Improving Toponym Resolution by Predicting Attributes to Constrain Geographical Ontology Entries

Anonymous ACL submission

Abstract

Geocoding is the task of converting location mentions in text into structured geospatial data. We propose a new prompt-based paradigm for geocoding, where the machine learning algorithm encodes only the location mention and its context. We design a transformer network for predicting the country, state, and feature class of a location mention, and a deterministic algorithm that leverages the country, state, and feature class predictions as constraints in a search for compatible entries in the ontology. Our proposed architecture, GeoPLACE, achieves new state-of-the-art performance on multiple datasets. Code and models are available at `https://<anonymized>`.

1 Introduction

Geocoding is the task of matching locations in text to geospatial coordinates or entries in a geographical database. Geocoding systems support document categorization and retrieval (Bhargava et al., 2017), historical event analysis (Tateosian et al., 2017), monitoring the spread of infectious diseases (Hay et al., 2013), and disaster response mechanisms (Ashktorab et al., 2014; de Bruijn et al., 2018). Geocoding is challenging because identical place names may refer to different geographical locations (e.g., *San Jose* in Costa Rica vs. *San Jose* in California, USA), while distinct names can represent the same geographical location (e.g., *Leeuwarden* and *Ljouwert* in the Netherlands).

The traditional paradigm for geocoding systems is to train machine learning algorithms that encode both the location mention and the geographical ontology entries when predicting a label for the mention. For example, CamCoder's (Gritta et al., 2018) convolutional network encodes the location mention, the nearby context, and a population vector derived from the ontology entries, while GeoNorm's (Zhang and Bethard, 2023) transformer network encodes the location mention, the document level context, and the alternative names, location hierarchy, and population from the ontology.

We propose an alternative prompt-based approach to geocoding, where the machine learning algorithm at prediction time needs to encode only the location mention and its context, not the ontology. Rather than directly predicting ontology entries, our machine learning algorithm predicts attributes of ontology entries, such as the enclosing country and state. For example, our approach would predict that *Paris* in a document about Texas would have the attributes *"a Populated Place located in Texas in the United States."* The constraints implied by these predictions are used to deterministically filter the ontology entries. Our novel architecture, GEOgraphical normalization by Predicting Location Attributes to Constrain ontology Entries (GeoPLACE) is illustrated in Figure 1.

Our work makes the following contributions:

- We introduce a new paradigm for geocoding, where the machine learning algorithm encodes only the location mention and context.
- We design a transformer network for predicting the country, state, and feature class of a location mention, based on a masked language modeling objective and a novel prompt including all location mentions in the document.
- We introduce a novel deterministic algorithm that leverages the country, state, and feature class predictions as constraints in a search for compatible entries in the ontology.
- Our proposed approach achieves new state-of-the-art performance on multiple datasets.

2 Related Work

Prior work in geocoding included models based on hand-crafted rules and heuristics (Grover et al., 2010; Tobin et al., 2010; Lieberman et al., 2010; Lieberman and Samet, 2011; Berico Technologies, 2012; Karimzadeh et al., 2013), and tradi-

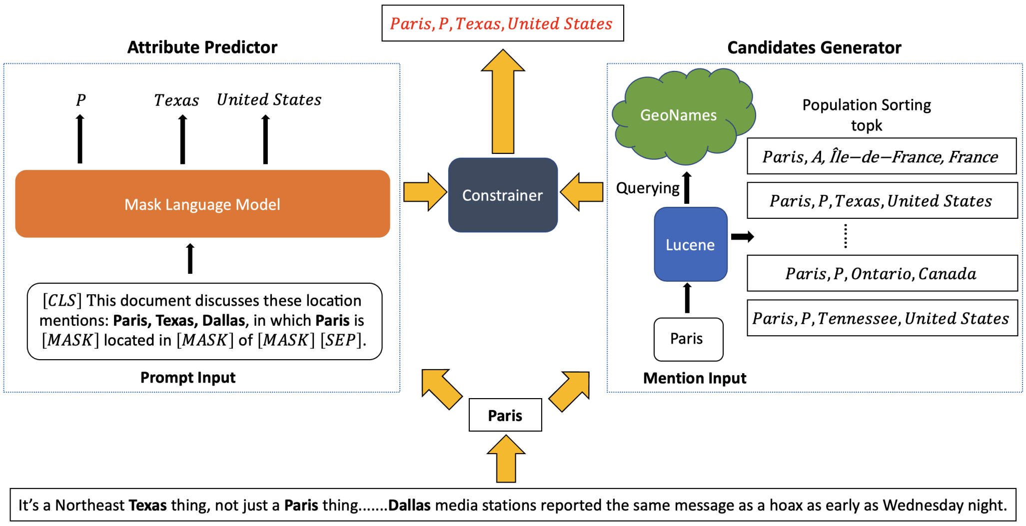

Figure 1: The architecture of our model: GEOgraphical normalization by Predicting Attributes to Constrain Ontology Entries (GeoPLACE). The figure shows how GeoPLACE normalizes a mention of *Paris*.

tional machine learning models such as support vector machines (Martins et al., 2010; Freire et al., 2011; Lieberman and Samet, 2012; Speriosu and Baldridge, 2013; Zhang and Gelernter, 2014; DeLozier et al., 2015; Kamalloo and Rafiei, 2018; Wang et al., 2019). However, most recent approaches to geocoding use neural networks.

