

Improving Toponym Resolution with Better Candidate Generation, Transformer-based Reranking, and Two-Stage Resolution

Zeyu Zhang and Steven Bethard

School of Information, The University of Arizona, Tucson, AZ, USA

{zeyuzhang, bethard}@arizona.edu

Abstract

Geocoding is the task of converting location mentions in text into structured data that encodes the geospatial semantics. We propose a new architecture for geocoding, GeoNorm. GeoNorm first uses information retrieval techniques to generate a list of candidate entries from the geospatial ontology. Then it reranks the candidate entries using a transformer-based neural network that incorporates information from the ontology such as the entry’s population. This generate-and-rerank process is applied twice: first to resolve the less ambiguous countries, states, and counties, and second to resolve the remaining location mentions, using the identified countries, states, and counties as context. Our proposed toponym resolution framework achieves state-of-the-art performance on multiple datasets. Code and models are available at <https://github.com/clulab/geonorm>.

1 Introduction

Geospatial information extraction is a type of semantic extraction that plays a critical role in tasks such as geographical document classification and retrieval (Bhargava et al., 2017), historical event analysis based on location data (Tateosian et al., 2017), tracking the evolution and emergence of infectious diseases (Hay et al., 2013), and disaster response mechanisms (Ashktorab et al., 2014; de Bruijn et al., 2018). Such information extraction can be challenging because different geographical locations can be referred to by the same place name (e.g., *San Jose* in Costa Rica vs. *San Jose* in California, USA), and different place names can refer to the same geographical location (e.g., *Leeuwarden* and *Ljouwert* are two names for the same city in the Netherlands). It is thus critical to resolve these place names by linking them with their corresponding coordinates from a geospatial ontology or knowledge base.

Geocoding, also called toponym resolution or toponym disambiguation, is the subtask of geoparsing that disambiguates place names (known as *toponyms*) in text. Given a textual mention of a location, a geocoder chooses the corresponding geospatial coordinates, geospatial polygon, or entry in a geospatial database. Approaches to geocoding include generate-and-rank systems that first use information retrieval systems to generate candidate entries and then rerank them with hand-engineered heuristics and/or supervised classifiers (e.g., Grover et al., 2010; Speriosu and Baldridge, 2013; Wang et al., 2019), vector-space systems that use deep neural networks to encode place names and database entries as vectors and measure their similarity (e.g., Hosseini et al., 2020; Ardanuy et al., 2020), and tile-classification systems that use deep neural networks to directly predict small tiles of the map rather than ontology entries (e.g., Gritta et al., 2018a; Cardoso et al., 2019; Kulkarni et al., 2021). The deep neural network tile-classification approaches have been the most successful, but they do not naturally produce an ontology entry, which contains semantic metadata needed by users.

We propose a new architecture, GeoNorm, shown in Figure 1, which builds on all of these lines of research: it uses pre-trained deep neural networks for the improved robustness in matching place names, while leveraging a generate-then-rank architecture to produce ontology entries as output. It couples this generate-and-rank process with a two-stage approach that first resolves the less ambiguous countries, states, and counties, and then resolves the remaining location mentions, using the identified countries, states, and counties as context.

Our work makes the following contributions:

- Our proposed architecture for geocoding achieves new state-of-the-art performance, outperforming prior work by large margins on toponym resolution corpora: 19.6% improvement on Local Global Lexicon (LGL), 9.0%

