Supervised Text-based Geolocation Using Language Models on an Adaptive Grid

Stephen Roller[†] **Michael Speriosu** [‡] **Sarat Rallapalli** [†]
Benjamin Wing [‡] **Jason Baldridge** [‡]
[†]Department of Computer Science, University of Texas at Austin
[‡]Department of Linguistics, University of Texas at Austin
{roller, sarat}@cs.utexas.edu, {speriosu, jbaldrid}@utexas.edu, ben@benwing.com

Abstract

The geographical properties of words have recently begun to be exploited for geolocating documents based solely on their text, often in the context of social media and online content. One common approach for geolocating texts is rooted in information retrieval. Given training documents labeled with latitude/longitude coordinates, a grid is overlaid on the Earth and *pseudo-documents* constructed by concatenating the documents within a given grid cell; then a location for a test document is chosen based on the most similar pseudo-document. Uniform grids are normally used, but they are sensitive to the dispersion of documents over the earth. We define an alternative grid construction using *k*-d trees that more robustly adapts to data, especially with larger training sets. We also provide a better way of choosing the locations for pseudo-documents. We evaluate these strategies on existing Wikipedia and Twitter corpora, as well as a new, larger Twitter corpus. The adaptive grid achieves competitive results with a uniform grid on small training sets and outperforms it on the large Twitter corpus. The two grid constructions can also be combined to produce consistently strong results across all training sets.

1 Introduction

The growth of the Internet in recent years has provided unparalleled access to informational resources. It is often desirable to extract summary metadata from such resources, such as the date of writing or the location of the author – yet only a small portion of available documents are explicitly annotated in this fashion. With sufficient training data, however, it is often possible to infer this information directly from a document's text. For example, clues to the geographic location of a document may come from a variety of word features, e.g. toponyms (*Toronto*), geographic features (*mountain*), culturally local features (*hockey*), and stylistic or dialectical differences (*cool* vs. *kewl* vs. *kool*).

This article focuses on text-based document geolocation, the prediction of the latitude and longitude of a document. Among the uses for this are region-based search engines; tracing the sources of historical documents; location attribution while summarizing large documents; tailoring of ads while browsing; phishing detection when a user account is accessed from an unexpected location; and "activist mapping" (Cobarrubias, 2009), as in the Ushahidi project.[1] Geolocation has also been used as a feature in automatic news story identification systems (Sankaranarayanan et al., 2009).

One of the first works on document geolocation is Ding et al. (2000), who attempt to automatically determine the geographic scope of web pages. They focus on named locations, e.g. cities and states, found in gazetteers. Locations are predicted based on toponym detection and heuristic resolution algorithms. A related, recent effort is Cheng et al. (2010), who geolocate Twitter users by resolving their profile locations against a gazetteer of U.S. cities and training a classifier to identify geographically local words.

An alternative to using a discrete set of locations from a gazetteer is to use information retrieval (IR) techniques on a set of geolocated training documents. A new test document is compared with each

[1] `http://ushahidi.com/`

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 1500–1510, Jeju Island, Korea, 12–14 July 2012.

training document and a location chosen based on the location(s) of the most similar training document(s). For image geolocation, Chen and Grauman (2011) perform mean-shift clustering over training images to discretize locations, then estimate a test image's location with weighted voting from the k most similar documents. For text, both Serdyukov et al. (2009) and Wing and Baldridge (2011) use a similar approach, but compute document similarity based on language models rather than image features. Additionally, they group documents via a uniform geodesic grid rather than a clustered set of locations. This reduces the number of similarity computations and removes the need to perform location clustering altogether, but introduces a new parameter controlling the granularity of the grid. Kinsella et al. (2011) predict the locations of tweets and users by comparing text in tweets to language models associated with zip codes and broader geopolitical enclosures. Sadilek et al. (2012) discretize by simply clustering data points within a small distance threshold, but only perform geolocation within fixed city limits.

While the above approaches discretize the continuous surface of the earth, Eisenstein et al. (2010) predict locations based on Gaussian distributions over the earth's surface as part of a hierarchical Bayesian model. This model has many advantages (e.g. the ability to compute a complete probability distribution over locations), but we suspect it will be difficult to scale up to the large document collections needed for high accuracy.

