GazPNE2: A general and annotation-free place name extractor for microblogs fusing gazetteers and transformer models

Xuke Hu [1]

Institute of Data Science, German Aerospace Center (DLR), Germany
xuke.hu@dlr.de

Zhiyong Zhou

Department of Geography, University of Zurich, Switzerland
zhiyong.zhou@geo.uzh.ch

Jens Kersten

Institute of Data Science, German Aerospace Center (DLR), Germany
Jens.Kersten@dlr.de

Matti Wiegmann

Web Technology and Information Systems, Bauhaus-Universität Weimar, Germany
matti.wiegmann@uni-weimar.de

Friederike Klan

Institute of Data Science, German Aerospace Center (DLR), Germany
Friederike.Klan@dlr.de

Abstract

Extracting precise location information from microblogs is a crucial task in many applications. Currently, there remains a lack of a robust and widely applicable place name extractor for English microblogs. In this paper, we attempt to overcome the gap by presenting GazPNE2, which fuses deep learning, global gazetteers (e.g., OpenStreetMap), pretrained transformer models, and rules requiring no manually annotated data. GazPNE2 can extract place names at both coarse (e.g., country and city) and fine-grained (e.g., street and creek) levels and place names with abbreviations (e.g., *'tx'* for *'Texas'* and *'studemont rd'* for *'studemont road'*). We compare GazPNE2 with 9 competing approaches on 11 public tweet data sets, containing 21,393 tweets and 16,790 place names across the world. It is the first time that different extractors are compared on such a large public dataset. The results show our proposed approach achieves SotA performance on the test data with an average F1 of 0.8. Code is available on the GitHub page: https://github.com/uhuohuy/GazPNE2.

2012 ACM Subject Classification Artificial intelligence → Information extraction

Keywords and phrases Location extraction; Gazetteer; Transformer model; Microblogs

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.53

1 Introduction

Social media platforms, such as Twitter and Weibo, are often the first place where situational information about current events is publicly posted. When an emergency event occurs, extracting location information from social media is crucial to inform people and authorities about affected areas and the locations of people in need. However, tweets are rarely geo-tagged. Thus, it is necessary to extract location information from tweet texts. This task is called location extraction and consists of two steps: place name extraction and geocoding. This study focuses on place name extraction.

[1] Corresponding author

11th International Conference on Geographic Information Science (GIScience 2021).
Editors: Piotr Jankowski; Article No. 53; pp. 53:1–53:6
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

However, all current approaches for place name extraction from microblogs have fundamental flaws: rule-based methods [2] do not generalize well, gazetteer-based methods [7] do not handle the place name ambiguity and variation issues well, and deep learning methods [12] require manually annotated data at an unfeasible scale. In this paper, we present a novel place name extractor, which first detects place names in tweets using a neural classifier that was trained on gazetteers, and then uses transformer models to resolve the ambiguities produced by the neural model.

2 Overall Approach

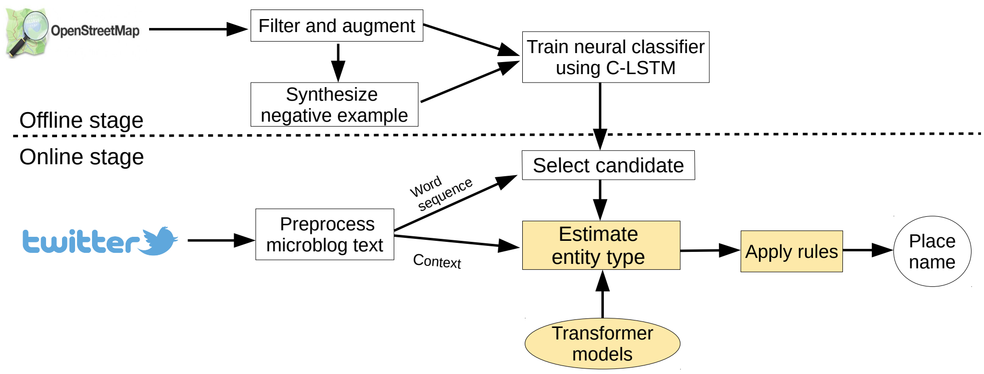

Figure 1 Workflow of our proposed place name extraction approach (GazPNE2).