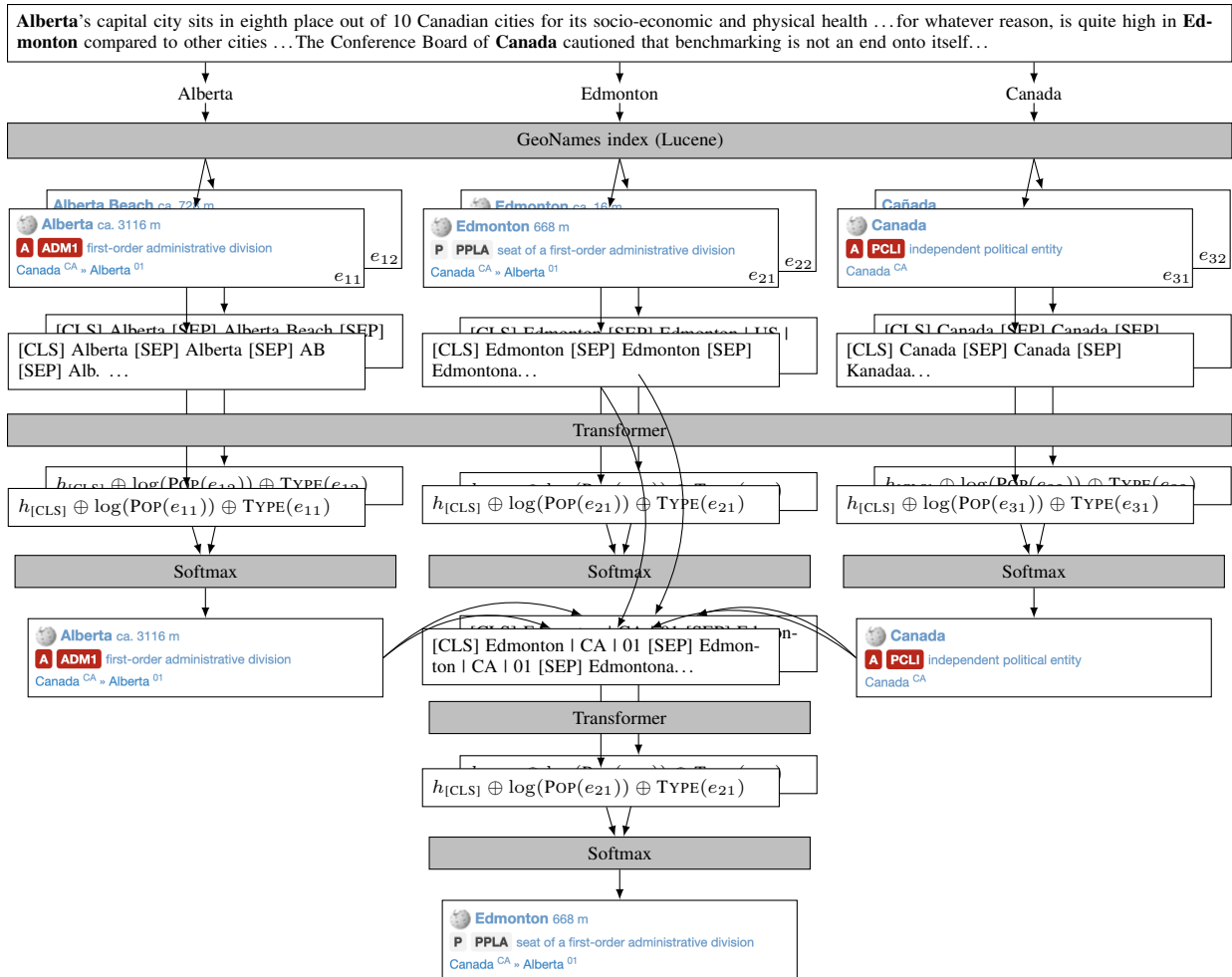


Figure 1: The architecture of our model, GeoNorm, applied to a sample text. The location mentions to be resolved are in bold.

on GeoWebNews, and 16.8% on TR-News.

- Our candidate generator alone, based on simple information retrieval techniques, outperforms more complex neural models, demonstrating the importance of establishing strong baselines for evaluation.
- Our reranker is the first application of pre-trained transformers for encoding location mentions and context for toponym resolution.
- Our two-stage resolution provides a simple and effective new approach to incorporating document-level context for geocoding.

2 Related Work

The current work focuses on mention-level geocoding. Related tasks include document-level geocoding and geotagging. Document-level geocoding takes as input an entire text and produces as output a location from a geospatial ontology, as in geolocating Twitter users or microblog posts (Roller

et al., 2012; Rahimi et al., 2015; Lee et al., 2015; Rahimi et al., 2017; Hoang and Mothe, 2018; Kumar and Singh, 2019; Luo et al., 2020) and geographic document retrieval and classification (Gey et al., 2005; Adams and McKenzie, 2018). Geotagging takes as input an entire text and produces as output a list of location phrases (Gritta et al., 2018b). Mention-level geocoding, the focus of the current article, takes as input location phrases from a text and produces as output their corresponding locations in a geospatial ontology. This is related to the task of linking phrases to Wikipedia, though geospatial ontologies do not have full text articles for each of their concepts, which are required for training many recent Wikipedia linking approaches (e.g., Yamada et al., 2022; Ayoola et al., 2022b).

Early systems for mention-level geocoding used hand-crafted rules and heuristics to predict geospatial labels for place names: Edinburgh geoparser (Grover et al., 2010), Tobin et al. (2010), Lieber-

man et al. (2010), Lieberman and Samet (2011), CLAVIN (Berico Technologies, 2012), GeoTxt (Karimzadeh et al., 2013), and Laparra and Bethard (2020). The most common features and heuristics were based on string matching, population count, and type of place (city, country, etc.).

Later geocoding systems used heuristics of rule-based systems as features in supervised machine learning models, including logistic regression (WISTR, Speriosu and Baldrige, 2013), support vector machines (Martins et al., 2010; Zhang and Gelernter, 2014), random forests (MG, Freire et al., 2011; Lieberman and Samet, 2012), stacked LightGBMs (DM_NLP, Wang et al., 2019) and other statistical learning methods (Topocluster, DeLozier et al., 2015; CBH, SHS, Kamaloo and Rafiei, 2018). These systems typically applied a generate-then-rerank framework: the mention text is used to query an information retrieval index of the geospatial ontology and produce candidate ontology entries, then a supervised machine-learning model reranks the candidates using additional features.

Some deep learning models approach geocoding as a vector-space problem. Both the mention text and ontology entries are converted into vectors, and vector similarity is used to select the most appropriate ontology entry for each mention (Hosseini et al., 2020; Ardanuy et al., 2020). Such approaches should allow more flexible matching of mentions to concepts, but we find that simple information retrieval techniques outperform these models.