We build on the IR approach with grids while addressing some of the shortcomings of a uniform grid. Uniform grids are problematic in that they ignore the geographic dispersion of documents and forgo the possibility of greater-granularity geographic resolution in document-rich areas. Instead, we construct a grid using a *k*-d tree, which adapts to the size of the training set and the geographic dispersion of the documents it contains. This can better benefit from more data, since it enables the training set to support more pseudo-documents when there is sufficient evidence to do so, while still ensuring that all pseudo-documents contain comparable amounts of data. It also has the desirable property of generally requiring fewer active cells than a uniform grid, drastically reducing the computation time required to label a test document.

We show that consistently strong results, robust across both Wikipedia and Twitter datasets, are obtained from the union of the pseudo-documents from a uniform and adaptive grid. In addition, a simple difference in the choice of location for a given grid cell – the centroid of the training documents in the cell, rather than the cell midpoint – results in across-the-board improvements. We also construct and evaluate on a much larger dataset of geolocated tweets than has been used in previous papers, demonstrating the scalability and robustness of our methods and confirming the ability of the adaptive grid to more effectively use larger datasets.

2 Data

We work with three datasets: a corpus of geotagged Wikipedia articles and two corpora of geotagged tweets.

GEOWIKI is a collection of 1,019,490 geotagged English articles from Wikipedia. The dump from Wikimedia requires significant processing to obtain article text and location, so we rely on the preprocessed data used by Wing and Baldridge (2011).

GEOTEXT is a small dataset consisting of 377,616 messages from 9,475 users tweeting across 48 American states, compiled by Eisenstein et al. (2010). A document in this dataset is the concatenation of all tweets by a single user, with a location derived from the earliest tweet with specific, GPS-assigned latitude/longitude coordinates.

UTGEO2011 is a new dataset designed to address the sparsity problems resulting from the size of the previous dataset. It is based on 390 million tweets collected across the entire globe between September 4th and November 29th, 2011, using the publicly available Twitter Spritzer feed and global search API. Not all collected tweets were geotagged. To be comparable to GEOTEXT, we discarded tweets outside of North America (outside of the bounding box with latitude/longitude corners at $(25, -126)$ and $(49, -60)$). Following Eisenstein et al. (2010), we consider all tweets of a user concatenated as a single document, and use the earliest collected GPS-assigned location as the gold location. Users without a gold location were discarded. To remove many spammers and robots, we only kept users following 5 to 1000 people, followed by at least 5 users, and authoring no more than 1000 tweets in the three month period. The resulting dataset contains 38 million tweets from 449,694 users, or roughly 85 tweets per user on average. We randomly selected 10,000 users each for development and held-out test evaluation. The remaining 429,694 users serve as a training set termed **UTGEO2011-LARGE**. We also randomly selected a 10,000 user training subset (**UTGEO2011-SMALL**) to facilitate comparisons with GEOTEXT and allow us to investigate the relative improvements for different models with more training data.

Our code and the UTGEO2011 data set are both available for download.[2]

3 Model

Assume we have a collection $\mathbf{d}$ of documents and their associated location labels $\mathbf{l}$. These documents may be actual texts, or they can be pseudo-documents comprised of a number of texts grouped via some algorithm (such as the grids discussed in the next section).

For a test document d_i, its similarity to each labeled document is computed, and the location of the most similar document assigned to d_i. Given an abstract function sim that can be instantiated with an appropriate similarity function (e.g. cosine distance or Kullback-Leibler divergence),

$$loc(d_i) = loc(\arg\max_{d_j \in \mathbf{d}} sim(d_i, d_j)).$$

This is a winner-takes-all strategy, which we follow in this paper. In related work on image geolocation, Hays and Efros (2008) use the same general framework, but compute the location based on the k-nearest neighbors (kNN) rather than the top one. They compute a distribution from the 120 nearest neighbors using mean shift clustering (Comaniciu and Meer, 2002) and choose the cluster with the most members. This produced slightly better results than choosing only the closest image. In future work, we will explore the kNN approach to see if it is more effective for text geolocation.

