Multi-Level Gazetteer-Free Geocoding

Sayali Kulkarni*
Google Research
sayali@google.com

Shailee Jain*†
University of Texas, Austin
shailee@cs.utexas.edu

Mohammad Javad Hosseini†
University of Edinburgh
javad.hosseini@ed.ac.uk

Jason Baldridge
Google Research
jasonbaldridge@google.com

Eugene Ie
Google Research
eugeneie@google.com

Li Zhang
Google Research
liqzhang@google.com

Abstract

We present a multi-level geocoding model (MLG) that learns to associate texts to geographic coordinates. The Earth's surface is represented using space-filling curves that decompose the sphere into a hierarchical grid. MLG balances classification granularity and accuracy by combining losses across multiple levels and jointly predicting cells at different levels simultaneously. It obtains large gains without any gazetteer metadata, demonstrating that it can effectively learn the connection between text spans and coordinates—and thus makes it a gazetteer-free geocoder. Furthermore, MLG obtains state-of-the-art results for toponym resolution on three English datasets without any dataset-specific tuning.

1 Introduction

Geocoding is the task of resolving location references in text to geographic coordinates or regions. It is often studied in social networks, where metadata and the network itself provide additional non-textual signals (Backstrom et al., 2010; Rahimi et al., 2015). If locations can be mapped to an entity in a knowledge graph, toponym resolution – a special case of entity resolution – can be used to resolve references to locations. Past work used heuristics based on location popularity (Leidner, 2007) and distance between candidate locations (Speriosu and Baldridge, 2013), as well as learned associations from text to locations. However, such approaches have a strong bias for highly-populated locations, especially for social media.

We present Multi-Level Geocoder (MLG, Fig. 1), a model that learns spatial language representations and maps toponyms to coordinates on Earth's surface. This geocoder is not restricted to resolving toponyms to specific location *entities*, but rather to geo-coordinates directly. MLG can thus be extended to any arbitrary location references in future without having to rely on its presence in the gazetteer. For comparative evaluation, we use three English toponym resolution datasets from distinct textual domains. MLG shows strong performance, even without gazetteer and population metadata.

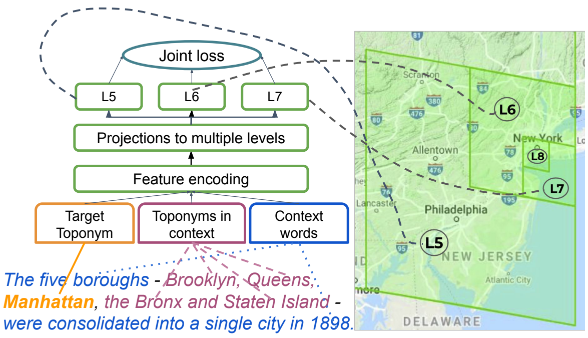


Figure 1: Overview of Multi-Level Geocoder, using multiple context features and jointly predicting cells at multiple levels of the S2 hierarchy.

MLG is a text-to-location neural geocoder. We represent the locations using S2 geometry[1]—a hierarchical discretization of the Earth's surface based on space-filling curves. S2 naturally supports spatial representation at multiple levels, including very fine grained cells (as small as 1cm^2 at level 30). Here, we use combinations of levels 4 ($\sim$300K km^2) to 8 ($\sim$1K km^2). Large cells are easy to predict accurately; however, they are too coarse on their own, and perform poorly on metrics that consider error distances. Smaller cells improve granularity but result in larger and harder output spaces with less training evidence per cell. MLG balances classification granularity and accuracy by predicting at multiple S2 levels and jointly optimizing for the loss at each level. Fig. 1 shows an area around New York City covered by cell id `0x89c25` at level 8 and `0x89c4` at level 5. This is more fine-grained than previous work that does text-to-location geocoding (Gritta et al., 2018a), which uses arbitrary square-degree cells, e.g. 2°-by-2° cells ($\sim$48K km^2).

Unlike previous work that relies on external gazetteer information, MLG is more flexible and can predict geolocation only from context. For instance, it predicts the location of *Manhattan* from the surrounding words (*The five boroughs - Brooklyn, Queens, the Bronx and*

[*] Equal contribution

[†] Work done during internship at Google

[1] https://s2geometry.io/

Proceedings of Second International Combined Workshop on Spatial Language Understanding andGrounded Communication for Robotics, pages 79–88
August 5–6, 2020. ©2021 Association for Computational Linguistics

Staten Island - . . .). Earlier approaches instead relied on a knowledge graph that had *Manhattan* as an entity. While the hierarchical geolocation model of Wing and Baldridge (2014) over kd-trees has some more fine-grained cells, MLG predicts over a much larger set of smaller cells. Furthermore, MLG is a single model that jointly incorporates multiple levels rather than ensembling independent per-cell models for each level.

Our main contributions are the following.