Other deep learning models approach geocoding as a classification problem by dividing the Earth’s surface into an $N \times N$ grid of tiles. Place names and their features are mapped to one of these tiles using convolutional (CamCoder, Gritta et al., 2018a; MLG, Kulkarni et al., 2021) or recurrent neural networks (Cardoso et al., 2019). Such approaches can flexibly match mentions to concepts and can also incorporate textual context, but do not naturally produce ontology entries, which contain semantic metadata needed by users.

Our proposed approach combines the tight ontology integration of the generate-and-rerank systems with the robust text and context encoding of the deep neural network classifiers.

3 Proposed Methods

We define the task of toponym resolution as follows. We are given an ontology or knowledge base with a set of entries $E = \{e_1, e_2, \dots, e_{|E|}\}$.

Each input is a text made up of sentences $T = \{t_1, t_2, \dots, t_{|T|}\}$ and a list of location mentions $M = \{m_1, m_2, \dots, m_{|M|}\}$ in the text. The goal is to find a mapping function $f(m_i) = e_j$ that maps each location mention in the text to its corresponding entry in the ontology.

We approach toponym resolution using a candidate generator followed by a candidate reranker. The candidate generator, $G(m, E) \rightarrow E_m$, takes a mention m and ontology E as input, and generates a list of candidate entries E_m , where $E_m \subseteq E$ and $|E_m| \ll |E|$. As the candidate generator must search a large ontology and produce only a short list of candidates, the goal for G will be high recall and high runtime efficiency. The candidate reranker, $R(m, E_m) \rightarrow \widehat{E}_m$, takes a mention m and the list of candidate ontology entries E_m , and sorts them by their relevance or importance to produce a new list, \widehat{E}_m . As the candidate reranker needs to work only with a short list of candidates, the goal for R will be high precision, especially at rank 1, with less of a focus on runtime efficiency.

3.1 Candidate Generator

Our candidate generator is inspired by prior work on geocoding in using information retrieval techniques to search for candidates in the ontology (Grover et al., 2010; Berico Technologies, 2012). Accurate candidate generation is essential, since the generator’s recall is the ceiling performance for the reranker. As we will see in section 5, our proposed candidate generator alone is competitive with complex end-to-end systems from prior work.

Our sieve-based approach, detailed in alg. 1, tries searches ordered from least precise to most precise until we find ontology entries that match the location mention. Intuitively, our goal is for mentions like *Austria* to match the entry AUSTRIA [2782113] in GeoNames before it matches AUSTRALIA [2077456], but still allow a typo like *Australa* to match AUSTRALIA [2077456].

We create one document in the index for each name n_e of an entry e in the GeoNames ontology. A location mention m is matched to a name n_e by attempting a search with each of the following matching strategies, in order:

EXACT m exactly matches (ignoring whitespace) the string n_e

FUZZY m is within a 2 character Levenshtein edit distance (ignoring whitespace) of n_e

Algorithm 1: Candidate generator.

Input: a location mention, m
a maximum number of candidates, k
the GeoNames ontology, E
Output: a list of candidate entries E_m
// Index ontology
1 $I \leftarrow \emptyset$
2 **for** $e \in E$ **do**
3 $name \leftarrow \text{CANONICALNAME}(E, e)$
4 $synonyms \leftarrow \text{SYNONYMS}(E, e)$
5 **for** $n \in \{name\} \cup synonyms$ **do**
6 $I \leftarrow I \cup \{\text{CREATEDOCUMENT}(n, e)\}$
// Search for candidates
7 $E_m \leftarrow \emptyset$
8 **for** $t \in \{\text{EXACT, FUZZY, CHARACTERNGRAM, TOKEN, ABBREVIATION, COUNTRYCODE}\}$ **do**
9 $E_m \leftarrow \text{SEARCH}(I, m, t)$
10 **if** $E_m \neq \emptyset$ **then**
11 **break**
// Select top entries by population
12 $E_m \leftarrow \text{SORT}(E_m, \text{KEY} = e \rightarrow \text{POPULATION}(E, e))$
13 **return** top k elements of E_m

CHARACTERNGRAM m has at least one character 3-gram overlap with n_e

TOKEN m has at least one token (according to the Lucene StandardAnalyzer) overlap with n_e

ABBREVIATION m exactly matches the capital letters of n_e

COUNTRYCODE e is a country and m exactly matches a e 's country code

Once one of the searches has retrieved a list of matching names, we recover the ontology entry for each name, sort those ontology entries by their population in the GeoNames ontology, and return the k most populous ontology entries. This list, E_m is then the input to the candidate reranker.

3.2 Candidate Reranker

Our candidate reranker is inspired by work on medical concept normalization (Xu et al., 2020; Ji et al., 2020). The reranker takes a mention, m , and the list of candidate entities from the candidate generator, E_m , encodes them with a transformer network, and uses these encoded representations to perform classification over the list to select the most probable entry. Formally, the model prediction, $\text{GEONORM}(m, E_m) = \hat{e}$, is calculated as:

$$s^i = \text{TOINPUT}(m, E_m^i)$$

$$\mathbf{A}^i = \text{TRANSFORMER}(s^i)$$

$$\mathbf{b}^i = \mathbf{A}_0^i \oplus \log(\text{POP}(E, E_m^i)) \oplus \text{TYPE}(E, E_m^i)$$

$$c^i = (\mathbf{b}^i \mathbf{W}_1^T) \mathbf{W}_2^T$$

$$\hat{\mathbf{y}} = \text{softmax}(c^0 \oplus \dots \oplus c^k)$$

where:

