10^{th} International Conference on Computational Social Science IC^{2}S^2
July 17-20, 2024, Philadelphia, USA

Where on Earth Do Users Say They Are?: Geo-Entity Linking for Noisy User Input

Tessa Masis (tmasis.github.io/) and Brendan O'Connor

Extended Abstract

The real-world geographic location of social media users is valuable data for many downstream tasks, including disaster response [6], disease surveillance [7], analyzing language variation [4], and comparing regional attitudes [9]. Many studies have used Twitter (now known as X) data for such analyses, focusing on geo-tagged tweets where each tweet is associated with latitude and longitude coordinates. However, geo-tagging with coordinates was deprecated in June 2019 and even before then only a small percentage of tweets ($< 2\%$) was geo-tagged [5]. It is thus increasingly necessary to infer location from user profiles and especially from the free text Location field, which is frequently specified with at least 40% of users providing recognizable locations [3]. The task of linking a location reference to the actual geographic location is known as *geo-entity linking* (see Table 1 for examples). There are few multilingual geo-entity linking tools available and existing ones are often either rule-based [1, 2], which break easily in social media settings, or LLM-based [11], which are too expensive for large-scale datasets.

In this work, we propose a method for geo-entity linking of noisy user-input location references to real-world geographic locations, by representing real-world locations with averaged embeddings from labeled user-input location names. Unlike previous methods, our method enables selective prediction via an adjustable threshold for cosine similarity scores, which we analogize with confidence scores. We compare performance of multiple variations of our proposed method on a global, multilingual dataset and show that all of them outperform the leading baseline. We plan to further evaluate our method on other domains with noisy location references, such as historical data, and to explore extensions of our method so it may be used for the broader task of geoparsing any unstructured text.

Task definition: We define our geo-entity linking task as follows. Given a real-world geographic location database D, a training set T containing user-input location name and ground truth location pairs, and an unlabeled user-input location name n, we model

$$\arg\max_{d \in D} P(d|T,n)$$

where each geographic location $d \in D$ is represented by a triple containing a city name, primary administrative region name (e.g. state, province), and country name. The city and administrative region names may be empty strings if the location is at a higher granularity, e.g. a country. We note that a user-input location n may be noisy and contain no real locations; because of this, predicted location triples may be composed of only empty strings, indicating that no location could be confidently predicted.

Data: For our real-world geographic location database D, we use a modified version of the GeoNames[1] database, which contains location names and coordinates for over 11M countries, administrative regions, counties, and cities across the globe. We filtered this database to exclude cities with populations under 15,000; our final database contains 75,000 total locations.

[1] https://www.geonames.org/

For our train set T and test set, we use data from the Twitter-Global dataset [10], which contains data from 15.3M global tweets posted from 2013-2021. We use the 4.1M tweets from this dataset which are geo-tagged and posted by users with a non-empty Location field. The coordinates for each geo-tagged tweet were mapped to the closest city in our modified GeoNames database; this location was used as the ground truth location for the tweet.

Method: Our proposed method (which we refer to as *UserGeo*) starts with an embedding representation for each location $d \in D$ and, for a given user input n, calculates the cosine similarity between the embedded user input and each embedded location in D; the location embedding with the highest cosine similarity is the predicted real-world location. If the cosine similarity with all location embeddings is sufficiently low, then no prediction is made.

The key contribution of our method is the way that the embeddings for the locations in D are created: First, the name of each location in D and each user input in the training set T are embedded using an SBERT model [8]. Each user input embedding is associated with its ground truth location. Then, each location in D is represented by the averaged embedding of its location name and all associated user input embeddings. The motivation behind this method is that it leverages millions of examples of user-defined location names, essentially creating a user-defined location embedding database.

We also evaluate a variation of our method (referred to as *NameGeo*) where the locations in D are represented only by the location name embedding (that is, no user-input data is used).

Experiments: We divide the Twitter-Global data into a 90/10 split, with 3.7M examples in the training set and .4M in the test set. We evaluate each of our methods with three SBERT base models: the popular *all-MiniLM-L6-v2* model, the multilingual *paraphrase-multilingual-miniLM-L12-v2* model, and the larger *all-mpnet-base-v2* model. We also evaluate our methods with a cosine similarity threshold of 0.5 – if the predicted location has a cosine similarity less than the threshold, this is interpreted as a low confidence score for the prediction and thus the predicted location is replaced with a triple of empty strings.

We evaluate at three levels of geographic granularity (city, administrative region, and country) and use five metrics for evaluation (coverage, accuracy, precision, recall, and F1-score; see Table 3 caption for metric definitions). We compare performance with the one prior open-source and multilingual method, Carmen 2.0 [10], which uses a combination of regular expressions and manually curated aliases to predict real-world locations.

Results: We include results in Table 3. UserGeo has the highest coverage, accuracy, recall, and F1-score, with gains over Carmen 2.0 in the range of 20 to 50 points. And while Carmen 2.0 has the highest precision, it also has the lowest coverage, accuracy, and recall – in other words, it is often correct when it makes a prediction but it does not often make a prediction. In contrast, our proposed methods all have higher coverage, accuracy, recall, and F1-scores. The 0.5 cosine similarity threshold also demonstrates the ability to adjust the precision/recall balance if, for a given application, it is more important to get predictions correct or if it is more important to make more predictions. (See Table 2 for error analysis examples.)

Across SBERT bases, the *all-MiniLM-L6-v2* model surprisingly performs better than the multilingual *paraphrase-multilingual-miniLM-L12-v2* model and performs comparably with the larger *all-mpnet-base-v2* model. Comparing accuracy across countries, we observe that the number of examples per country in the training set (which may differ by multiple orders of magnitude) does not seem to influence test set performance, which suggests that an unbalanced training set doesn't negatively impact performance as it might for a traditional supervised learning method.