- We define MLG, a model that jointly predicts cells at multiple levels, including finer-grained cells than previous work.
- We show that S2 provides a strong and standardized hierarchical discretization of the Earth's surface for cell-based geocoders.
- We show that it is possible and even preferable to eschew gazetteer metadata. In particular, our experiments show that this strategy generalizes much better.
- We show state-of-the-art performance on three English datasets *without* any fine-tuning.
- When analyzing these datasets, we found inconsistencies in the true coordinates that we unify to support consistent evaluation.[2]

2 Spatial representations

Geocoders map text spans to geo-coordinates—a prediction over a continuous space representing the surface of a sphere. We relax the problem from continuous space to discrete space by quantizing the Earth's surface as a grid and performing multi-class prediction over the grid's cells. We construct a hierarchical grid using the S2 library.[3] S2 projects the six faces of a cube onto the Earth's surface and each face is recursively divided into 4 quadrants, as shown in Figure 1. Cells at each level are indexed using a Hilbert curve. Each S2 cell is represented as a 64-bit unsigned integer and can correspond to areas as small as $\approx 1cm^2$. S2 cells preserve cell size across the globe better than commonly-used degree-square grids (e.g. 1°x1°) (Serdyukov et al., 2009; Wing and Baldridge, 2011). Hierarchical triangular meshes (Szalay et al., 2007) and Hierarchical Equal Area iso-Latitude Pixelation (Melo and Martins, 2015) are alternatives that preserve cell size better, but S2 is easier to work with and has strong, standard tooling.

Our experiments go as far as S2 level eight (of thirty), but our approach is extendable to any level of granularity and could support very fine-grained locations like buildings and landmarks. The built-in hierarchical nature of S2 cells makes it well suited as a scaffold for models that learn and combine evidence from multiple levels. This combines the best of both worlds: specificity at finer levels and aggregation/smoothing at coarser levels.

Roller et al. (2012) use adaptive, variable shaped cells based on k-d trees; such grids can adapt to the different shapes of a region but depend on the locations of labeled examples in a training resource. As such, a k-d tree grid may not generalize well to examples with different distributions from training resources. Spatial hierarchies based on containment relations among entities rely heavily on metadata like GeoNames (Kamalloo and Rafiei, 2018). Polygons for geopolitical entities such as city, state, and country (Martins et al., 2015) are perhaps ideal, but these too require detailed metadata for all toponyms, managing non-uniformity of the polygons, and general facility with GIS tools. The Point-to-City (P2C) method applies an iterative k-d tree-based method for clustering coordinates and associating them with cities (Fornaciari and Hovy, 2019b). S2 can represent such hierarchies in various levels without relying on external metadata.

S2 Level	number of cells	Avg area
L4	1.5k	332
L5	6.0k	83
L6	24.0k	21
L7	98.0k	5
L8	393.0k	1

Table 1: S2 levels used in MLG. Average area is in 1k km^2.

In accordance with the nature of the problem over continuous space, studies using bivariate Gaussians on multiple flattened regions (Eisenstein et al., 2010; Priedhorsky et al., 2014)) perform well on distance based metrics, but this involves difficult trade-offs between flattened region sizes and the level of distortion they introduce. Some of the early models used with grid-based representations were probabilistic language models that produce document likelihoods in different geospatial cells (Serdyukov et al., 2009; Wing and Baldridge, 2011; Dias et al., 2012; Roller et al., 2012). Extensions include domain adapting language models from various sources (Laere et al., 2014), hierarchical discriminative models (Wing and Baldridge, 2014; Melo and Martins, 2015), and smoothing sparse grids with Gaussian priors (Hulden et al., 2015). Alternatively, Fornaciari and Hovy (2019a) use a multi-task learning setup that assigns probabilities across grids and also predicts the true location through regression. Melo and Martins (2017) cover a broad survey of document geocoding. Much of this work has been conducted on social media data like Twitter, where additional information beyond the text—such as the network connections and user and document metadata—have been used (Backstrom et al., 2010; Cheng et al., 2010; Han et al., 2014; Rahimi et al., 2015, 2016, 2017). MLG is not trained on social media data and hence, does not need additional network information. Further, the data does not have a character limit like tweets, so models can learn from long text sequences.

[2] https://github.com/google-research-datasets/mlg_evaldata

[3] https://s2geometry.io/

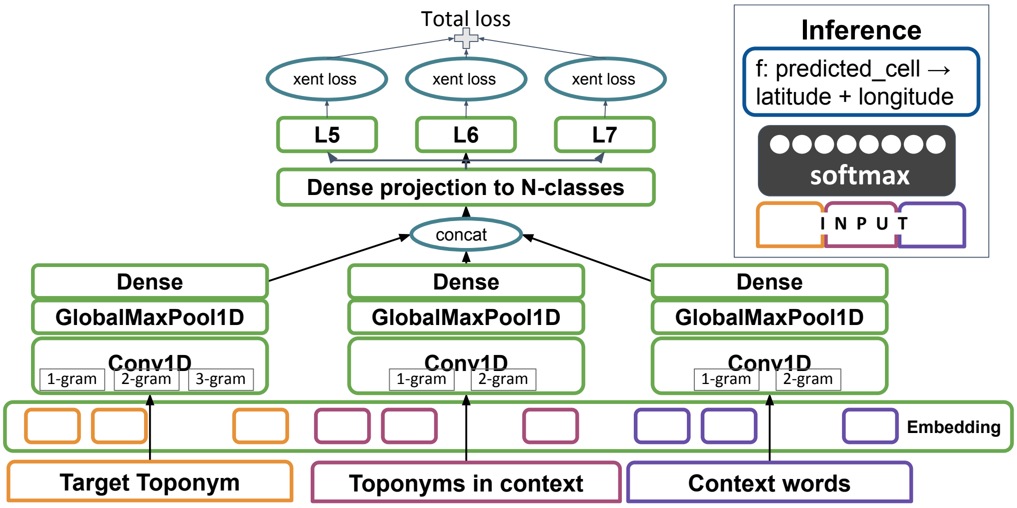


Figure 2: Multi-Level Geocoder model architecture and inference setup.