- E_m^i is the i^{th} candidate entry for mention m
- $\text{TOINPUT}(m, e)$ produces a string of the form $[\text{CLS}] m [\text{SEP}] C(E, e) [\text{SEP}] S(E, e)_1 [\text{SEP}] \dots [\text{SEP}] S(E, e)_{|S(E, e)|} [\text{SEP}]$, where $C(E, e)$ is the canonical name of e in the ontology, and $S(E, e)$ is the list of alternate names of e in the ontology.
- $\text{TRANSFORMER}(s)$ tokenizes the string s into word-pieces and produces contextualized embeddings for each of the word-pieces.
- \mathbf{A}_0^i is the contextualized representation for the $[\text{CLS}]$ token of candidate entry i 's input string
- $\text{POP}(E, e)$ is the population of concept e in the ontology E
- $\text{TYPE}(E, e)$ is a one-hot vector identifying which of the T types in the ontology E the concept represents¹
- \oplus denotes vector concatenation
- $\mathbf{W}_1 \in \mathbb{R}^{150 \times (H+1+T)}$ and $\mathbf{W}_2 \in \mathbb{R}^{1 \times 150}$ are learned weight matrices, where H is the transformer's hidden dimension
- $\hat{\mathbf{y}}$ is a probability distribution over the k entries proposed by the candidate generator

We represent the mention text + candidate entity synonyms with the contextualized representation of the $[\text{CLS}]$ token, similar to applications of transformers to text classification. We include the population feature to allow the model to learn that locations in text are more likely to refer to high population than low population places (e.g., Paris, France vs. Paris, Texas, USA), and we take the logarithm of the population under the assumption that it is more important to capture the order of magnitude (e.g., thousands vs. millions) than the exact number. We include the type feature to allow the model to learn that locations in text are more likely to refer to some types of geographical features than others (e.g., San José, the capital of Costa Rica, vs. San José, the province).

The candidate reranker is trained with a standard classification loss:

$$L_R = \mathbf{y} \cdot \log(\hat{\mathbf{y}})$$

where $\mathbf{y} \in \mathbb{R}^{|E_m|}$ is a one-hot vector representing the correct candidate entry.

¹GeoNames has $T = 681$ types. For example, PPLC means *capital of a political entity*. Definitions for all types ("feature codes") are at http://download.geonames.org/export/dump/featureCodes_en.txt

3.3 Context Incorporation

The text around a mention may provide clues (e.g., the context *Minnesota State Patrol urges motorists to drive with caution... in Becker, Clay, and Douglas* suggests that *Clay* refers to Clay County, Minnesota, even though Clay County, Missouri is more populous). Thus, we consider two approaches to incorporating context.

context=csent A simple approach is to take the c -sentence window surrounding the mention m and encode it with the the same transformer as was used to encode $m + e$. The contextualized representation of the c -sentence window’s [CLS] token can then be concatenated into \mathbf{b} alongside the other features. The 512 word-piece limit on the size of the transformer input means that this approach cannot incorporate the entire document.

context=2stage To include the full document context, we take advantage of the fact (demonstrated in appendix A.1) that toponyms at the top of the hierarchy, like countries and states, can often be resolved precisely without context as they are less ambiguous. We thus propose Algorithm 2, a two-stage approach to geocoding. Lines 3-7 are the context-free stage, where GeoNorm is first applied to all location mentions. If the feature type of a predicted entry, $\text{TYPE}(e)$, is an administrative district 1-3 (i.e., the top of the geographic hierarchy: countries, states, or counties), then the prediction is accepted. Such predictions are converted to their administrative codes (e.g., *United States* \rightarrow US) and added to the context. Lines 8-11 are the second stage, where the geocoding system is applied to all remaining location mentions but this time incorporating the collected context. The context is formed by concatenating together the collected toponym codes, where for example, if Canada (CA) and Alberta (01) were found in the document as in fig. 1, the context string would look like “CA || 01”.

4 Experiments

4.1 Datasets

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

Algorithm 2: Two-stage toponym resolution using document-level context.

```

Input: location mentions,  $M$ 
         GeoNames ontology,  $E$ 
1  $\hat{R} \leftarrow \{\}$ 
2  $C \leftarrow \emptyset$ 
   // Resolve toponyms without context
3 for  $m \in M$  do
4    $\hat{e} \leftarrow \text{GEO}\text{NORM}(m, E)$ 
5   if  $\text{TYPE}(\hat{e}) \in \{\text{adm1}, \text{adm2}, \text{adm3}\}$  then
6      $\hat{R}[m] \leftarrow \hat{e}$ 
7      $C \leftarrow C \cup \{\text{CODE}(\hat{e})\}$ 
   // Resolve toponyms with context
8  $c \leftarrow "||".\text{join}(C)$ 
9 for  $m \in M$  do
10  if  $m \notin \hat{R}$  then
11   $\hat{R}[m] \leftarrow \text{GEO}\text{NORM}(m + c, E)$ 
12 return  $\hat{R}$ 

```

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

Table 1: Numbers of articles (Art.) and manually annotated toponyms (Topo.) in the train, development, and test splits of the toponym resolution corpora.

global and local news sources. As there are no standard publicly available splits for these datasets, we split each dataset into a train, development, and test set according to a 70%, 10%, and 20% ratio. To enable replicability, we will release these splits upon publication. The statistics of all datasets are shown in table 1.