Neural network based models have approached geocoding both as a ranking problem, trying to sort ontology entries by their appropriateness as a label for a location mention (Hosseini et al., 2020; Ardanuy et al., 2020; Ayoola et al., 2022; Zhang and Bethard, 2023) and as a classification problem, trying to map a location mention directly to one of an $N \times N$ grid of tiles covering the Earth's surface (Gritta et al., 2018; Cardoso et al., 2019; Kulkarni et al., 2021). The most successful approaches encode not just the mention and ontology entry names, but also context around the mention and information from the ontology such as population (Gritta et al., 2018; Ayoola et al., 2022; Zhang and Bethard, 2023). Many neural architectures have been considered, including convolutional (Gritta et al., 2018; Kulkarni et al., 2021), recurrent (Cardoso et al., 2019), and transformer networks (Ayoola et al., 2022; Zhang and Bethard, 2023).

In contrast to these approaches, we predict geographical attributes (e.g., enclosing state) and use those to deterministically select an ontology entry.

3 Proposed Methods

The problem of geocoding can be formalized as defining a function $f(m|T, M, E) = \hat{e}$ where T is the text of a document, M is the set of location mentions in the document, E is the set of geographical database entries, $m \in M$ is the mention under consideration, and $\hat{e} \in E$ is the entry predicted by f for m. In our paradigm for geocoding, we formulate f to first predict the country, state, and feature of m, next query the ontology with m to find candidate entries, then select the entry that violates the fewest constraints implied by the predicted attributes as the prediction $\hat{e}$. Formally:

$$\begin{aligned}\hat{C}_m, \hat{S}_m, \hat{F}_m &= \textsc{AttributePredictor}(m, M) \\ \hat{E} &= \textsc{CandidateGenerator}(m, E) \\ f(m|T, M, E) &= \textsc{Constrainer}(\hat{E}, \hat{C}_m, \hat{S}_m, \hat{F}_m)\end{aligned}$$

where C_m, S_m, F_m are the lists of predicted countries, states, and feature classes for m, and AttributePredictor, CandidateGenerator, and Constrainer are defined in the following sections.

We leverage the best CandidateGenerator from prior work, and implement new solutions for the newly introduced elements of our geocoding paradigm, AttributePredictor and Constrainer.

3.1 Attribute Predictor

This function predicts the country, state, and feature class of m using a model that combines prompting and a masked language modeling objective. The prediction targets are defined as:

Feature Class is one of the nine types defined by GeoNames: A, Administrative boundaries (e.g., countries, states, provinces); P, Populated places (e.g., cities, towns, villages); U, Undersea features (e.g., oceanic ridges, trenches), etc.

State is the canonical name of one of the 3871 first-order administrative divisions in GeoNames, such as states, provinces, or regions.

Country is the canonical name of one of the 252 countries in GeoNames.

We implement prediction of these targets as:

$$Z = \textsc{Transformer}(\textsc{ToInput}(m, M))$$
$$\hat{C}_m = \text{softmax}(Z_c W_c)$$
$$\hat{S}_m = \text{softmax}(Z_s W_s)$$
$$\hat{F}_m = \text{softmax}(Z_f W_f)$$

where $\textsc{ToInput}$ produces text of the form `[CLS] This document discusses these location mentions:` m_1, m_2, `...`, $m_{|M|}$ `in which` m `is [MASK] located in [MASK] of [MASK] [SEP]`, f, s, c, are the indexes of the three `[MASK]` tokens, $W_c, W_s, W_f \in \mathbb{R}^{N \times H}$ are the learnable parameters of the three classification heads, N is the size of the transformer tokenizer's vocabulary, and H is the size of the transformer's contextualized representations. We add new tokens to the transformer's tokenizer to ensure every country, state, and feature class is a single token, e.g., *United States*.

The model is trained with the classification loss:

$$L = C_m log(\hat{C}_m) + S_m log(\hat{S}_m) + F_m log(\hat{F}_m)$$

where C_m, S_m, and F_m are one-hot vectors of size N representing the true country, state, and feature class for mention m. At prediction time, we constrain the outputs of the softmax to the subset of the vocabulary appropriate for each prediction type. For example, when the model predicts the word for the country `<mask>`, only the 252 country names are allowed to be non-zero.

We train this model on the labeled data in the toponym datasets. Optionally, we also pre-train (before the fine-tuning) on additional data that we synthesize directly from the GeoNames ontology following the prompt format of $\textsc{ToInput}$. See appendix A.2 for details.