3 Multi-Level Geocoder (MLG)

Multi-Level Geocoder (MLG, Figure 2) is a text-to-location CNN-based geocoder. Context features are similar to CamCoder (Gritta et al., 2018a) but we exclude its metadata-based MapVec feature. Locations are represented using a hierarchical S2 grid; this enables joint multi-level prediction, by optimizing for total loss computed from all levels.

3.1 Prior geocoding models

Toponym resolution identifies place mentions in text and predicting the precise geo-entity in a knowledge base (Leidner, 2007; Gritta et al., 2018b). The knowledge base is then used to obtain the geo-coordinates of the predicted entity for the geocoding task. Rule-based toponym resolvers (Smith and Crane, 2001; Grover et al., 2010; Tobin et al., 2010; Karimzadeh et al., 2013) rely on hand-built heuristics like population from metadata resources like Wikipedia and GeoNames[4] gazetteer. This works well for many common places, but it is brittle and cannot handle unknown or uncommon place names. As such, machine learned approaches that use toponym context features have demonstrated better performance (Speriosu and Baldridge, 2013; Zhang and Gelernter, 2014; DeLozier et al., 2015; Santos et al., 2015). A straightforward–but data hungry–approach learns a collection of multi-class classifiers, one per toponym with a gazetteer's locations for the toponym as the classes (e.g., the WISTR model of Speriosu and Baldridge (2013)).

A hybrid approach that combines learning and heuristics by predicting a distribution over the grid cells and then filtering the scores through a gazetteer works for systems like TRIPDL (Speriosu and Baldridge, 2013) and TopoCluster (DeLozier et al., 2015). A combination of classification and regression loss to predict over recursively partitioned regions shows promising results with in-domain training (Cardoso et al., 2019). CamCoder (Gritta et al., 2018a) uses this strategy with a much stronger neural model and achieves state-of-the-art results. It incorporates side metadata in the form of its *MapVec* feature vector, which encodes knowledge of potential locations and their populations matching all toponym in the text. It thus uses population signals in both the MapVec feature in training and in output predictions biasing the predictions toward locations with larger populations.

3.2 Building blocks

MLG uses a convolutional neural network to map input text to S2 cells at a given granularity.

Input MLG extracts three features from the input context: (a) token sequence ($w_{a,1:l_a}$) is all the tokens in input, (b) toponym mentions ($w_{b,1:l_b}$) is the list of all locations words in the context, and (c) surface form of the target toponym ($w_{c,1:l_c}$) that is to be geo-located. All text inputs are transformed uniformly, using shared model parameters. Let input text content be denoted as a word sequence $w_{x,1:l} = [w_{x,1}, \dots, w_{x,l}]$, initialized using GloVe embeddings $\phi(w_{x,1:l}) = [\phi(w_{x,1}), \dots, \phi(w_{x,l})]$ (Pennington et al., 2014).

Consider a short context for *Manhattan* as "*Manhattan is the smallest and most densely populated borough compared to others - Bronx, Brooklyn, Queens, and Staten Island.*" All tokens are lower cased and we get w_a as ["*is*", "*the*", "*smallest*", "*and*", ...], toponym mentions w_b are ["*bronx*", ... , "*staten*", "*island*"], and surface form of target toponym w_c would be "*manhattan*".

Model 1D convolutional filters capture n-gram sequences through $\texttt{Conv1D}_n(\cdot)$, followed by max pooling and then projection to a dense layer to get $\texttt{Dense}(\texttt{MaxPool}(\texttt{Conv1D}_n(\phi(w_{x,1:l}))))\in\mathbb{R}^{2048}$, where n=$\{1,2\}$ for the token sequence and toponym mentions, and n=$\{1,2,3\}$ for the target toponym.

[4] www.geonames.org

These projections are concatenated to form the full input representation. MLG is designed to study effectiveness of spatial language representation without any gazetteer information. Hence we choose a CNN-based architecture, but can be extended to large scale pretrained language models (Devlin et al. (2018)).

Output An S2 cell is predicted at the highest granularity using a softmax over the output space. The center of the predicted S2 cell is taken as the predicted coordinates. *Optionally*, the predicted cells may be snapped to the closest valid cells that overlap the potential gazetteer locations for the toponym, weighted by their population (similar to previous work, like CamCoder).

3.3 Multi-level classification

MLG's core block is a multi-class classifier using a CNN. Rather than predicting cells at a single level, we project the output onto multiple levels with a multi-headed model. The penultimate layer maps representations of the input to probabilities over the finest-grained cells. Gradient updates are computed using cross entropy loss between predicted probabilities $\mathbf{p}$ and the one-hot true class vector $\mathbf{c}$.