The workflow of the proposed approach is shown in Figure 1. It consists of two main stages: offline and online. The offline stage is to train a classifier based on gazetteers such that it can recognize unseen multi-word place names. Specifically, we obtain and augment positive examples from a gazetteer, such as to generate *'east studemont rd'* from *'east studemont road'* by replacing a word (*'road'*) with its abbreviation (*'rd'*). We then synthesize negative examples from the positive ones in a rule-based fashion, such as to extract the sub set (e.g., *'City of'*) of a place name (e.g., *'City of New York'*). Next, we train a neural classifier with the C-LSTM [13] architecture based on the positive and negative examples. The online stage consists of two steps. The first step is to select candidates using the trained classifier. Specifically, a microblog text is first preprocessed by tokenizing the text, tagging the Part-of-Speech (POS) of tokens, and selecting valid n-grams by a simple POS rule. Then, the neural classifier is applied to classify the valid n-grams and the top non-overlapping n-grams with the highest positive probability are selected as the candidate place names. The second step is to disambiguate the candidates produced in the first step using two pretrained transformer models and features based on the context given in the microblog. While the offline stage was originally presented in [5], this work extends the disambiguation stage of the previously proposed extractor to substantially improve the overall extraction performance.

3 Place Name Disambiguation

The detections of the classifier which was trained on gazetteers require disambiguation based on contexts, since the entities it detects may be of a different entity type (*'Washington'* was also a person). We propose utilizing BERT [4] and BERTweet [8] models for disambiguation. BERT has previously been used for unsupervised named entity disambiguation [10], which inspired the idea of this study. Our proposed disambiguation stage consists of four steps.

Table 1 Examples of proposed method for disambiguation. Bold texts denote the candidate place names detected by the classifier. P, L, and O denote *Person*, *Location*, and non-type, respectively.

Tweet	Masked Sentence	Alternatives	Type	Prob	Result
#**Trump** landing his plane in LA	Trump is a <mask>	[President, Person, Leader, Village]	[P, P, P, L]	[L:0.25, P:0.75]	invalid
	# <mask> landing his plane in LA	[President, He Trump, Obama]	[P, P, P, P]	[L:0, P:1]	
Storm near 8 Miles E of **Clinton** moving NE	Clinton is a <mask>	[President, Leader, Artist, Town]	[P, P, P, L]	[L:0.25, P:0.75]	valid
	Storm near 8 Miles E of <mask> moving NE	[Houston, Texas, LA, Louisiana]	[L, L, L, L]	[L:1]	
I am stuck on **I 290**	I 290 is a <mask>	[song, comet, band, highway]	[O, O, O, L]	[L:0.25]	valid
	I am stuck on <mask>	[bridge, road, street, traffic]	[L, L, L, O]	[L:0.75]	

(1) **Word-entity-type dictionary creation.** For each word in the BERT vocabulary, we first calculate the cosine similarity of the word vectors between the word and the representative word of 6,111 annotated clusters. The clusters were generated in [10] by clustering the words in BERT by using the cosine similarity between the word vectors in BERT's word embedding space. Each cluster was then assigned with a type (e.g., *Person* and *Location*) manually, which took five man-hours in total. Then, we count the entity type of top-K neighboring clusters of the word and the proportion of a certain type is treated as the prior probability of the word being of the type. We name the dictionary that assigns an entity type with a prior probability to each word word-entity-type dictionary.

(2) **Semantic expansion**. The second step expands each candidate place name by retrieving alternative words from the semantic context. These alternatives are retrieved by first constructing two sentences based on intrinsic and extrinsic features of the candidate, respectively, with each containing the candidate and a *'<mask>'*, and subsequently predicting the mask with BERT and BERTweet, respectively, as shown in Table 1. Intrinsic and extrinsic features denote the candidate itself and its context in texts, respectively.