3.2 Candidate Generator

We adopt the candidate generator of Zhang and Bethard (2023), which outperformed prior candidate generators and some end-to-end systems. It uses Lucene to index GeoNames entries by their canonical and alternative names, selects entries for a mention by applying a series of searches including exact string matching and character 3-gram matching, and sorts the resulting entries by their population in GeoNames to place most populous countries at the top of the list.

Algorithm 1: Constrained Entry Selection

Input: a list of candidate entries, $\hat{E}_m$
top 3 predicted countries, $\hat{C}_m$
top 3 predicted states, $\hat{S}_m$
top 3 predicted feature classes, $\hat{F}_m$
Output: selected candidate entry $\hat{e}$

Def $\textsc{Score}(x, L)$**:**
  if $x = L_0$ **then return** *2*
  else return $x \in L$
Def $\textsc{EntryKey}(e)$**:**
  $c \leftarrow \textsc{Country}(e)$
  $s \leftarrow \textsc{State}(e)$
  $f \leftarrow \textsc{Feature}(e)$
  $key_1 \leftarrow \textsc{Score}(c, \hat{C}_m) \cdot \textsc{Score}(s, \hat{S}_m)$
  $key_2 \leftarrow (c \in \hat{C}_m) \cdot (s \in \hat{S}_m) \cdot \textsc{Score}(f, \hat{F}_m)$
  return $(key_1, key2)$
return $\textsc{Max}(\hat{E}_m, \textsc{key} = \textsc{EntryKey})$

3.3 Constrainer

Algorithm 1 defines our process for sorting the output of the candidate generator (entries) using the output of the attribute predictor (countries, states, and feature classes). We define the $\textsc{Score}$ of a prediction as 2 if it was the top ranked prediction, 1 if it was the second or third ranked prediction, and 0 otherwise. Entries are then sorted by the product of the country and state $\textsc{Score}$s, with the $\textsc{Score}$ of the feature class used to break ties. Intuitively, if the attribute predictor predicts C and S as the most probable country and state, then the constrainer will rank entries from GeoNames that are within country C and state S higher than other entries. We use a stable sort, so candidates that are assigned the same score retain their population-based sorting from the candidate generator.

See appendix A.3 for an illustration of the algorithm and evaluation of several variants.

4 Experiments

We conduct primary experiments on three toponym resolution datasets: Local Global Lexicon (LGL; Lieberman et al., 2010), a collection 588 news articles from local and small U.S. news sources; GeoWebNews (Gritta et al., 2019) a collection of 200 articles from 200 globally distributed news sites; and TR-News (Kamalloo and Rafiei, 2018) a collection 118 articles from various global and local

Model	LGL (test)				GeoWebNews (test)				TR-News (test)			
	Acc	A161	Err	AUC	Acc	A161	Err	AUC	Acc	A161	Err	AUC
ReFinED (Ayoola et al., 2022)	.576	-	-	-	.658	-	-	-	.720	-	-	-
ReFinED (fine-tuned by Zhang and Bethard, 2023)	.786	-	-	-	.782	-	-	-	.858	-	-	-
Candidate Generator (Zhang and Bethard, 2023)	.606	.685	119	.263	.694	.774	92	.194	.716	.812	95	.169
GeoNorm (Zhang and Bethard, 2023)	.807	.824	46	.135	**.828**	.862	**55**	.114	.918	.933	34	.057
GeoPLACE (ours)	**.863**	**.894**	**21**	**.084**	.822	**.878**	57	**.112**	**.947**	**.957**	**18**	**.038**
GeoPLACE (-synthesized pre-training)	.851	.886	24	.093	.809	.864	63	.123	.904	.922	20	.062
GeoPLACE (-masking, +generative fine-tuned BART)	.633	.696	111	.250	.704	.776	92	.191	.727	.812	95	.167
GeoPLACE (-masking, +generative zero-shot GPT-3)	.733	.795	80	.176	.719	.811	85	.171	.830	.869	63	.115

Table 1: Performance on the test sets. Higher is better for accuracy (Acc) and accuracy@161km (A161). Lower is better for mean error (Err) and area under the error distances curve (AUC). We do not report distance-based metrics for ReFinED as this extraction+disambiguation system does not make predictions for all mentions. The best performance on each dataset+metric is in bold.

news sources. All datasets use as their ontology GeoNames, a crowdsourced database of almost 7 million entries that contains geographic coordinates (latitude and longitude), alternative names, feature class (country, city, river, mountain, etc.), population, elevation, and positions within a political geographic hierarchy. See appendix A.1 for statistics of the datasets.

We adopt the train, development, and test splits and evaluation metrics of prior work (Zhang and Bethard, 2023). We refer the reader to that paper for details, but briefly, *accuracy* measures how often the correct database entry was predicted, while *accuracy@161km*, *mean error distance*, and *area under the curve* all give some partial credit for predicting entries that are wrong but geographically close to the correct entry.