MLG exploits the natural hierarchy of geographic locations by jointly predicting at different levels of granularity. CamCoder uses 7.8K output classes representing 2x2 degree tiles (after filtering cells that have no support in training, such as over bodies of water, to limit the class space). This requires maintaining a cumbersome mapping between actual grid cells and the classes. MLG's multi-level hierarchical representation overcomes this problem by including coarser levels (like L5) to guide the predictions at finer-grained levels. We focus on three levels that are appropriate for the task: L5, L6 and L7 (shown in Table 1), each giving 6K, 24K, and 98K output classes, respectively.

We define losses at each level (L5, L6, L7) and minimize them jointly, i.e., $\mathcal{L}_{\text{total}} = (\mathcal{L}(\mathbf{p}_{\text{L5}}, \mathbf{c}_{\text{L5}}) + \mathcal{L}(\mathbf{p}_{\text{L6}}, \mathbf{c}_{\text{L6}}) + \mathcal{L}(\mathbf{p}_{\text{L7}}, \mathbf{c}_{\text{L7}}))/3$. At inference time, a single forward pass computes probabilities at all three levels. The final score for each L7 cell is dependent on its predicted probability as well as the probabilities in its corresponding parent L6 and L5 cells. Then the final score for $s_{\text{L7}}(f) = \mathbf{p}_{\text{L7}}(f) * \mathbf{p}_{\text{L6}}(e) * \mathbf{p}_{\text{L5}}(d)$ and the final prediction is $\hat{y} = \text{argmax}_y\, s_{\text{L7}}(y)$. This approach is easily extensible to capture additional levels of resolution—we also present results with finer resolution at L8, with $\sim$1K km^2 area and coarser resolution at L4 with $\sim$300K km^2 area for comparison.

3.4 Gazetteer-constrained prediction

The only way MLG uses geographic information is from training labels for toponym targets. At test time, MLG predicts a distribution over all cells at each S2 level given the input features and picks the highest probability cell at the most granular level. We use the center of the cell as predicted coordinates. However, when the goal is to resolve a specific toponym, an effective heuristic is to use a gazetteer to filter the output predictions to only those that are valid for the toponym. Furthermore, gazetteers come with population information that can be used to nudge predictions toward locations with high populations—which tend to be discussed more than less populous alternatives. Like DeLozier et al. (2015), we consider both gazetteer-free and gazetteer-constrained predictions.

Gazetteer-constrained prediction makes toponym resolution a sub-problem of entity resolution. As with broader entity resolution, a strong baseline is an alias table (the gazetteer) with a popularity prior. For geographic data, the population of each location is an effective quantity for characterizing popularity: choosing Paris, France rather than Paris, Texas for the toponym *Paris* is a better bet. This is especially true for zero-shot evaluation where one has no in-domain training data.

We follow the strategy of Gritta et al. (2018a) for gazetteer constrained predictions. We construct an alias table which maps each mention m to a set of candidate locations, denoted by $C(m)$ using link information from Wikipedia and the population $\text{pop}(\ell)$ for each location ℓ is read from WikiData.[5] For each of the gazetteer's candidate locations we compute a population discounted distance from the geocoder's predicted location p and choose the one with smaller value as $\text{argmin}_{\ell \in C(m)} \text{dist}(p, \ell) \cdot (1 - c \cdot \text{pop}(\ell) / \text{pop}(m))$. Here, $\text{pop}(m)$ is the maximum population among all candidates for mention m, $\text{dist}(p, \ell)$ is the great circle distance between prediction p and location ℓ, and c is a constant in $[0, 1]$ that indicates the degree of population bias applied. For c=0, the location nearest the prediction is chosen (ignoring population); for c=1, the most populous location is chosen, (ignoring p). This is set to 0.9, which worked best on the development set.

3.5 Training Data and Representation

MLG is trained on geographically annotated Wikipedia pages, *excluding* all pages in WikToR (see Sec. 4.1). For each page with geo-coordinates, we consider context windows of up to 400 tokens (respecting sentence boundaries) as training example candidates. Only context windows that contain the target Wikipedia toponym are used. We use Google Cloud Natural Language API libraries to tokenize[6] the page text and for identifying[7] toponyms in the contexts. We use the July 2019 English Wikipedia dump, which has 1.11M location annotated pages giving 1.76M training examples. This is split 90/10 for training/development.

[5] http://www.wikidata.org

[6] https://cloud.google.com/natural-language/docs/analyzing-syntax

[7] https://cloud.google.com/natural-language/docs/analyzing-entities

Gaz		AUC of error curve ↓				accuracy@161 ↑				Mean error ↓			
Used	Model	WTR	LGL	GV	Avg	WTR	LGL	GV	Avg	WTR	LGL	GV	Avg
Yes	PopBaseline	66	42	41	50	22	57	68	49	4175	1933	898	2335
	CamCoder	24	32	15	24	72	63	82	72	440	877	315	544
	SLG 7	17	28	**13**	19	82	72	**86**	80	480	648	305	478
	MLG 5-7	**15**	**27**	**13**	**18**	**85**	**73**	85	**81**	**347**	**620**	**276**	**414**
No	CamCoder	49	60	65	58	70	38	26	45	239	1419	2246	1301
	SLG 7	39	55	56	50	86	49	48	61	424	1688	1956	1356
	MLG 5-7	**37**	**54**	**55**	**49**	**91**	**53**	**49**	**64**	**180**	**1407**	**1690**	**1092**

Table 2: Comparing population baseline, CamCoder benchmark (our implementation), and our SLG and MLG models on the *unified* data, both with and without the gazetteer filter.