Following previous work in document geolocation, particularly Serdyukov et al. (2009) (henceforth SMvZ) and Wing and Baldridge (2011) (henceforth W&B), we geolocate texts using a language modeling approach to information retrieval (Ponte and Croft, 1998; Zhai and Lafferty, 2001). For each document d_i, we construct a unigram probability distribution θ_{d_i} over the vocabulary.

We smooth documents using the pseudo-Good-Turing method of W&B, a nonparametric discounting model that backs off from the unsmoothed distribution $\tilde{\theta}_{d_i}$ of the document to the unsmoothed distribution $\tilde{\theta}_D$ of all documents. A general discounting model is as follows:

$$P(w|\theta_{d_i}) = \begin{cases} (1-\lambda_{d_i})P(w|\tilde{\theta}_{d_i}), & \text{if } P(w|\tilde{\theta}_{d_i}) > 0 \\ \lambda_{d_i}\frac{P(w|\tilde{\theta}_D)}{U_{d_i}}, & \text{otherwise,} \end{cases}$$

where $U_{d_i} = 1 - \sum_{w \in d_i} P(w|\tilde{\theta}_D)$ is a normalization factor that is precomputed when the distribution for d_i is constructed. The discount factor λ_{d_i} indicates how much probability mass to reserve for unseen words. For pseudo-Good-Turing, it is

$$\lambda_{d_i} = \frac{|w \in d_i \text{ s.t. count}(w \in d_i) = 1|}{|w \in d_i|},$$

i.e. the fraction of words seen once in d_i.

We experimented with other smoothing methods, including Jelinek-Mercer and Dirichlet smoothing. A disadvantage of these latter two methods is that they have an additional tuning parameter to which their performance is highly sensitive, and even with optimal parameter settings neither consistently outperformed pseudo-Good-Turing. We also found no consistent improvement from using interpolation in place of backoff.

We also follow W&B in using Kullback-Leibler (KL) divergence as the similarity metric, since it outperformed both naive Bayes classification probability and cosine similarity:

$$KL(\theta_{d_i}||\theta_{d_j}) = \sum_k \theta_{d_i}(k) \log \frac{\theta_{d_i}(k)}{\theta_{d_j}(k)}.$$

The motivation for computing similarity with KL is that it is a measure of how well each document in the labeled set explains the word distribution found in the test document.

[2] https://github.com/utcompling/textgrounder/wiki/RollerEtAl_EMNLP2012

4 Collapsing Documents with an Adaptive Grid

In the previous section, we used the term "document" loosely when speaking of training documents. A simplistic approach might indeed involve comparing a test document to each training document. However, in the winner-takes-all model described above, we can rely only on the result of comparing with the single best training document, which may not contain enough information to make a good prediction.

A standard strategy to deal with this problem is to collapse groups of geographically nearby documents into larger pseudo-documents. This also has the advantage of reducing the computation time, as fewer training documents need to be compared against. Formally, this involves partitioning the training documents into a set of sets of documents $G = \{g_1 \ldots g_n\}$. A collection $\tilde{\mathbf{d}}$ of pseudo-documents is formed from this set, such that the pseudo-document for a particular group g_i is simply the concatenation of the documents in the group:

$$\tilde{d}_{g_i} = \bigcup_{d_j \in g_i} d_j.$$

A location must be associated with each pseudo-document. This can be chosen based on the partitioning function itself or the locations of the documents in each group.

Both W&B and SMvZ use *uniform* grids consisting of cells of equal degree size to partition documents. We explore an alternative that uses k-d (k-dimensional) trees to construct a non-uniform grid that *adapts* to training sets of different sizes more gracefully. It ensures a roughly equal number of documents in each cell, which means that all pseudo-documents compete on similar footing with respect to the amount of training data.

W&B define the location for a cell to be its geographic *center*, while SMvZ only perform error analysis in terms of choosing the correct cell. We obtain consistently improved results using the *centroid* of the cell's documents, which takes into account where the documents are concentrated.

4.1 *k*-d Trees

A *k*-d tree is a space-partitioning data structure for storing points in k-dimensional space, which groups nearby points into buckets. As one moves down the tree, the space is split into smaller regions along chosen dimensions. In this way, it is a generalization of a binary search tree to multiple dimensions. The *k*-d tree was first introduced by Bentley (1975) and has since been applied to numerous problems, e.g. Barnes-Hut simulation (Anderson, 1999) and nearest-neighbors search (Friedman et al., 1977).