We compare to the state-of-the-art geocoders:

ReFinED In Ayoola et al. (2022), transformer-generated embeddings for tokens in the text are matched to embeddings of ontology entries by by comparing their dot products. ReFinED is an end-to-end model originally trained on Wikipedia, but Zhang and Bethard (2023) leveraged the existing links to GeoNames IDs to fine-tune it for just the toponym resolution.

GeoNorm Zhang and Bethard (2023) uses Lucene to index and generate candidate entries from the ontology, applies a transformer network jointly over the mention and each candidate entry to predict a single entry, and applies a two-stage process to first resolve countries and states and use them as context to resolve other mentions.

Before evaluating on the test sets, we performed model selection on the development sets as described in appendix A.3.

5 Results

The top of table 1 compares our model to the existing state-of-the-art on LGL, GeoWebNews, and TR-News. (See appendices A.4 to A.6 for comparisons against other models and results on other datasets.) GeoPLACE outperforms prior work by large margins (more than 30% error reduction) on LGL and TR-News, while achieving similar performance on GeoWebNews. See appendix A.7 for a qualitative analysis of GeoPLACE prior work.

The bottom of table 1 shows an ablation of our model. Pre-training on synthesized data provides small but consistent gains across all datasets. We also try replacing our masked language modeling objective with a sequence-to-sequence style generative objective, asking the model to directly produce `is <feature-type> located in <state> of <country>`. (See appendix A.2 for prompting details.) This approach yields worse performance than our masked language modeling approach both when fine-tuning BART-large and when using GPT3 in zero-shot mode.

We release our model for English geocoding under the Apache License v2.0, for off-the-shelf use at `https://<anonymized>`.

6 Conclusion

We introduced a new paradigm for geocoding where instead of trying to map directly from text to an ontology entry, we predict geographical attributes and use those to deterministically constrain the set of valid ontology entries. Our approach leads to large error reduction over the current state-of-the-art on the LGL and TR-News datasets.

7 Limitations

The possible space of prompts is large, and while our location-based prompt worked well with our masked language modeling objective, it did not work well for generative models like BART. It is possible that more intensive exploration of alternative prompts could bring the performance of these generative models up to the performance of our masked language model. We also only explored zero-shot approaches for GPT-3, and though full fine-tuning BART did not yield acceptable performance, it is possible that few-shot approaches or fully fine-tuning GPT-3 would.

GeoPLACE is limited by its training and evaluation data, which covers only thousands of English toponyms from news articles, while there are many millions of toponyms across the world. It is likely that there are regional differences in GeoPLACE's accuracy that will need to be addressed by future research.

GeoPLACE is currently limited to geocoding. To apply this approach to other entity linking problems, one would need to identify the attributes that help constrain the search from the ontology, and then explore a few definitions of keys as we have in appendix A.3. This would be an interesting area for future research.

References

Mariona Coll Ardanuy, Kasra Hosseini, Katherine McDonough, Amrey Krause, Daniel van Strien, and Federico Nanni. 2020. A deep learning approach to geographical candidate selection through toponym matching. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, pages 385–388.

Zahra Ashktorab, Christopher Brown, Manojit Nandi, and Aron Culotta. 2014. Tweedr: Mining twitter to inform disaster response. In *ISCRAM*, pages 269–272.

Tom Ayoola, Joseph Fisher, and Andrea Pierleoni. 2022. Improving entity disambiguation by reasoning over a knowledge base. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2899–2912, Seattle, United States. Association for Computational Linguistics.

Berico Technologies. 2012. Cartographic location and vicinity indexer (clavin).

Preeti Bhargava, Nemanja Spasojevic, and Guoning Hu. 2017. Lithium NLP: A system for rich information extraction from noisy user generated text on social media. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 131–139, Copenhagen, Denmark. Association for Computational Linguistics.

Ana Bárbara Cardoso, Bruno Martins, and Jacinto Estima. 2019. Using recurrent neural networks for toponym resolution in text. In *EPIA Conference on Artificial Intelligence*, pages 769–780. Springer.

Jens A de Bruijn, Hans de Moel, Brenden Jongman, Jurjen Wagemaker, and Jeroen CJH Aerts. 2018. Taggs: grouping tweets to improve global geoparsing for disaster response. *Journal of Geovisualization and Spatial Analysis*, 2(1):2.

Grant DeLozier, Jason Baldridge, and Loretta London. 2015. Gazetteer-independent toponym resolution using geographic word profiles. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2382–2388. AAAI Press.

Nuno Freire, José Borbinha, Pável Calado, and Bruno Martins. 2011. A metadata geoparsing system for place name recognition and resolution in metadata records. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pages 339–348.

Milan Gritta, Mohammad Taher Pilehvar, and Nigel Collier. 2018. Which Melbourne? augmenting geocoding with maps. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1285–1296, Melbourne, Australia. Association for Computational Linguistics.

Milan Gritta, Mohammad Taher Pilehvar, and Nigel Collier. 2019. A pragmatic guide to geoparsing evaluation. *Language Resources and Evaluation*, pages 1–30.