	AUC of error curve ↓				accuracy@161 ↑				Mean error ↓			
Inference	WTR	LGL	GV	Avg	WTR	LGL	GV	Avg	WTR	LGL	GV	Avg
L5-7	**37**	**54**	**55**	**49**	**91**	**53**	**49**	**64**	**180**	**1407**	**1690**	**1092**
Only L5	48	60	62	57	79	45	39	54	285	1599	1957	1280
Only L6	43	57	60	53	90	51	44	62	265	1534	2003	1267
Only L7	38	**54**	56	50	89	51	48	63	349	1525	2014	1296

Table 3: Prediction granularity: performance of MLG trained with multi-level loss on L5, L6 and L7 but using single level at inference time.

4 Evaluation

We train MLG as a general purpose geocoder and evaluate it on toponym resolution. A strong baseline is to choose the most populous candidate location (PopBaseline): i.e. $\text{argmax}_{\ell \in C(m)} \text{pop}(\ell)$

4.1 Evaluation Datasets

We use three public datasets: Wikipedia Toponym Retrieval (WikToR) (Gritta et al., 2018b), Local-Global Lexicon (LGL) (Lieberman et al., 2010), and GeoVirus (Gritta et al., 2018a). See Gritta et al. (2018b) for extensive discussion of other datasets.

WikToR (WTR) is the largest programmatically created corpus that allows for comprehensive evaluation of toponym resolvers. By construction, ambiguous location mentions were prioritized (e.g. "*Lima, Peru*" vs. "*Lima, Ohio*" vs. "*Lima, Oklahoma*" vs "*Lima, New York*"). As such, population-based heuristics are counter-productive in WikToR.

LGL consists of 588 news articles from 78 different news sources. This dataset contains 5,088 toponyms and 41% of these refer to locations with small populations. About 16% of the toponyms are for street names, which do not have coordinates; and hence dropped from our evaluation set. About 2% have an entity that does not exist in Wikipedia, which were also dropped thus leaving 4,172 examples for evaluation.

GeoVirus (GV) is based on 229 WikiNews[8] articles about global epidemics obtained using keywords such as "Bird Flu" and "Ebola". Place mentions are manually tagged and assigned Wikipedia page URLs. In total, this dataset provides 2,167 toponyms for evaluation.

WikToR serves as in-domain Wikipedia-based evaluation data, while both LGL and GeoVirus provide out-of-domain news corpora evaluation.

4.2 Unified evaluation sets

We use the publicly available versions of the three datasets used in CamCoder.[9] However, after analyzing examples across all of them, we identified inconsistencies in location target coordinates.

First, WikToR's evaluation set delivers annotations based on GeoNames DB and Wikipedia APIs. We discovered that WikToR was annotated with an older version of GeoNames DB, which has a known issue of sign flip in either latitude or longitude of some locations. For example, *Santa Cruz, New Mexico* was incorrectly tagged as (35, 106) instead of (35, -106). This affects 296 out of 5,000 locations in WikToR—mostly cities in the United States and a few in Australia.

Second, the target coordinates are inconsistent across the 3 datasets. For example, Canada is (60.0, -95.0) in GeoVirus, (60.0, -96.0) in LGL and (45.4, -75.7) in WikToR. Given our point-based representations, we need consistent coordinates across the evaluation sets. So we re-annotated all three datasets to unify the coordinates for target toponyms.[2] This was done Wikidata to be consistent with Wikipedia training labels.

4.3 Evaluation Metrics

We use three metrics for evaluation: AUC for the error curve, accuracy@161km and mean distance error. AUC[10] is the area under the discrete curve of sorted log-error distances. This is captures the entire distribution of errors and is not sensitive to outliers. It uses the log of the error distances, which appropriately focuses the metric on smaller error distances. Accuracy is the percentage of toponyms that are resolved to within 161km

[8] https://en.wikinews.org

[9] https://github.com/milangritta/Geocoding-with-Map-Vector/tree/master/data

[10] Unlike the standard AUC, lower is better for AUC since this is based on the curve of error distances.

	Dev	AUC of error curve ↓				accuracy@161 ↑				Mean error ↓			
Model	loss	WTR	LGL	GV	Avg	WTR	LGL	GV	Avg	WTR	LGL	GV	Avg
MLG 4-7	8.71	37	55	54	49	91	51	51	64	197	1529	1570	1099
MLG 5-7	**7.25**	37	54	55	49	91	53	49	64	180	1407	1690	1092
MLG 5-8	13.28	38	58	67	54	89	45	24	53	272	1866	3058	1732

Table 4: Models trained with different granularities help trade-off between accuracy and generalization. Selected model MLG 5-7 is based on optimal performance of the holdout.

(100 miles) of their true location. Mean distance error is the average of all distances between predicted locations (center of the predicted S2 cell) and true locations of the target toponym.