Partitioning geolocated documents using a *k*-d tree provides finer granularity in dense regions and coarser granularity elsewhere. For example, documents from Queens and Brooklyn may show significant cultural distinctions, while documents separated by the same distance in rural Montana may appear culturally identical. A uniform grid with large cells will mash Queens and Brooklyn together, while small cells will create unnecessarily sparse regions in Montana.

An important parameter for a *k*-d tree is its *bucket size*, which determines the maximum number of points (documents in our case) that a cell may contain. By varying the bucket size, the cells can be made fine- or coarse-grained.

4.2 Partitioning with a *k*-d Tree

For geolocation, we consider the surface of earth to be a 2-dimensional space (k=2) over latitude, longitude pairs. We form a *k*-d tree by a recursive procedure over the training data. Initially, all documents are placed in the root node of the tree. If the number of documents in the node exceeds the bucket size, the node is split into two nodes along a chosen split dimension and point. This procedure is recursively called on each of the new child nodes, and repeats until no node is overflowing. The resulting leaves of the *k*-d tree form a patchwork of rectangles which cover the entire earth.[3]

When splitting an overflowing node, the choice of splitting dimension and point can greatly impact the structure of the resulting *k*-d tree. Following Friedman et al. (1977), we choose to always split a node

[3] We note that the grid "rectangles" are actually trapezoids due to the nature of the latitude/longitude coordinate system. We assume the effect of this is negligible, since most documents are away from the poles, where distortion is the most extreme.

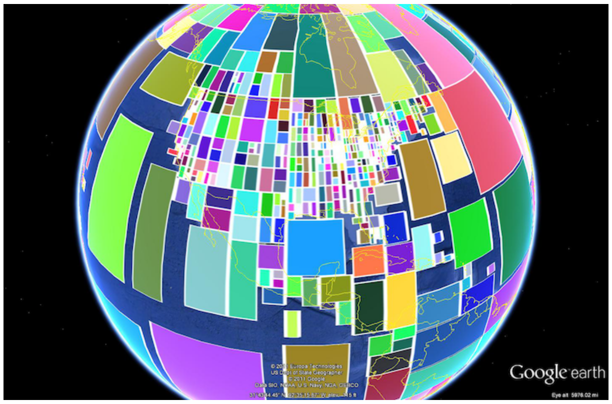


Figure 1: View of North America showing *k*-d leaves created from GeoWiki with a bucket size of 600 and the Midpoint method, as visualized in Google Earth.

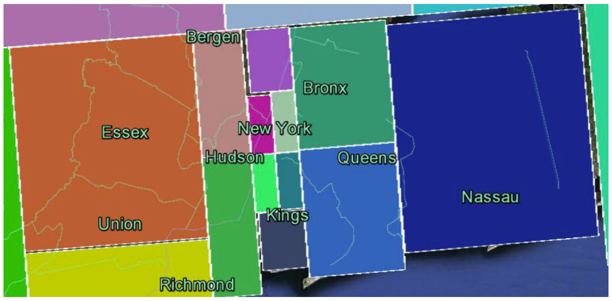


Figure 2: *k*-d leaves over the New York City and nearby areas from the same dataset and parameter settings as in Figure 1.

along the dimension exhibiting the greatest range of values. However, there still exist multiple methods for determining the split point, i.e. the point separating documents into "left" and "right" nodes. In this paper, we consider two possibilities for selecting this point: the **Midpoint** method, and the **Friedman** method. The latter splits at the median of all the points, resulting in an equal number of points in both the left and right nodes and a perfectly balanced *k*-d tree. The former splits at the midpoint between the two furthest points, allowing for a greater difference in the number of points in each bin. For geolocation, the Friedman splitting method will likely lead to less sparsity, and therefore more accurate cell selection. On the other hand, the Midpoint method is likely to draw more geographically desirable boundaries.

Figure 1 shows the leaves of the *k*-d tree formed over North America using the GeoWiki dataset, the Midpoint node division method, and a bucket size of 600. Figure 2 shows the leaves over New York City and its surrounding area for the same dataset and settings. More densely populated areas of the earth (which in turn tend to have more Wikipedia documents associated with them) contain smaller and more numerous leaf cells. The cells over Manhattan are significantly smaller than those of Queens, the Bronx, and East Jersey, even at such a coarse bucket size. Though the leaves of the *k*-d tree implicitly cover the entire surface of the earth, our illustrations limit the size of each box by its data, leaving gaps where no training documents exist.