(3) **Entity type estimation.** Equation 1 shows how to calculate the probability of a candidate place name being of a certain entity type T.

$$p(T) = \sum_{i=1}^{n} \frac{(t_i \equiv T) \cdot s_i}{\sum_{i=1}^{n} s_i} \tag{1}$$

Here, n denotes the size of the top-n (set to 40 in this study) alternative (predicted) words, s_i denotes BERT's or BERTweets' confidence scores for each alternative word, and t_i denotes the most likely entity-prior for each alternative word. $t_i \equiv T$ is a Boolean expression, denoting if t_i equals T. For simplicity, we name the entity type probability calculated based on intrinsic and extrinsic features as intrinsic probability and extrinsic probability, respectively. Note that, if the candidate has only one word and is in the BERT's vocabulary, its intrinsic probability is obtained directly from the word-entity dictionary. To simplify the presentation of Table 1, we assume that the intrinsic probability of all the candidates is estimated by requesting BERT.

(4) **Rules application.** In the last step, the following rules are applied sequentially to decide if a candidate place name in a text is a valid location or not.

R1. **Reject person entities:** Reject the one-word candidate (e.g., *'Trump'*) if all tokens of one of its parental sequences (e.g., *'Donald Trump'*) are proper noun and if the intrinsic probability of the sequence of *Person* surpasses a threshold (set to 0.6) and if the extrinsic likelihood of the candidate of *Person* is larger than that of *Location*.

R2. **Accept abbreviations and location with numbers:** Accept the candidate as a location if the candidate contains numbers or it is a one-word abbreviation (e.g., *'uk'*) and if the extrinsic probability of *Location* surpasses a certain threshold (set to 0.2).

R3. **Accept likely locations:** Accept the candidate if the sum of the extrinsic and intrinsic probability of *Location* surpasses a certain threshold (set to 0.5) and is the largest among the total types. Accept the candidate if the extrinsic probability of *Location* surpasses a certain threshold (set to 0.3) and is the largest among the total types. For instance, in Table 1, *'Trump'* and *'Clinton'* are candidates and have a low intrinsic probability of *Location*. However, *'Trump'* and *'Clinton'* are still correctly recognized as invalid and valid place names respectively.

4 Experiments

4.1 Data preparation

We collect 18 million positive examples (place names) and 590 million negative examples to train a neural classifier. For English-speaking countries, we retrieve all the place names in OSMNames, which lists the place names derived from OpenStreetMap. The place names include coarse and fine-grained places, such as city and street, and abbreviation of places at country and state levels (e.g., *'tx'* for *'Texas'*). For the remaining non-English-speaking countries, we retrieve the place name at country, state, city, county, and town levels since the English names at these levels are provided, such as *'Munich'* for *'München'*, and the abbreviations of places at country levels, such as *'de'* for *'Germany'*.

We evaluate our approach on 11 public datasets. Those include five Location Extraction (LE) datasets, denoted by a, b, c, d, and e, respectively and six Name Entity Recognition (NER) datasets [3], denoted by f, g, h, i, j, and k, respectively. The five LE datasets correspond to three flood-related datasets [1], one hurricane-related dataset [12], and GeoCorpora [2]. The LE datasets only annotate *Location* while the NER datasets annotate *Location*, *Person*, and *Organization*. Table 2 summarizes the datasets.