We study the benefits of resolving toponyms over multiple levels to account for the range of populations, resolution ambiguity, topological shapes and sizes of different toponyms. We leave the shaping of the output space as future work (e.g., using geopolitical polygons instead of points).

5 Experiments

5.1 Training

MLG is trained using TensorFlow (Abadi et al., 2016) distributed across 13 P100 GPUs. Each training batch processes 512 examples. The model trains up to 1M steps, although they converge around 500K steps. We found an optimal initial learning rate of 10^{-4} decaying exponentially over batches after initial warm-up. For optimization, we use Adam (Kingma and Ba, 2015) for stability.

We considered S2 levels 4 through 8, including single level (SLG) and multi-level (MLG) variations. MLG's architecture offers the flexibility of doing multi-level training but performing prediction with just one level. Based on the loss on Wikipedia development split, we chose multi-level training and prediction with levels 5, 6 and 7.

We stress that our focus is *geocoding without gazetteer information at inference time*. However, we also show that additional gains can be achieved using gazetteers to select relevant cells for a given toponym, and scale the output using the population bias (c) as described in section 3.4.

5.2 Results

Table 2 shows results for the PopBaseline, CamCoder, SLG and MLG models on all three datasets for all metrics. For CamCoder, SLG and MLG, we include results with and without gazetteer filtering (sect. 3.4). Results are reported on the unified datasets. The CamCoder results are from our own implementation and trained on the same examples as MLG training set.

Overall trends The most striking result is MLG's improvement over CamCoder without gazetteer filtering, especially on WikToR—a dataset specifically designed to counteract population priors. MLG clearly generalizes better by leaving out the non-lexical MapVec feature and thereby avoiding the influence of its population bias for the toponyms in the context.

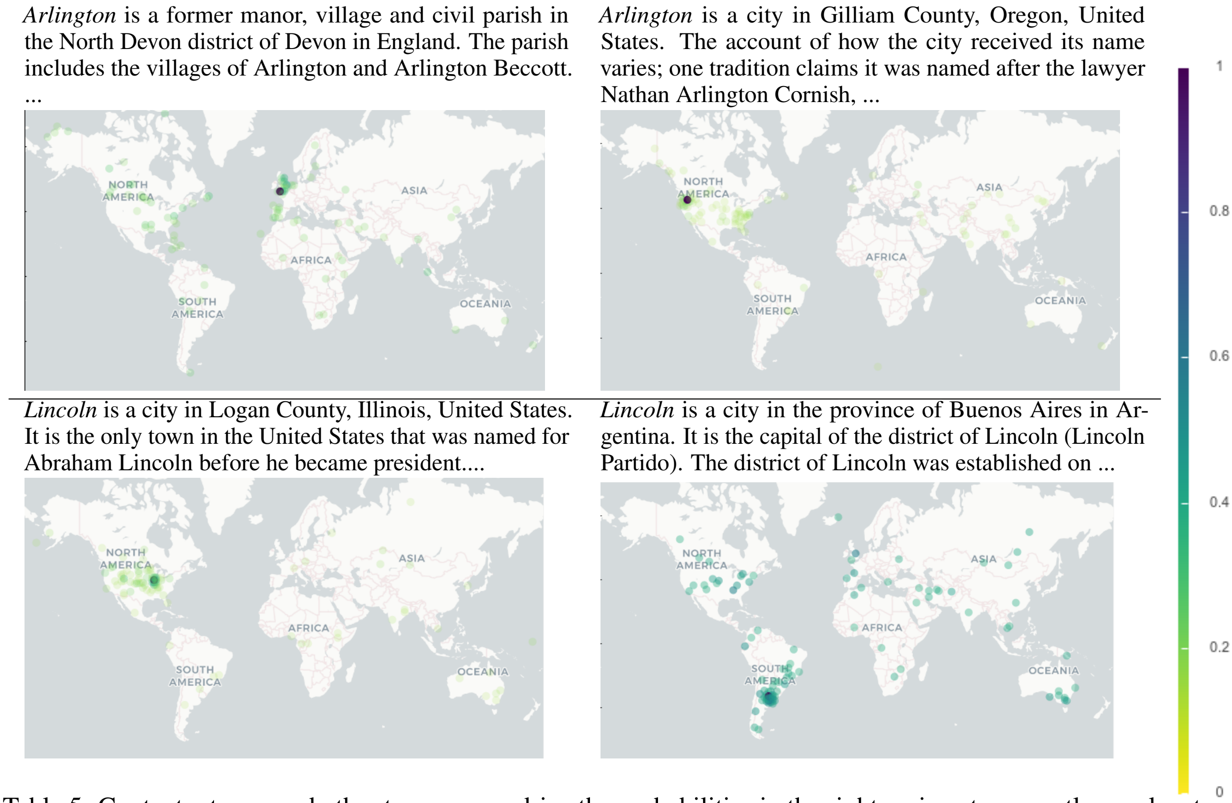
Fine-grained multi-level learning and prediction pays off, both with and without gazetteer filtering. This is particularly clear with AUC, where MLG is 6% better (averaged over all datasets) than CamCoder with the gazetteer filter. Without the filter, MLG has an even larger gain of 9%.