Claire Grover, Richard Tobin, Kate Byrne, Matthew Woollard, James Reid, Stuart Dunn, and Julian Ball. 2010. Use of the edinburgh geoparser for georeferencing digitized historical collections. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1925):3875–3889.

Andrew Halterman. 2017. Mordecai: Full text geoparsing and event geocoding. *The Journal of Open Source Software*, 2(9).

Simon I Hay, Katherine E Battle, David M Pigott, David L Smith, Catherine L Moyes, Samir Bhatt, John S Brownstein, Nigel Collier, Monica F Myers, Dylan B George, et al. 2013. Global mapping of infectious disease. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 368(1614):20120250.

Kasra Hosseini, Federico Nanni, and Mariona Coll Ardanuy. 2020. DeezyMatch: A flexible deep learning approach to fuzzy string matching. In *Proceedings of* *the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 62–69, Online. Association for Computational Linguistics.

Ehsan Kamalloo and Davood Rafiei. 2018. A coherent unsupervised model for toponym resolution. In *Proceedings of the 2018 World Wide Web Conference*, pages 1287–1296.

Morteza Karimzadeh, Wenyi Huang, Siddhartha Banerjee, Jan Oliver Wallgrün, Frank Hardisty, Scott Pezanowski, Prasenjit Mitra, and Alan M MacEachren. 2013. Geotxt: a web api to leverage place references in text. In *Proceedings of the 7th workshop on geographic information retrieval*, pages 72–73.

Sayali Kulkarni, Shailee Jain, Mohammad Javad Hosseini, Jason Baldridge, Eugene Ie, and Li Zhang. 2021. Multi-level gazetteer-free geocoding. In *Proceedings of Second International Combined Workshop on Spatial Language Understanding and Grounded Communication for Robotics*, pages 79–88, Online. Association for Computational Linguistics.

Michael D Lieberman and Hanan Samet. 2011. Multifaceted toponym recognition for streaming news. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 843–852.

Michael D Lieberman and Hanan Samet. 2012. Adaptive context features for toponym resolution in streaming news. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 731–740.

Michael D Lieberman, Hanan Samet, and Jagan Sankaranarayanan. 2010. Geotagging with local lexicons to build indexes for textually-specified spatial data. In *2010 IEEE 26th international conference on data engineering (ICDE 2010)*, pages 201–212. IEEE.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2021. Self-alignment pretraining for biomedical entity representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4228–4238.

Bruno Martins, Ivo Anastácio, and Pável Calado. 2010. A machine learning approach for resolving place references in text. In *Geospatial thinking*, pages 221–236. Springer.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Michael Speriosu and Jason Baldridge. 2013. Text-driven toponym resolution using indirect supervision. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1466–1476.

Laura Tateosian, Rachael Guenter, Yi-Peng Yang, and Jean Ristaino. 2017. Tracking 19th century late blight from archival documents using text analytics and geoparsing. In *Free and open source software for geospatial (FOSS4G) conference proceedings*, volume 17, page 17.

Richard Tobin, Claire Grover, Kate Byrne, James Reid, and Jo Walsh. 2010. Evaluation of georeferencing. In *proceedings of the 6th workshop on geographic information retrieval*, pages 1–8.

Jimin Wang and Yingjie Hu. 2019. Enhancing spatial and textual analysis with eupeg: An extensible and unified platform for evaluating geoparsers. *Transactions in GIS*, 23(6):1393–1419.

Xiaobin Wang, Chunping Ma, Huafei Zheng, Chu Liu, Pengjun Xie, Linlin Li, and Luo Si. 2019. Dm_nlp at semeval-2018 task 12: A pipeline system for toponym resolution. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 917–923.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wei Zhang and Judith Gelernter. 2014. Geocoding location expressions in twitter messages: A preference learning method. *Journal of Spatial Information Science*, 2014(9):37–70.

Zeyu Zhang and Steven Bethard. 2023. Improving toponym resolution with better candidate generation, transformer-based reranking, and two-stage resolution. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 48–60, Toronto, Canada. Association for Computational Linguistics.

A Appendix

A.1 Dataset details

The number of toponyms and articles in each of the splits of each of the datasets is shown in table A1.

A.2 Implementation details

We adopt the candidate reranker of Zhang and Bethard (2023). We implement the attribute predictor with the PyTorch[1] v1.7.0 APIs in Huggingface Transformers v2.11.0 (Wolf et al., 2020), using `bert-base`. We train with the AdamW optimizer, a learning rate of 5e-6, a maximum sequence length of 256 tokens, and a number of epochs of 40. When training, we use one NVIDIA A100 GPU with 40G memory and a batch size of 64. The total number of parameters in our model is 112M and the training time is about 0.15 hours.

When synthesizing data from the geographical ontology for pre-training, we filtered all of cities, states and countries with less than 100 population and take the entries from some other special feature classes, such as H (stream, lake), L (parks, area) and T (mountain, hill, rock). To construct the input for pre-training, we used the same prompt with finetuning and sampled a different number of locations within the same country as the document mentions. Most of the hyperparameters are same with finetuning just the batch size is 32 and training epochs is 10.