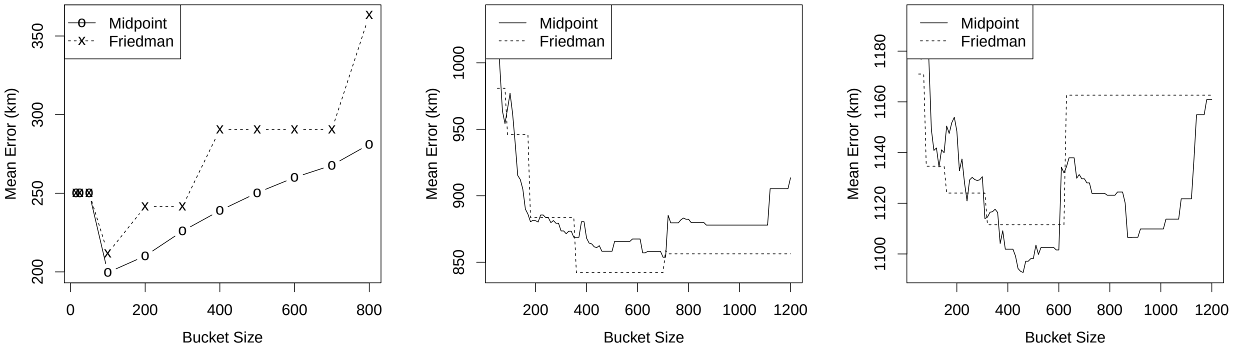
4.3 Selecting a Representative Location

W&B use the geographic **center** of a cell as the geolocation for the pseudo-document it represents. However, this ignores the fact that many cells will have imbalances in the dispersion of the documents they contain – typically, they will be clumpy, with documents clustering around areas of high population or activity. An alternative is to select the **centroid** of the locations of all the documents contained within a cell. Uniform grids with small cells are not especially sensitive to this choice since the absolute distance between a center or centroid prediction will not be great, and empty cells are simply discarded. Nonetheless, using the centroid has the benefit of making a uniform grid less sensitive to cell size, such that larger cells can be used more reliably – especially important when there are few training documents.

In contrast, when choosing representative locations for the leaves of a *k*-d tree, it is quite important to use the centroid because the leaves necessarily span the entire earth and none are discarded (since all have a roughly similar number of documents in them). Some areas with low document density are thus assigned very large cells, such as those over the oceans, as seen in Figures 1 and 2. Using the centroid allows these large leaves to be in the mix, while still predicting the locations in them that have the greatest document density.

5 Experimental Setup

Configurations. We experiment with several configurations of grids and representative locations.



(a) (b) (c)

Figure 3: Development set comparisons for (a) GEOWIKI, (b) GEOTEXT, and (c) UTGEO2011-SMALL.

W&B refers to a uniform grid and geographic-center location selection, **UNIFCENTROID** to a uniform grid with centroid location selection, **KDCENTROID** to a k-d tree grid with centroid location selection, and **UNIFKDCENTROID** to the union of pseudo-documents constructed by UNIFCENTROID and KDCENTROID.

We also provide two baselines, both of which are based on a uniform grid with centroid location selection. **RANDOM** predicts a grid cell chosen at random uniformly; **MOSTCOMMONCELL** always predicts the grid cell containing the most training documents. Note that a most-common k-d leaf baseline does not make sense, as all k-d leaves contain approximately the same number of documents.

Evaluation. We use three metrics to measure geolocation performance. The output of each experiment is a predicted coordinate for each test document. For each prediction, we compute the error distance along the surface of the earth to the gold coordinate. We report the **mean** and **median** of all such distances as in W&B and Eisenstein et al. (2011). We also report the **fraction of error distances less than 161 km**, corresponding to Cheng et al. (2010)'s measure of predictions within 100 miles of the true location. This third measure can reveal differences between models not obvious from just mean and median.