Generalization When not using the gazetteer filter, MLG actually beats the population baseline for WikToR, and it is much closer to the strong population baselines for LGL and GeoVirus than CamCoder and SLG. This indicates that the multi-level approach allows the use of training evidence to generalize better over examples drawn globally (entire world in GeoVirus) as well as locally (the United States of America in LGL).

Multi-level prediction helps. Table 3 compares performance of using individual levels from the same MLG model trained on levels L5, L6 and L7 (without the gazetteer filter). The trade off of predicting at different granularity is clear: when we use lower granularity, e.g. L5 cells, our model can generalize better, but it may be less precise given the large size of the cells. On the other hand, when using finer granularity, e.g. L7 cells, the model can be more accurate in dense regions, but could suffer in sparse regions where there is less training data. Combining the predictions from all levels balances the strengths effectively.

Levels five through seven offer best tradeoff Table 4 shows performance of MLG by training and predicting with multiple levels at different granularities. Overall, using levels five through seven (which has the best development split loss) provides the strongest balance between generalization and specificity. For locating cities, states and countries, especially when choosing from candidate locations in a gazetteer, L8 cells do not provide much greater precision than L7 and suffer from fewer examples as evidence in each cell.

Qualitative examples An effective use of context in correctly predicting coordinates is shown in Table 5 on two examples, *Arlington* and *Lincoln*. In both pairs, the context helps to shift the predictions in the right regions on the map. It is not biased by just the most populous place. Here we only show a part of the context for clarity though the actual context is longer (see Sec. 3.5).



Arlington is a former manor, village and civil parish in the North Devon district of Devon in England. The parish includes the villages of Arlington and Arlington Beccott. ...

Arlington is a city in Gilliam County, Oregon, United States. The account of how the city received its name varies; one tradition claims it was named after the lawyer Nathan Arlington Cornish, ...

Lincoln is a city in Logan County, Illinois, United States. It is the only town in the United States that was named for Abraham Lincoln before he became president....

Lincoln is a city in the province of Buenos Aires in Argentina. It is the capital of the district of Lincoln (Lincoln Partido). The district of Lincoln was established on ...

Table 5: Context – terms and other toponyms – drive the probabilities in the right regions to correctly geo-locate *Arlington* (top) and *Lincoln* (bottom) distributions in different parts of the world.

	AUC of error curve ↓				accuracy@161 ↑				Mean error ↓			
Ablation	WTR	LGL	GeoV	Avg	WTR	LGL	GeoV	Avg	WTR	LGL	GeoV	Avg
all features	37	54	55	49	91	53	49	64	180	1407	1690	1092
- target	38	60	69	55	91	39	18	49	174	2032	2811	1672
- all toponyms	69	75	82	76	29	14	4	16	4487	4442	6360	5096

Table 6: Effect of ablating location features from the input to demonstrate their importance in MLG 5-7.

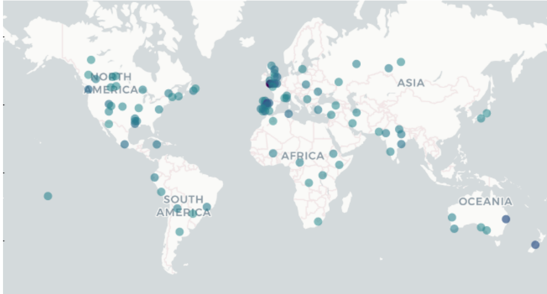


Figure 3: Ablating all toponyms at inference time spreads out the probabilities (points lighted up all over the map) but can still correctly predict *Arlington (England)* purely from context.

Ablations Table 6 shows ablation of salient features at inference time, removing either the target toponym or all toponyms. While masking the target toponym does not change results much except for GeoVirus, masking all other toponyms degrades performance considerably. Nevertheless, it may still be possible with just the context words, which include other named entities, characteristics of the place, and location-focused words in few cases. For example, *Arlington (England)* can be geolocated after all toponyms are masked (Fig. 3), though the distribution is more spread out in this case.

6 Conclusion and Future work

MLG uses multi-level optimization for the inherently hierarchical problem of geocoding. With just textual inputs, we can predict the location of a target toponym with minimal to no metadata from gazetteer and outperform existing benchmark models. MLG can thus be used as a gazetteer-free geocoder, on inputs like historical texts (DeLozier et al., 2016). Further, the models generalize very well across domains, and thus can be used in real-time datasets like news feeds. The multi-level loss can be further refined by using approaches like hierarchical softmax (Morin and Bengio, 2005) to incorporate the conditional probabilities across layers more effectively.

A natural extension would be to fine-tune large pre-trained language models for the geocoding task. We expect that the potential value of this is orthogonal to the contribution of our multi-level loss and the use of S2 cells. Another future direction involves smoothing the label space during training to capture the relations among spatial close cells by defining the loss as a function of Earth mover's distance with approximations like Sinkhorn divergence. This would also enable shaping the output class space to polygons instead of points, which is more realistic for geographical regions.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. TensorFlow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283.

Lars Backstrom, Eric Sun, and Cameron Marlow. 2010. Find me if you can: Improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, pages 61–70.

Ana Cardoso, Bruno Martins, and Jacinto Estima. 2019. *Using Recurrent Neural Networks for Toponym Resolution in Text*.