When using a generative sequence-to-sequence objective instead of a masked language modeling objective, we utilize `bart-large` with the PyTorch v2.0.0 APIs in Huggingface Transformers v4.11.3 (Wolf et al., 2020) and FAIRSEQ v0.12.2 (Ott et al., 2019). We train with the AdamW optimizer, a initial learning rate of 1e-5, a learning rate scheduler type of polynomial, a maximum sequence length of 1024 tokens, and the steps of training of 40000. When training, we use one NVIDIA A100 GPU with 40G memory and a batch size of 8. During evaluation, we use beam search with a beam size of 5. The total number of parameters in our model is 406M and the training time is about 1.3 hours. We use one model to generate only one attribute, when we generate the country name, we use the prompt `[CLS] This document discusses these location mentions:` m_1, m_2, `...`, $m_{|M|}$. `Which country is START` m `END located ?`, the prefix prompt for output generation is m `is located in`. When we generate the state name, we use the prompt `[CLS] This document discusses these location mentions:` m_1, m_2, `...`, $m_{|M|}$. `Which state is START` m `END located ?`, the prefix prompt for output generation is m `is located in`. When we generate the feature class, we use the prompt `[CLS] This document discusses these location mentions:` m_1, m_2, `...`, $m_{|M|}$. `Which feature class does START` m `END belong to ?`, the prefix prompt for output generation is m `belong to`

[1] https://pytorch.org/

A.3 Model selection

For the attribute predictor, we explored a small number of learning rates (1e-6, 2e-6, 5e-6, 1e-5) and number of epochs (10, 20, 30, 40). The best learning rate and number of epochs was selected based on accuracy on the attribute prediction task (not on the full geocoding task).

For the constrainer, we explored three different ways to define key_1 and key_2.

alg3 defines key_1 and key_2 as in alg. 1.

alg2 allows scores to range from 0 to the length of the list, rather than just from 0 to 2. It defines:

$$key_1 \leftarrow \text{RINDEX}(c, \hat{C}_m) \cdot \text{RINDEX}(s, \hat{S}_m)$$
$$key_2 \leftarrow (c \in \hat{C}_m) \cdot (s \in \hat{S}_m) \cdot \text{RINDEX}(f, \hat{F}_m)$$

$$\textbf{Def } \text{RINDEX}(x, L): \textbf{ if } x \notin L \textbf{ then } 0 \quad \textbf{else } |L| - lst.index(val)$$

alg1 prioritizes matching the first country, and also allows scores to range from 0 to the length of the list. It defines:

$$key_1 \leftarrow (c = \hat{C}_{m_0}) \cdot \text{RINDEX}(s, \hat{S}_m)$$
$$key_2 \leftarrow (c \in \hat{C}_m) \cdot (s \in \hat{S}_m) \cdot \text{RINDEX}(f, \hat{F}_m)$$

Table A2 shows that there were not large differences between these algorithms in terms of accuracy, but alg3 performed slightly better.

For the constrainer, we also explored four different ways to define the number of predictions to consider in the constrainer.

top3 Only the top 3 countries, states, and feature classes are considered

top4 Only the top 4 countries, states, and feature classes are considered

top5 Only the top 5 countries, states, and feature classes are considered

Dataset	Train		Dev.		Test	
	Toponyms	Articles	Toponyms	Articles	Toponyms	Articles
LGL	3112	411	419	58	931	119
GeoWebNews	1641	140	281	20	477	40
TR-News	925	82	68	11	282	25

Table A1: Numbers of articles and manually annotated toponyms in the train, development, and test splits of the toponym resolution corpora.

	Accuracy		
Model	LGL (dev)	GeoWebNews (dev)	TR-News (dev)
GeoPLACE (alg1 top553)	.885	.811	.926
GeoPLACE (alg1 top553 synthesized pre-training)	.902	**.872**	.912
GeoPLACE (alg2 top553)	.885	.815	.926
GeoPLACE (alg2 top553 synthesized pre-training)	.902	**.872**	.912
GeoPLACE (alg3 top553)	.900	.815	.926
GeoPLACE (alg3 top553 synthesized pre-training)	.902	**.872**	.912
GeoPLACE (alg3 top555)	.900	.815	.926
GeoPLACE (alg3 top555 synthesized pre-training)	.902	**.872**	.912
GeoPLACE (alg3 top444)	.893	.815	**.941**
GeoPLACE (alg3 top444 synthesized pre-training)	**.912**	**.872**	.912
GeoPLACE (alg3 top333)	.893	.826	**.941**
GeoPLACE (alg3 top333 synthesized pre-training)	**.912**	.868	.926

Table A2: Model selection on the development sets. The top performance on each dataset is in bold, the second best performance is underlined.

top553 The top 5 countries, top 5 states, and top 3 feature classes are considered

Table A2 shows that there were not large differences between these strategies in terms of accuracy, but top3 performed slightly better.