Table 2 Number of tweets and places in the 11 test datasets in thousands.

	a	b	c	d	e	f	g	h	i	j	k	Total
Tweet Count	1.5k	1.5k	1.5k	1k	6.6k	2k	0.2k	2k	2.1k	2k	1k	21.4k
Place Count	2.3k	3k	3.7k	2.1k	3.1k	0.2k	0.1k	0.6k	1.3k	0.3k	0.1k	16.8k

4.2 Results

We compare GazPNE2 with 9 competitive approaches. They are Google NLP [3], Stanza [9] , OpenNLP [7], CLIFF [4], NeuoTPR [12], Spotlight [6], TwitIE-Gate [2], and OSU Twitter

[2] https://github.com/geovista/GeoCorpora
[3] https://cloud.google.com/natural-language/
[4] https://cliff.mediacloud.org/

NLP [11]. We adopt standard comparison metrics: Precision (P), Recall (R), and F1-Score (F). The results of different approaches are shown in Table 3. GazPNE2 achieves the best average F1-score of 0.8. GazPNE2 achieves the best F1 on 5 of 5 LE datasets. GazPNE2 achieves the best F1 on 3/6 NER datasets because of the different definition of *Location*. For instance, in the text, *'Louisiana police is helping rescue people affected by flood'*, LE datasets would tag *'Louisiana'* as *Location* while NER datasets would tag it as *Organization*. Many such cases exist in the NER datasets, causing a low F1.

Table 3 Tagging results of different place name extractors. The first column denotes the 11 test datasets. P, R, and F denote precision, recall, and F1-score, respectively. Bold and underline texts denote the best and second-best results, respectively.

		Google NLP	Spotlight	Stanza	Cliff	Open NLP	OSU NLP	TwitIE -Gate	Neuro -TPR	Geoparsepy	GazPNE2
a	P	0.40	0.41	0.43	**0.93**	0.41	0.82	0.40	0.43	0.42	0.92
	R	0.78	0.71	0.77	0.73	0.62	0.59	0.74	0.83	0.78	**0.85**
	F	0.50	0.52	0.55	0.82	0.50	0.69	0.52	0.57	0.55	**0.88**
b	P	0.40	0.60	0.61	0.88	0.63	0.67	0.54	0.64	0.57	**0.90**
	R	0.65	0.48	0.65	0.43	0.40	0.30	0.40	0.65	0.50	**0.71**
	F	0.49	0.53	0.63	0.58	0.49	0.41	0.46	0.64	0.53	**0.80**
c	P	0.43	0.67	0.53	0.89	0.37	0.77	0.55	0.68	0.31	**0.93**
	R	0.62	0.52	0.54	0.33	0.09	0.25	0.28	0.56	0.07	**0.80**
	F	0.51	0.58	0.53	0.48	0.15	0.38	0.37	0.61	0.11	**0.86**
d	P	0.56	0.73	0.66	**0.87**	0.65	0.63	0.64	0.80	0.43	0.83
	R	0.72	0.30	0.66	0.35	0.30	0.23	0.32	0.71	0.60	**0.81**
	F	0.63	0.42	0.66	0.50	0.41	0.34	0.43	0.75	0.50	**0.82**
e	P	0.29	0.43	0.41	**0.81**	0.42	0.64	0.44	0.50	0.18	0.75
	R	**0.79**	0.55	0.75	0.63	0.44	0.40	0.66	0.75	0.45	0.77
	F	0.43	0.48	0.53	0.71	0.43	0.50	0.53	0.60	0.26	**0.76**
f	P	0.17	0.28	0.26	**0.69**	0.19	0.57	0.27	0.35	0.18	0.47
	R	0.66	0.62	0.58	0.51	0.27	0.41	0.66	**0.81**	0.45	0.74
	F	0.27	0.38	0.36	**0.59**	0.22	0.48	0.39	0.49	0.26	0.58
g	P	0.16	0.22	0.25	**0.69**	0.22	0.48	0.25	0.30	0.23	0.63
	R	0.66	0.52	0.62	0.54	0.37	0.34	0.60	0.74	0.54	**0.82**
	F	0.25	0.31	0.35	0.60	0.28	0.40	0.36	0.43	0.32	**0.71**
h	P	0.25	0.38	0.31	**0.77**	0.26	**0.77**	0.39	0.42	0.37	0.67
	R	**0.83**	0.63	0.78	0.67	0.33	0.40	0.72	0.76	0.61	0.63
	F	0.39	0.48	0.44	**0.72**	0.29	0.54	0.51	0.54	0.46	0.65
i	P	0.28	0.40	0.34	**0.84**	0.33	0.62	0.38	0.47	0.36	0.71
	R	0.74	0.49	0.67	0.47	0.37	0.32	0.56	**0.75**	0.54	0.74
	F	0.40	0.44	0.45	0.60	0.35	0.43	0.46	0.58	0.43	**0.72**
j	P	0.37	0.54	0.48	**0.88**	0.43	0.76	0.50	0.60	0.48	0.66
	R	**0.79**	0.53	0.76	0.59	0.46	0.46	0.67	0.71	0.63	0.59
	F	0.50	0.54	0.59	**0.71**	0.44	0.57	0.57	0.65	0.55	0.62
k	P	0.26	0.28	0.35	**0.87**	0.30	0.61	0.32	0.44	0.27	0.57
	R	0.68	0.42	0.57	0.44	0.34	0.31	0.50	0.63	0.43	**0.77**
	F	0.37	0.33	0.43	0.59	0.32	0.41	0.39	0.52	0.33	**0.66**
ave	F	0.43	0.46	0.50	0.63	0.35	0.47	0.45	0.58	0.41	**0.80**