4.2 Database

Our datasets use GeoNames², a crowdsourced database of geospatial locations, with almost 7 million entries and a variety of information such as geographic coordinates (latitude and longitude), alternative names, feature type (country, city, river, mountain, etc.), population, elevation, and positions within a political geographic hierarchy. An example entry from GeoNames is shown in fig. 2.

4.3 Evaluation Metrics

There is not yet agreement in the field of toponym resolution on a single evaluation metric. Therefore, we gather metrics from prior work and use all of them for evaluation.

²<https://www.geonames.org/>

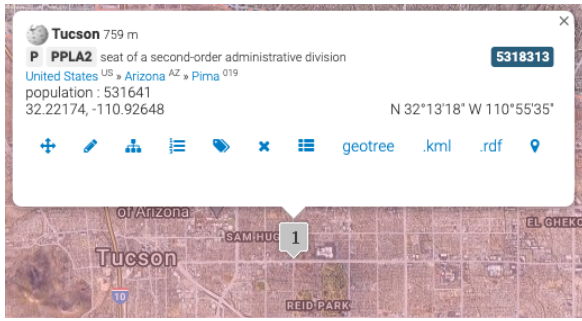


Figure 2: An entry for *Tucson* in GeoNames

Accuracy is the number of location mentions where the system predicted the correct database entry ID, divided by the number of location mentions. Higher is better, and a perfect model would have accuracy of 1.0.

Accuracy@161km measures the fraction of system-predicted (latitude, longitude) points that were less than 161 km (100 miles) away from the human-annotated (latitude, longitude) points. Higher is better, and a perfect model would have Accuracy@161km of 1.0.

Mean error distance calculates the mean over all predictions of the distance between each system-predicted and human-annotated (latitude, longitude) point. Lower is better, and a perfect model would have a mean error distance of 0.0.

Area Under the Curve calculates the area under the curve of the distribution of geocoding error distances. Lower is better, and a perfect model would have an area under the curve of 0.0.

4.4 Implementation details

We implement the candidate reranker with Lucene³ v8.4.1 under Java 1.8. When indexing GeoNames, we also index countries under their adjectival forms in Wikipedia⁴. We implement the candidate reranker with the PyTorch⁵ v1.7.0 APIs in Huggingface Transformers v2.11.0 (Wolf et al., 2020), using either `bert-base-uncased` or `bert-multilingual-uncased`. We train with the Adam optimizer, a learning rate of $1e-5$, a maximum sequence length of 128 tokens, and a number of epochs of 30. We explored a small number of learning rates ($1e-5$, $1e-6$, $5e-6$) and epoch

³<https://lucene.apache.org/>

⁴https://en.wikipedia.org/wiki/List_of_adjectival_and_demonymic_forms_for_countries_and_nations

⁵<https://pytorch.org/>

numbers (10, 20, 30, 40) on the development data. When training without context, we use one Tesla V100 GPU with 32GB memory and a batch size of 8. When training with context, we use four Tesla V100 GPU with 32GB memory and a batch size of 32. The total number of parameters in our model is 168M and the training time is about 3 hours.

4.5 Systems

We compare to a variety of geocoding systems:

Edinburgh Grover et al. (2010) introduced a rule-based extraction and disambiguation system that uses heuristics such as population count, spatial minimization, type, country, and some contextual information (containment, proximity, locality, clustering) to score, rank, and choose a candidate.

Mordecai Halterman (2017) introduced a generate-and-rank approach that uses Elasticsearch to generate candidates and neural networks based on word2vec (Mikolov et al., 2013) to rerank them. Its models are trained on proprietary data.

CamCoder Gritta et al. (2018a) introduced a tile-classification approach that combines a convolutional network over the target mention and 400 tokens of context with a population vector derived from location mentions in the context and populations from GeoNames. CamCoder predicts one of 7823 tiles of the earth’s surface. See appendix A.2 for further CamCoder details.

DeezyMatch Hosseini et al. (2020) introduced a vector-space approach that first pre-trains an LSTM-based classifier on GeoNames taking string pairs as input, and then fine-tunes the pair classifier on the target dataset. The trained DeezyMatch model compares mentions to database entries by generating vector representations for both and measuring their L2-norm distance or cosine similarity.

SAPBERT Liu et al. (2021) introduced a vector-space approach that pretrains a transformer network on the database using a self-alignment metric learning objective and online hard pairs mining to cluster synonyms of the same concept together and move different concepts further away. The pre-trained SAPBERT is then fine-tuned on the target dataset. SAPBERT was trained for the biomedical domain, but is easily retrained for other domains. We pre-train SAPBERT on GeoNames and finetune it on the toponym resolution datasets.