6 Results

This section provides results for the datasets described previously: GEOWIKI, GEOTEXT, UTGEO2011-LARGE and UTGEO2011-SMALL. We first give details for how we tuned parameters and algorithmic choices using the development sets, and then provide performance on the test sets based on these determinations.

6.1 Tuning

The specific parameters are (1) the partition location method; (2) the bucket size for k-d partitioning; (3) the node division method for k-d partitioning; (4) the degree size for uniform grid partitioning. We tune with respect to mean error, like W&B.

Partition Location Method. Development set results show that the centroid always performs better than the center for all datasets, typically by a wide margin (especially for large partition sizes). To save space, we do not provide details, but point the reader to the differences in test set results between W&B and UNIFCENTROID (which are identical except that the former uses the center and the latter uses the centroid) in Tables 1 and 2. All further parameter tuning is done using the centroid method.

***k*-d Tree Bucket Size.** Bucket size should not be too large as a proportion of the total number of training documents. Larger bucket sizes tend to produce larger leaves, so documents in a partition will have a higher average distance to the center or centroid point. This will result in predictions being made at too coarse a granularity, greatly limiting obtainable precision even when the correct leaf is chosen.

Conversely, small bucket sizes lead to fewer training documents per partition. A bucket size of one reduces to the situation where no pseudo-documents are used. While this might work well if location prediction is done using the kNNs for a test document, it

Test dataset	GeoWiki				GeoText			
Method	Parameters	Mean	Med.	Acc.	Parameters	Mean	Med.	Acc.
Random	0.1°	7056	7145	0.3	5°	2008	1866	1.6
MostCommonCell	0.1°	4265	2193	5.0	5°	1158	757	31.3
Eisenstein et al.	-	-	-	-	-	**845**	501	-
Wing & Baldridge	0.1°	221	11.8	-	5°	967	479	-
UnifCentroid	0.1°	181	**11.0**	**90.3**	5°	897	**432**	**35.9**
KdCentroid	B100, Midpt.	192	22.5	87.9	B530, Fried.	958	549	35.3
UnifKdCentroid	0.1°, B100, Midpt.	**176**	13.4	90.3	5°, B530, Fried.	890	473	34.1

Table 1: Performance on the held-out test sets of GeoWiki and GeoText, comparing to the results of Wing and Baldridge (2011) and Eisenstein et al. (2011).

is likely to perform very poorly for the 1NN rule we adopt. It would also require efficient similarity comparisons, using techniques such as locality-sensitive hashing (Kulis and Grauman, 2009).

The graphs in Figure 3 show development set performance when varying bucket size. For GeoWiki and UTGeo2011-Large (not shown), increments of 100 were used, but for the smaller GeoText and UTGeo2011-Small, more fine-grained increments of 10 were used. In the case of plateaus, as was common with the Friedman method, we chose the middle of the plateau as the bucket size. Overall, we found optimal bucket sizes of 100 for GeoWiki, 530 for GeoText, 460 for UTGeo2011-Small, and 1050 for UTGeo2011-Large. That the Wikipedia data requires a smaller bucket size is unsurprising: the documents themselves are generally longer and there are many more of them, so a small bucket size provides good coverage and granularity without sacrificing the ability to estimate good language models for each partition.

Node Division Method. The graphs in Figure 3 also display the difference between the two splitting methods. Midpoint is clearly better for GeoWiki, while Friedman is better for GeoText in the range of bucket sizes producing the best results. Friedman is best for UTGeo2011-Large (not shown), but Midpoint is best for UTGeo2011-Small.

These results only partly confirm our expectations. We expected Friedman to perform better on smaller datasets, as it distributes the documents evenly and avoids many sparsity issues. We expected Midpoint to win on larger datasets, where all nodes receive plentiful data and the *k*-d tree would choose more representative geographical boundaries.

Cell Size. Following W&B, we choose a cell degree size of 0.1° for GeoWiki, and a cell degree size of 5.0° for GeoText. For UTGeo2011-Large and UTGeo2011-Small, we follow the procedure of W&B, trying sizes $0.1°, 0.5°$, 1.0°, 5.0°, and 10.0°, selecting the one which performed best on the development set. For UTGeo2011-Small, this resulted in coarse cells of 10.0°, while for UTGeo2011-Large, cell sizes of 0.1° were best.