For the constrainer, we also explored whether or not it helps to pre-train on synthesized data before fine-tuning on the toponym resolution datasets. Table A2 shows that pre-training on synthesized data consistently helped on LGL and GeoWebNews but led to small drops in performance on TR-News.

Figure 2 shows an example about how the alg3 top3 constrainer works.

A.4 EUPEG results

We also report results using the Extensible and Unified Platform for Evaluating Geoparsers (EUPEG; Wang and Hu, 2019). This platform evaluates not geocoders, but geoparsers, where a model must both detect locations and match them to ontology entries. So we couple our geocoder with the best location detection model on EUPEG, the Stanford-NER system.

This platform reports several metrics that are incomparable across systems. Accuracy, accuracy@161km, mean error, and area under the error distances curve are all calculated only over locations that were detected, so that a model that detects only 1% of locations but matches 100% of them to their correct ontology entries would get perfect values for these scores, while a model that detects 100% of locations and matches 90% of them to their correct ontology entries would score lower. We nonetheless report these incomparable metrics as EUPEG provides no alternative. EUPEG results are shown in table A3

A.5 Recall of Geographical Attributes Prediction

Table A4 shows the performance of the geographical attribute prediction classifiers alone, i.e., as classifiers rather than as components in a geocoding system. We report recall@3 since the constrainer considers the top 3 predictions of the attribute predictor. Performance across all datasets and all classifiers is 0.84 or higher.

A.6 Full table of Test Performance

Table A5 compares GeoPLACE to other systems that, due to space limitations, we could not include in table 1.

A.7 Qualitative Analysis

Table A6 presents a qualitative analysis of errors encountered by GeoNorm (Zhang and Bethard, 2023)

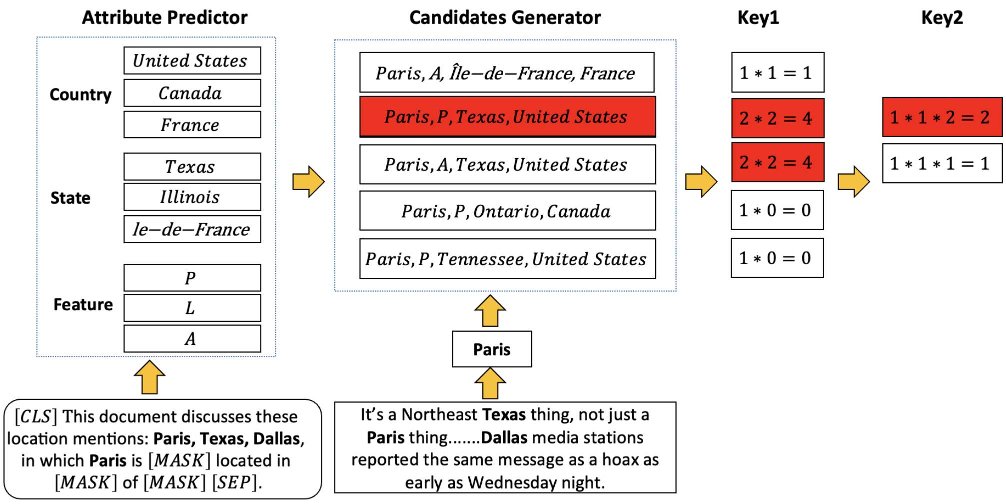

Figure 2: Illustration of the alg3 top3 constrainer applied to *Paris* in the context *It's a northeast Texas thing, not just a Paris thing. . . Dallas media stations reported the same message as a hoax as early as Wednesday night.*.

	LGL (test)						GeoWebNews (test)					
Model	Pre	Rec	F1	A161	Err	AUC	Pre	Rec	F1	A161	Err	AUC
Edinburgh	.776	.353	.486	.775	60	.187	.787	.520	.626	**.944**	33	**.056**
StanfordNER + Pop	.762	.635	.692	.592	135	.360	.866	.648	.741	.673	86	.257
StanfordNER + GeoPLACE	.762	.635	.692	**.888**	**23**	**.109**	.866	.648	.741	.929	**30**	.072
	TR-News (test)						GeoVirus					
Model	Pre	Rec	F1	A161	Err	AUC	Pre	Rec	F1	A161	Err	AUC
Edinburgh	.752	.592	.663	.844	78	.121	.860	.559	.678	.807	44	.319
StanfordNER + Pop	.906	.752	.822	.651	119	.287	.927	.903	.915	.655	79	.378
StanfordNER + GeoPLACE	.906	.752	.822	**.967**	**15**	**.033**	.927	.903	.915	**.837**	**23**	**.297**
	WikToR						GeoCorpora					
Model	Pre	Rec	F1	A161	Err	AUC	Pre	Rec	F1	A161	Err	AUC
Edinburgh	.230	.298	.259	.591	217	.378	.832	.505	.628	.848	96	.140
StanfordNER + Pop	.209	.540	.301	.184	460	.702	.899	.526	.664	.676	106	.270
StanfordNER + GeoPLACE	.209	.540	.301	**.629**	**171**	**.342**	.899	.526	.664	**.875**	**48**	**.122**
	Hu2014						Ju2016					
Model	Pre	Rec	F1	A161	Err	AUC	Pre	Rec	F1	A161	Err	AUC
Edinburgh	.486	.656	.559	**.114**	**86**	**.607**	.000	.000	.000	—	–	—
StanfordNER + Pop	.504	.788	.615	.000	228	.758	.162	.010	.019	0.0	**203**	**.743**
StanfordNER + GeoPLACE	.504	.788	.615	.071	92	.632	.162	.010	.019	**.046**	354	.768