Model	LGL (test)		GeoWebNews (test)		TR-News (test)	
	R@1	R@20	R@1	R@20	R@1	R@20
DeezyMatch (Hosseini et al., 2020)	.172	.538	.262	.671	.206	.702
SAPBERT (Liu et al., 2021)	.245	.742	.428	.746	.355	.780
GeoNorm (+gen, -rank)	.606	.962	.694	.866	.716	.965

Table 2: Performance of candidate generators on the test sets. R@1 is useful for measuring the accuracy of the candidate generator when used directly as a geocoder. R@20 is useful for estimating the ceiling performance of a top-20 reranker based on that candidate generator.

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., 2018a)	.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., 2022a)	.576	-	-	-	.658	-	-	-	.720	-	-	-
ReFinED (fine-tuned)	.786	-	-	-	.782	-	-	-	.858	-	-	-
GeoNorm (+gen -rank)	.606	.685	119	.263	.694	.774	92	.194	.716	.812	95	.169
GeoNorm (+gen +rank, -context)	.761	.785	59	.167	.788	.834	61	.131	.798	.816	89	.154
GeoNorm (+gen +rank, +context=2stage)	.807	.824	46	.135	.828	.862	55	.114	.918	.933	34	.057
GeoNorm (+gen +rank, +context=2stage, +alldata)	.799	.828	52	.136	.832	.876	54	.104	.897	.911	36	.073

Table 3: 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 Edinburgh or ReFinED as these extraction+disambiguation systems do not make predictions for all mentions. The best performance on each dataset+metric is in bold (excluding the final model that was trained on more data).

ReFinED Ayoola et al. (2022a) introduced a vector-space approach for joint extraction and disambiguation of Wikipedia entities. One transformer network generates contextualized embeddings for tokens in the text, another generates embeddings for entries in the ontology, and tokens are matched to entries by comparing dot products over embeddings. ReFinED was trained on Wikipedia, and Wikipedia entries for place names have GeoNames IDs, so ReFinED can be used as a geocoder.

ReFinED (fine-tuned) ReFinED can also be fine-tuned, so we take the released version of ReFinED and fine-tune it for geocoding on each of the toponym datasets.

5 Results

We first evaluate our context-free candidate generator, comparing it to recent context-free candidate generators. Table 2 shows that our approach outperforms approaches from prior work by large margins, both in accuracy of the top entry (R@1) and whether the correct entry is in the top 20 (R@20).

We next evaluate our complete generate-and-rank system against other geocoders. We first per-

form model selection on the development set as described in appendix A.3 to select four models to run on the test set: the candidate generator alone, the best generate-and-rank system with no context, and the best generate-and-rank system with context. Table 3 shows that our proposed GeoNorm model outperforms all prior work across all toponym resolution test sets on all metrics. Even without incorporating context, our generate-and-rank framework meets or exceeds the performance of almost all models from prior work. The exception is ReFinED, where our context-free model outperforms ReFinED out-of-the-box, but slightly underperforms our finetuned version of ReFinED. However, adding the novel two-stage document-level context yields large gains over the context free version of our model, and outperforms even the finetuned ReFinED. The final row the table shows the performance of a model trained on the combined training data from all datasets, which we release for English geocoding under the Apache License v2.0, for off-the-shelf use at <https://github.com/clulab/geonorm>.

Example	Candidate				Rank				
	Name	Pop.	Type	State	RF	G	GR	GRC3	GRCD
1 <i>The educational philosophy at the Washington Latin School in <u>Alexandria</u> is somewhat similar to Ahlstrom’s previous endeavors.</i>	Alexandria	159467	PPLA2						1
	City of Alexandria	139966	ADM2		1				
2 <i>It was Los Angeles police officers she at-tempted to blow up.</i>	Los Angeles County	9818605	ADM2		1	2			
	Los Angeles	3971883	PPLA2		2	1			
	Los Angeles	125430	PPLA2		3	3			
	Los Angeles	4217	PPL		4	4			
3 <i>the Minnesota State Patrol urges motorists to drive with caution as flooding continues to affect area highways. Water over the road-way is currently affecting the following areas in Becker, <u>Clay</u>, and Douglas</i>	Clay County	221939		Missouri	1				4
	Clay County	190865		Florida	2				3
	Clay County	58999		Minnesota	3				1
	Clay County	26890		Indiana	4				2
4 <i>he writes, as do my efforts to insure <u>New London</u> is a safe community.</i>	New London County	274055	ADM2		1		3		4
	New London	27179	PPL		2		1		1
	New London	7172	PPL		3		2		3
	New London	1882	PPL		4		4		2

Table 4: Examples of predictions from ReFinED (RF), our candidate generator alone (G), our generate-and-rerank system without context (GR), our system with sentence context (GRC3), and our system with 2-stage document context (GRCD). Target location mentions are underlined. Human annotated ontology entries are in bold.

6 Qualitative Analysis

Table 4 shows some qualitative analysis of errors that ReFinED and different variants of GeoNorm made. Row 1 shows an example where ReFinED fails but GeoNorm succeeds, by more effectively using geospatial metadata such as population and feature type. Row 2 shows an example where GeoNorm fails with a candidate generator alone but succeeds with a context-free reranker, by not relying on population alone and instead jointly considering the name, population, and feature type information (ADM2 represents a county, PPLA2 represents a city). Row 3 shows an example where GeoNorm fails without context but succeeds with context, by taking advantage of the *Minnesota* in the context to select the *Clay County* that would otherwise seem implausible due to its lower population. Finally, row 4 shows an example where our best GeoNorm model still fails. The candidate generator includes the correct ontology entry in its top-k list, but neither the name, population, feature code, nor nearby context suggest the correct candidate. The global context includes toponyms from the same state, allowing the model with document context to move the correct answer up from rank 4 to rank 2. But fully addressing this issue would likely require predicting countries and states of toponyms in the text before resolving them.