5 Conclusion

In this study, we propose a novel place name extractor for English tweets. It was compared with 9 competitive tools on 11 benchmark datasets, containing 21,393 tweets and 16,790 places across the globe. Our approach achieves the highest average F1 score of 0.8, proving the generality and robustness of our approach.

References

1 Hussein Al-Olimat, Krishnaprasad Thirunarayan, Valerie Shalin, and Amit Sheth. Location name extraction from targeted text streams using gazetteer-based statistical language models. *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1986–1997, August 2018. URL: `https://www.aclweb.org/anthology/C18-1169`.

2 Kalina Bontcheva, Leon Derczynski, Adam Funk, Mark A Greenwood, Diana Maynard, and Niraj Aswani. Twitie: An open-source information extraction pipeline for microblog text. In *Proceedings of the international conference recent advances in natural language processing RANLP 2013*, pages 83–90, 2013.

3 Leon Derczynski, Kalina Bontcheva, and Ian Roberts. Broad twitter corpus: A diverse named entity recognition resource. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1169–1179, 2016.

4 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

5 Xuke Hu, Hussein Al-Olimat, Jens Kersten, Matti Wiegmann, Friederike Klan, Yeran Sun, and Hongchao Fan. Gazpne: Annotation-free deep learning for place name extraction from microblogs leveraging gazetteer and synthetic data by rules. *International Journal of Geographical Information Science*, pages 1–28, 2021. `doi:10.1080/13658816.2021.1947507`.

6 Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8, 2011.

7 Stuart E Middleton, Giorgos Kordopatis-Zilos, Symeon Papadopoulos, and Yiannis Kompatsiaris. Location extraction from social media: Geoparsing, location disambiguation, and geotagging. *ACM Transactions on Information Systems (TOIS)*, 36(4):1–27, 2018.

8 Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. Bertweet: A pre-trained language model for english tweets. *arXiv preprint arXiv:2005.10200*, 2020.

9 Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020. URL: `https://nlp.stanford.edu/pubs/qi2020stanza.pdf`.

10 Ajit Rajasekharan. Unsupervised ner using bert, 2020. URL: `https://handsonnlpmodelreview.quora.com/Unsupervised-NER-using-BERT`.

11 Alan Ritter, Sam Clark, Oren Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1524–1534, 2011.

12 Jimin Wang, Yingjie Hu, and Kenneth Joseph. Neurotpr: A neuro-net toponym recognition model for extracting locations from social media messages. *Transactions in GIS*, 2020.

13 Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. A c-lstm neural network for text classification. *arXiv preprint arXiv:1511.08630*, 2015.