Table A3: Performance on the test sets. Precision (Pre), Recall (Rec), and F1 are on the location detection task, while the other metrics are on the geocoding task Higher is better for accuracy (Acc) and accuracy@161km (A161). Lower is better for mean error (Err) and area under the error distances curve (AUC). The best performance on each dataset and geocoding metric is in bold.

Model	LGL (test)	GeoWebNews (test)	TR-News (test)
Country	.992	.932	.891
State	.929	.873	.849
Feature Class	.996	.944	.996

Table A4: Geographical Attribute Prediction Performance of Recall@3 on the test sets.

Model	LGL (test)				GeoWebNews (test)				TR-News (test)			
	Acc	A161	Err	AUC	Acc	A161	Err	AUC	Acc	A161	Err	AUC
Edinburgh (Grover et al., 2010)	.611	-	-	-	.738	-	-	-	.750	-	-	-
CamCoder (Gritta et al., 2018)	.580	.651	82	.288	.572	.665	155	.290	.660	.778	89	.196
Mordecai (Halterman, 2017)	.322	.375	926	.594	.291	.333	1072	.633	.472	.553	6558	.427
DeezyMatch (Hosseini et al., 2020)	.172	.182	654	.704	.262	.323	537	.601	.206	.220	741	.705
SAPBERT (Liu et al., 2021)	.245	.260	566	.630	.428	.499	357	.446	.355	.362	595	.568
ReFinED (Ayoola et al., 2022)	.576	-	-	-	.658	-	-	-	.720	-	-	-
ReFinED (fine-tuned by Zhang and Bethard, 2023)	.786	-	-	-	.782	-	-	-	.858	-	-	-
Candidate Generator (Zhang and Bethard, 2023)	.606	.685	119	.263	.694	.774	92	.194	.716	.812	95	.169
GeoNorm (Zhang and Bethard, 2023)	.807	.824	46	.135	**.828**	.862	**55**	.114	.918	.933	34	.057
GeoPLACE (ours)	**.863**	**.894**	**21**	**.084**	.822	**.878**	57	**.112**	**.947**	**.957**	**18**	**.038**

Table A5: Performance on the test sets. Higher is better for accuracy (Acc) and accuracy@161km (A161). Lower is better for mean error (Err) and area under the error distances curve (AUC). We do not report distance-based metrics for ReFinED as this extraction+disambiguation system does not make predictions for all mentions. The best performance on each dataset+metric is in bold.

and our latest state-of-the-art model, GeoPLACE.

The first row displays an example where GeoNorm falls short while GeoNorm excels. This can be attributed to GeoNorm's superior ability to employ masked language models for accurately predicting the countries, states, and feature codes of toponyms in the text prior to their resolution.

The second row portrays an instance where our most proficient model, GeoPLACE, experiences a failure. This occurs because predicting feature codes with the aid of a masked language model proves to be more challenging compared to predicting countries and states. Thoroughly resolving this problem is likely to necessitate improvements in the prediction performance for all types of geographical metadata.

Example	Candidate					Rank	
	Name	Pop.	Type	State	Country	GeoNorm	GeoPLACE
1 *But the Mt. Pleasant News has reviewed legal documents......he writes, as do my efforts to insure New London is a safe community.*	New London County	274055	ADM2	Connecticut	United States	1	2
	New London	27179	PPL	Connecticut	United States	2	3
	New London	7172	PPL	Wisconsin	United States	3	4
	New London	1882	PPL	Iowa	United States	4	1
2 *John-Paul Delaney (18), is charged with assault, assault causing harm and theft of a mobile phone at Main Street, Tipperary, on the same date.*	Tipperary	159553	ADM2	Munster	Ireland		1
	Tipperary	4979	PPL	Munster	Ireland		2
	Tipperary	0	HMSD	Western Australia	Australia		3
	Tipperary	0	HMSD	New South Wales	Australia		4

Table A6: Examples of predictions from GeoNorm (Zhang and Bethard, 2023) and our new SOTA model, GeoPLACE. Target location mentions are underlined. Human annotated ontology entries are in bold. (ADM2 represents a county, PPL represents a city, HMSD represents a residence specific to Australia and New Zealand)