7 Limitations

GeoNorm’s candidate generator is based on information retrieval. This is efficient but not very flexible in string matching, and when the candidate generator fails to produce the correct candidate entry, the candidate reranker also necessarily fails. For example, as table 2 shows, GeoNorm’s reranker achieves only .866 recall@20 on the GeoWebNews dataset, meaning that 13.4% of the time, the correct candidate is not in the top 20 results returned by the candidate generator. One solution might be to replace the information retrieval based candidate generator with a neural network to provide more robust string matching, though the neural network candidate generators from prior work in table 2 actually perform worse than GeoNorm’s candidate generator. Another solution may be to find smarter ways to filter the generated candidates, perhaps by building on the two-stage resolution approach to use document-level context to filter the candidates to those in appropriate countries and states.

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

8 Conclusion

We propose a new toponym resolution architecture, GeoNorm, that combines the tight ontology integration of generate-and-rerank systems with the robust text encoding of deep neural networks. GeoNorm consists of an information retrieval-based candidate generator, a BERT-based reranker that incorporates features important to toponym resolution such as population and type of location, and a novel two-stage resolution strategy that incorporates document-level context. We evaluate our proposed architecture against prior state-of-the-art, using multiple evaluation metrics and multiple datasets. GeoNorm achieves new state-of-the-art performance on all datasets.

References

- Benjamin Adams and Grant McKenzie. 2018. Crowdsourcing the character of a place: Character-level convolutional networks for multilingual geographic text classification. *Transactions in GIS*, 22(2):394–408.
- 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. 2022a. [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.
- Tom Ayoola, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, and Andrea Pierleoni. 2022b. [ReFinED: An efficient zero-shot-capable approach to end-to-end entity linking](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 209–220, Hybrid: Seattle, Washington + Online. 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 Baldrige, 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.
- Fredric Gey, Ray Larson, Mark Sanderson, Hideo Joho, Paul Clough, and Vivien Petras. 2005. Geoclef: the clef 2005 cross-language geographic information retrieval track overview. In *Workshop of the cross-language evaluation forum for european languages*, pages 908–919. Springer.
- Milan Gritta, Mohammad Taher Pilehvar, and Nigel Collier. 2018a. [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.
- Milan Gritta, Mohammad Taher Pilehvar, Nut Limsoatham, and Nigel Collier. 2018b. [What’s missing in geographical parsing?](#) *Language Resources and Evaluation*, 52(2):603–623.
- 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.
- Thi Bich Ngoc Hoang and Josiane Mothe. 2018. Location extraction from tweets. *Information Processing & Management*, 54(2):129–144.
- 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.
- Zongcheng Ji, Qiang Wei, and Hua Xu. 2020. Bert-based ranking for biomedical entity normalization. *AMIA Summits on Translational Science Proceedings*, 2020:269.
- Ehsan Kamaloo 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 Baldrige, 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.
- Abhinav Kumar and Jyoti Prakash Singh. 2019. Location reference identification from tweets during emergencies: A deep learning approach. *International journal of disaster risk reduction*, 33:365–375.
- Egoitz Laparra and Steven Bethard. 2020. [A dataset and evaluation framework for complex geographical description parsing](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 936–948, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Sunshin Lee, Mohamed Farag, Tarek Kanan, and Edward A Fox. 2015. Read between the lines: A machine learning approach for disambiguating the geo-location of tweets. In *Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 273–274.
- 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.
- Xiangyang Luo, Yaqiong Qiao, Chenliang Li, Jiangtao Ma, and Yimin Liu. 2020. An overview of microblog user geolocation methods. *Information Processing & Management*, 57(6):102375.
- 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.
- Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Efficient estimation of word representations in vector space](#). In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.
- Afshin Rahimi, Timothy Baldwin, and Trevor Cohn. 2017. [Continuous representation of location for geolocation and lexical dialectology using mixture density networks](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 167–176, Copenhagen, Denmark. Association for Computational Linguistics.
- Afshin Rahimi, Duy Vu, Trevor Cohn, and Timothy Baldwin. 2015. [Exploiting text and network context for geolocation of social media users](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1362–1367, Denver, Colorado. Association for Computational Linguistics.

- Stephen Roller, Michael Speriosu, Sarat Rallapalli, Benjamin Wing, and Jason Baldridge. 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*, pages 1500–1510, Jeju Island, Korea. Association for Computational Linguistics.
- 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, Sofia, Bulgaria. Association for Computational Linguistics.
- 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.
- 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, Minneapolis, Minnesota, USA. Association for Computational Linguistics.
- 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.
- Dongfang Xu, Zeyu Zhang, and Steven Bethard. 2020. [A generate-and-rank framework with semantic type regularization for biomedical concept normalization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8452–8464, Online. Association for Computational Linguistics.
- Ikuya Yamada, Koki Washio, Hiroyuki Shindo, and Yuji Matsumoto. 2022. [Global entity disambiguation with BERT](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3264–3271, Seattle, United States. 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.