References

[1] ALEX, B., LLEWELLYN, C., GROVER, C., OBERLANDER, J., AND TOBIN, R. Homing in on twitter users: Evaluating an enhanced geoparser for user profile locations. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (2016), pp. 3936–3944.

[2] DREDZE, M., PAUL, M. J., BERGSMA, S., AND TRAN, H. Carmen: A twitter geolocation system with applications to public health. In *AAAI workshop on expanding the boundaries of health informatics using AI (HIAI)* (2013), vol. 23, Citeseer, p. 45.

[3] HUANG, B., AND CARLEY, K. M. A large-scale empirical study of geotagging behavior on twitter. In *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining* (2019), pp. 365–373.

[4] HUANG, Y., GUO, D., KASAKOFF, A., AND GRIEVE, J. Understanding us regional linguistic variation with twitter data analysis. *Computers, environment and urban systems 59* (2016), 244–255.

[5] KRUSPE, A., HÄBERLE, M., HOFFMANN, E. J., RODE-HASINGER, S., ABDULAHHAD, K., AND ZHU, X. X. Changes in twitter geolocations: Insights and suggestions for future usage. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)* (2021), pp. 212–221.

[6] KUMAR, A., AND SINGH, J. P. Location reference identification from tweets during emergencies: A deep learning approach. *International journal of disaster risk reduction 33* (2019), 365–375.

[7] LEE, K., AGRAWAL, A., AND CHOUDHARY, A. Real-time disease surveillance using twitter data: demonstration on flu and cancer. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (2013), pp. 1474–1477.

[8] REIMERS, N., AND GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (2019), pp. 3982–3992.

[9] ROSENBUSCH, H., EVANS, A. M., AND ZEELENBERG, M. Interregional and intraregional variability of intergroup attitudes predict online hostility. *European Journal of Personality 34*, 5 (2020), 859–872.

[10] ZHANG, J., DELUCIA, A., AND DREDZE, M. Changes in tweet geolocation over time: A study with carmen 2.0. In *Proceedings of the Eighth Workshop on Noisy User-generated Text (W-NUT 2022)* (2022), pp. 1–14.

[11] ZHANG, J., DELUCIA, A., ZHANG, C., AND DREDZE, M. Geo-seq2seq: Twitter user geolocation on noisy data through sequence to sequence learning. In *Findings of the Association for Computational Linguistics: ACL 2023* (2023), pp. 4778–4794.

User-input location	**Real-world location**	**Type of noise**
TURKEY/SİNOP	Sinop, Sinop, TR	Uncommon punctuation use
福島県いわき市	Iwaki, Fukushima, JP	Non-Latin script
Catskills	Hyde Park, New York, US	Informal/alternative name
where the wild things are	N/A	Not a real location

Table 1: Examples of user-input location references, the real-world locations they should be linked with, and the type of noise that the geo-entity linking model must be able to handle.

User-input	**Carmen 2.0**	**NameGeo @0.5**	**UserGeo @0.5**
TURKEY/SİNOP	””, ””, ””	””, Sinop, TR	Boyabat, Sinop, TR
福島県いわき市	””, ””, ””	Zhongshu, Yunnan, CN	Iwaki, Fukushima, JP
Catskills	””, ””, ””	Catalca, Istanbul, TR	Greenburgh, New York, US
where the wild things are	””, ””, ””	””, ””, ””	””, ””, ””

Table 2: Error analysis of the same user-input examples as in Table 1 (see Table 1 for corresponding real-world locations). Results from NameGeo and UserGeo are using the *all-MiniLM-L6-v2* SBERT model. Empty strings indicate that the model was not able to confidently make a prediction at that level. We observe that: (1) Carmen 2.0 rarely makes predictions for user-inputs with unexpected punctuation or in non-Latin scripts, (2) NameGeo often incorrectly predicts locations that look superficially similar to the user input (e.g. it predicts a location in China for a user input written in Japanese, and a location named 'Catalca' for the user-input 'Catskills'), (3) UserGeo often correctly predicts locations for non-Latin inputs and alternate/informal location names, and (4) all three models are frequently able to identify user-inputs that are not real locations.

Model	**Coverage (%)**	**Accuracy (%)**	**Precision**	**Recall**	**F1-score**
Carmen 2.0	47.28	43.50	**.92**	.45	.61
NameGeo	**100.00**	59.74	.60	**1.00**	.75
NameGeo @0.5	73.5	56.82	.78	.68	.72
UserGeo	**100.00**	**67.49**	.67	**1.00**	**.81**
UserGeo @0.5	90.73	65.94	.73	.88	.79

Table 3: Preliminary results on Twitter-Global data, with all metrics at the country-level. *NameGeo* refers to the method where locations are represented by embedded location names; *UserGeo* refers to the method where locations are represented by averaged user-input location names; results for both methods are using *all-MiniLM-L6-v2* SBERT model. The *@0.5* indicates that a cosine similarity threshold of 0.5 is being used, where predictions are not made for user inputs with cosine similarities below this threshold. *Coverage* is the percentage of examples for which the method made a location prediction (i.e. did not predict a triple of empty strings). *Accuracy* is the percentage of examples for which the method made a correct location prediction. We use standard definitions for *precision*, *recall*, and *F1-score*, where a true positive is a correct prediction, a false positive is an incorrect prediction, and a false negative is no prediction (i.e. a triple of empty strings).