Zhiyuan Cheng, James Caverlee, and Kyumin Lee. 2010. You are where you tweet: A content-based approach to geo-locating Twitter users. In *Proceedings of the 19th ACM International Conference Information and Knowledge Management (CIKM 2010)*, Toronto, Canada.

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 2015)*, pages 2382–2388, Austin, Texas. AAAI Press.

Grant DeLozier, Ben Wing, Jason Baldridge, and Scott Nesbit. 2016. Creating a novel geolocation corpus from historical texts. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 188–198, Berlin, Germany. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Duarte Dias, Ivo Anastácio, and Bruno Martins. 2012. Geocodificação de documentos textuais com classificadores hierárquicos baseados em modelos de linguagem. *Linguamática*, 4(2):13–25.

Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. 2010. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1277–1287, Cambridge, MA. Association for Computational Linguistics.

Tommaso Fornaciari and Dirk Hovy. 2019a. Geolocation with attention-based multitask learning models. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 217–223, Hong Kong, China. Association for Computational Linguistics.

Tommaso Fornaciari and Dirk Hovy. 2019b. Identifying linguistic areas for geolocation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 231–236.

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 (ACL 2018)*, pages 1285–1296, Stroudsburg, Pennsylvania.

Milan Gritta, Mohammad Taher Pilehvar, Nut Limsopatham, and Nigel Collier. 2018b. What's missing in geographical parsing? *Language Resources and Evaluation*, 52(2):603–623.

Claire Grover, Richard P. 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*, 368:3875–3889.

Bo Han, Paul Cook, and Timothy Baldwin. 2014. Text-based Twitter user geolocation prediction. *Journal of Artificial Intelligence Research*, 49(1):451–500.

Mans Hulden, Miikka Silfverberg, and Jerid Francom. 2015. Kernel density estimation for text-based geolocation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2015)*, pages 145–150, Austin, Texas.

David Jurgens, Tyler Finethy, James McCorriston, Yi Tian Xu, and Derek Ruths. 2015. Geolocation prediction in twitter using social networks: A critical analysis and review of current practice. In *ICWSM*.

Ehsan Kamalloo and Davood Rafiei. 2018. A coherent unsupervised model for toponym resolution. 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 (GIR 2013)*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, California.

Olivier Van Laere, Steven Schockaert, Vlad Tanasescu, Bart Dhoedt, and Christopher Jones. 2014. Georeferencing wikipedia documents using data from social media sources. *ACM Transactions on Information Systems*, pages 1–32.

Jochen L. Leidner. 2007. Toponym resolution in text: annotation, evaluation and applications of spatial grounding. *SIGIR Forum*, 41:124–126.

M.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.

Bruno Martins, Francisco J. López-Pellicer, and Dirk Ahlers. 2015. Expanding the utility of geospatial knowledge bases by linking concepts to wikitext and to polygonal boundaries. In *GIR '15*.

Fernando Melo and Bruno Martins. 2015. Geocoding textual documents through the usage of hierarchical classifiers. In *Proceedings of the 9th Workshop on Geographic Information Retrieval (GIR 15)*, Paris, France.

Fernando Melo and Bruno Martins. 2017. Automated geocoding of textual documents: A survey of current approaches. *Transactions in GIS*, 21(1):3–38.

Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics (AISTATS 2005)*, pages 246–252.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1532–1543, Doha, Qatar.

Reed Priedhorsky, Aron Culotta, and Sara Y. Del Valle. 2014. Inferring the origin locations of tweets with quantitative confidence. pages 1523–1536. Conference on Computer-Supported Cooperative Work.

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, Trevor Cohn, and Timothy Baldwin. 2016. pigeo: A Python geotagging tool. In *Proceedings of ACL-2016 System Demonstrations*, pages 127–132, Berlin, Germany.

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 (NAACL HLT 2015)*, pages 1362–1367, Denver, Colorado.

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 (EMNLP-CONLL 2012)*, page 1500–1510, Jeju, Korea.

João Santos, Ivo Anastácio, and Bruno Martins. 2015. Using machine learning methods for disambiguating place references in textual documents. *GeoJournal*, 80(3):375–392.

Pavel Serdyukov, Vanessa Murdock, and Roelof van Zwol. 2009. Placing flickr photos on a map. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA. Association for Computing Machinery.

David A. Smith and Gregory Crane. 2001. Disambiguating geographic names in a historical digital library. In *Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2001)*, pages 127–136, Berlin, Heidelberg.

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 (ACL 2013)*, pages 1466–1476, Sofia, Bulgaria.

Alexander Szalay, Jim Gray, George Fekete, Peter Kunszt, Peter Kukol, and Ani Thakar. 2007. Indexing the sphere with the hierarchical triangular mesh.

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 (GIR 2010)*, Zurich, Switzerland.

Benjamin Wing and Jason Baldridge. 2014. Hierarchical discriminative classification for text-based geolocation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 336–348, Dohar, Qatar.

Benjamin P. Wing and Jason Baldridge. 2011. Simple supervised document geolocation with geodesic grids. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, pages 955–964, Portland, Oregon.

Wei Zhang and Judith Gelernter. 2014. Geocoding location expressions in Twitter messages: A preference learning method. *Journal of Spatial Information Science*, 9.