A Appendix

A.1 Performance by toponym type

Table A1 shows that without context, GeoNorm is most precise at resolving toponyms at the top of the hierarchy, like countries and states.

A.2 CamCoder details

The original CamCoder code, when querying GeoNames to construct its input population vector from location mentions in the context, assumes it has been given canonical names for those locations. Since canonical names are not known before locations have been resolved to entries in the ontology, we have CamCoder use mention strings instead of canonical names for querying GeoNames.

A.3 Model selection

We performed model selection on the development sets as shown in table A2. All GeoNorm models that included a reranker (R) outperformed the candidate generator (G) alone. We explored the population (P) and type (T) features in models without context, and found that they helped slightly on LGL and GeoWebNews but hurt slightly on TR-News. For models with context, rerankers fine-tuned from `bert-multilingual-uncased` (M) slightly outperformed models fine-tuned from `bert-base-uncased`. Adding sentence level context (C1/C3/C5) to the rerankers helped on TR-News, but did not help on LGL or GeoWebNews. Applying the two-stage algorithm for document-level context led to large gains on LGL and TR-News, but did not help on GeoWebNews.

We thus selected the following models for evaluation: GeoNorm G, GeoNorm GRPT, and GeoNorm GRPTMCD.

A.4 Artifact intended use and coverage

The intended use of `bert-base-uncased` and `bert-multilingual-uncased` is to be “fine-tuned on tasks that use the whole sentence”⁶. We have used them for that purpose when encoding the context, but also for the related task of encoding place names, which are usually short phrases. These artifacts are trained on English books and English Wikipedia and released under an Apache 2.0 license which is compatible with our use.

The intended use of our geocoding model is matching English place names in text to the Geo-

Names ontology. Though GeoNames covers millions of place names, our evaluation corpora cover only English news articles, and thus the performance we report is only predictive of performance in that domain.

⁶<https://huggingface.co/bert-base-uncased>

Dataset	Precision				Recall			
	Country	State	County	Other	Country	State	County	Other
LGL	0.968	0.806	0.829	0.745	0.893	0.915	0.739	0.763
GWN	1.000	0.765	0.778	0.752	0.966	0.591	1.000	0.810
TR-News	1.000	1.000	0.000	0.830	1.000	1.000	0.000	0.830

Table A1: Precision and recall of GeoNorm (without context) on three geocoding development sets.

Model	LGL (dev)				GeoWebNews (dev)				TR-News (dev)			
	Acc	A161	Err	AUC	Acc	A161	Err	AUC	Acc	A161	Err	AUC
GeoNorm G	.594	.671	201	.289	.644	.858	73	.165	.677	.735	187	.242
GeoNorm GR	.802	.819	64	.141	.865	.925	39.5	<u>.072</u>	.897	.912	64.0	<u>.081</u>
GeoNorm GRP	.792	.819	68	.141	.861	.918	34.7	<u>.072</u>	.868	.882	65.7	.100
GeoNorm GRT	<u>.807</u>	.828	61	<u>.134</u>	.865	.915	31.9	<u>.073</u>	.897	.912	42.7	.074
GeoNorm GRPT	.797	<u>.821</u>	57	.140	.886	.940	29.8	.060	<u>.882</u>	<u>.897</u>	<u>63.5</u>	.090
GeoNorm GRPTM	.814	.828	<u>60</u>	.132	<u>.879</u>	.922	43.2	<u>.072</u>	<u>.882</u>	<u>.897</u>	65.0	.092
GeoNorm GRPTC1	.807	.823	<u>55</u>	.132	.865	.915	39.3	.075	.882	.882	110	.109
GeoNorm GRPTC3	.807	.816	65	.142	.868	.918	40.3	.073	.882	.897	64.9	.092
GeoNorm GRPTC5	.802	.814	68	.145	.865	.911	42.8	.078	.897	.912	64.0	.081
GeoNorm GRPTMC1	<u>.816</u>	.831	62	.133	.872	.940	23.5	.057	.882	.897	64.6	.090
GeoNorm GRPTMC3	.809	<u>.833</u>	59	<u>.129</u>	<u>.875</u>	.922	35.4	.073	<u>.912</u>	<u>.927</u>	<u>40.6</u>	<u>.063</u>
GeoNorm GRPTMC5	.807	.823	61	.137	.872	.940	<u>29.4</u>	<u>.060</u>	.868	.882	72.6	.103
GeoNorm GRPTMCD	.885	.897	29	.079	.879	<u>.925</u>	31.0	.065	.971	.985	6.8	.010

Table A2: Performance on the development 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). The top score in each group is in bold, the second best score is underlined. Model features are indicated by the string of characters: G means the candidate generator was applied, R means a reranker was applied, P means the reranker included the population feature, T means the reranker included the type feature, M means the reranker was fine-tuned from bert-multilingual-uncased instead of bert-base-uncased, C1/C3/C5 means the reranker included 1/3/5 sentences of context, and CD means the reranker included the two-stage document-level context algorithm.