweet is post. Geolocation related meta-tags are: a) Geo-coordinates b) Place location c) Profile location.

Carmen-Plus: We propose a system which improves on Carmen and aims to increase the search space. We improve on Carmen as follows:

Proposed Approach:

1) Improve on location database with help of GeoNames Service.
2) Make use of population heuristics, and add custom radius instead of keeping a generalized radius for all locations.
3) Add content based classification for states in a country.
4) Aggregation around each user to ensure better accuracy.



Training Data

We used the existing dataset such as GeoText [2], Twitter-WORLD [3], and UTGeo2011 [4] to compute the accuracy with which our system predicts user's location . Apart from these datasets, we also build up on custom datasets for ten countries with most active users on Twitter.

Carmen: Carmen is a geolocation system that infers structured location information based on both geo-coordinates and user profile information in the form of – country, state, county, city – for Twitter users [1].

Limitations of Carmen:

1) It works only when metadata is present. Thus, most of the times, Carmen won't work.
2) Location database is limited to only big cities, and towns.
3) It doesn't account for acronyms/aliases, or spelling variations.

Test Data: We test Carmen and Carmen-Plus (our system) on two datasets namely *"User Timeline"* and *"Streamed"*. *"User Timeline"* is generated by choosing a sample of random users from a user's followers, and extracting the tweets in their timeline. *"Streamed"* is a collection of general Twitter data collected on 29th January 2021.

Resolution Method : Carmen			
Dataset	**#Tweets**	**#Place Resolved**	**#Profile Resolved**
Homeline	4069	21	704
Streamed	2784	12	378

Resolution Method : Carmen-Plus			
Dataset	**#Tweets**	**#Place Resolved**	**#Profile Resolved**
User Timeline	4069	29	3974
Streamed	2784	17	2443

Custom Dataset for Countries

The dataset has been prepared by searching for four keywords mentioning the place of residence of Twitter users.

Country	#Regions	#Tweets
United States of America	55	788456
Republic of India	36	299298
United Kingdom of Great Britain	12	130404
Federal Republic of Germany	16	22022
People's Republic of China	34	64819

Keywords:

- "I live in …"
- "I reside in …"
- "I stay at …"
- "I am from …"

Evaluation

Comparative results of various transformer based models for custom dataset for Germany :

Architecture	LR (x 10^{-5})	F1-Score	Precision	Recall
BERT	3	0.872	0.843	0.902
BERTweet	3	0.899	0.896	0.906
DistilBERT	1	0.835	0.839	0.831
RoBERTa	6	**0.924**	0.897	**0.952**
XLNET	5	0.903	**0.922**	0.866

Based on the various experiments, we settled that the learning rate in the range of 0.00001 - 0.00006, batch size of 8, patience of 2 and 3 epochs of training gave the best performance on the models.

FastText

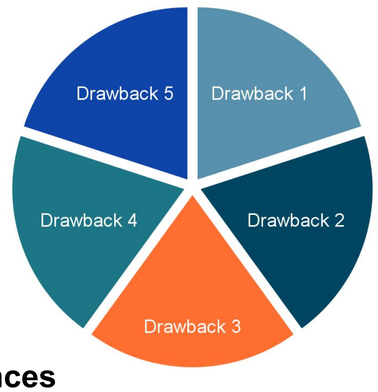
FastText is an open-source, free, lightweight library that allows users to learn text representations and text classifiers.

A simple fasttext model can be trained as shown in the snippets below.

```
# 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
```

Error Analysis

There are a few drawbacks of CarmenPlus. We have kept them as a part of future work. They are as follows:



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] Mark Dredze, Michael Paul, Shane Bergsma, & Hieu Tran (2013). Carmen: A Twitter Geolocation System with Applications to Public Health. AAAI, 20-24.

[2] Eisenstein, E. (2010). A Latent Variable Model for Geographic Lexical Variation. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (pp. 1277–1287). Association for Computational Linguistics.

[3] 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.

[4] 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.