With these tuned parameters, the average number of training tokens per *k*-d leaf was approximately 26k for GeoWiki, 197k for GeoText, 250k for UTGeo2011-Small, and 954k for UTGeo2011-Large.

6.2 Held-out Test Sets

Table 1 shows the performance on the test sets of GeoWiki and GeoText of the different configurations, along with that of W&B and Eisenstein et al. (2011) where possible. The results obtained by W&B on GeoWiki are already very strong, but we do see a clear improvement by changing from the center-based locations for pseudo-documents they used to the centroid-based locations we employ: mean error drops from 221 km to 181 km, and median error from 11.8 km to 11.0 km. Also, we reduce the mean error further to 176 km for the configuration that combines the uniform grid and the *k*-d partitions, though at the cost of increasing median error somewhat. The 161 km accuracy is around 90% for all configurations, indicating that the general language modeling approach we employ is very

Test dataset	UTGeo2011							
Training dataset	UTGeo2011-Small				UTGeo2011-Large			
Method	Parameters	Mean	Med.	Acc.	Parameters	Mean	Med.	Acc.
Random	10°	1975	1833	2.3	0.1°	1627	1381	2.0
MostCommonCell	10°	1522	1186	9.3	0.1°	1525	1185	11.8
Wing & Baldridge	10°	1223	825	3.4	0.1°	956	570	30.9
UnifCentroid	10°	1147	782	12.3	0.1°	956	570	30.9
KdCentroid	B460, Midpt.	1098	733	**18.1**	B1050, Fried.	**860**	**463**	**34.6**
UnifKdCentroid	10°, B460, Midpt.	**1080**	**723**	**18.1**	0.1°, B1050, Fried.	913	532	33.0

Table 2: Performance on the held-out test set of UTGeo2011 for different configurations trained on UTGeo2011-Small (comparable in size to GeoText) and UTGeo2011-Large. The numbers given for W&B were produced from their implementation, and correspond to uniform grid partitioning with locations from centers rather than centroids.

robust for fact-oriented texts that are rich in explicit toponyms and geographically relevant named entities.

For GeoText, the results show that the uniform grid with centroid locations is the most effective of our configurations. It improves on Eisenstein et al. (2011) by 69 km with respect to median error, but has 52 km worse performance than their model with respect to mean error. This indicates that our model is generally more accurate, but that it is comparatively more wildly off on some documents. Their model is a sophisticated one that attempts to build detailed models of the geographic linguistic variation found in the dataset. Dialectal cues are actually the most powerful ones in the GeoText dataset, and it seems our general approach of winner-takes-all (1NN) hurts performance in this respect, especially with a very small training set.

Table 2 shows the performance on the test set of UTGeo2011 with the UTGeo2011-Small and UTGeo2011-Large training sets. (Performance for W&B is obtained from their code.[4]) With the small training set, error is worse than with GeoText, reflecting the wider geographic scope of UTGeo2011. KdCentroid is much more effective than the uniform grids, but combining it with the uniform grid in UnifKdCentroid edges it out by a small amount. More interestingly, KdCentroid is the strongest on all measures when using the large training set, beating UnifCentroid by an even larger margin for mean and median error than with the small training set. The bucket size used with the large training set is double that for the small one, but there are many more leaves created since there are 42 times more training documents. With the extra data, the model is able to adapt better to the dispersion of documents and still have strong language models for each leaf that work well even with our greedy winner-takes-all decision method.

Note that the accuracy measurements for all UTGeo2011 experiments are substantially lower than those reported by Cheng et al. (2010), who report a best accuracy within 100 miles of 51%. While UTGeo2011-Large contains a substantially larger number of tweets, Cheng et al. (2010) limit themselves to users with at least 1,000 tweets, while we have an average of 85 tweets per user. Their reported mean error distance of 862 km (versus our best mean of 860 km on UTGeo2011-Large) indicates that their performance is hurt by a relatively small number of extremely incorrect guesses, as ours appears to be.

Figure 4 provides a learning curve on UTGeo2011's development set for KdCentroid. Performance improves greatly with more data, indicating that GeoText performance would also improve with more training data. Parameters, especially bucket size, need retuning as data increases, which we hope to estimate automatically in future work

Finally, we note that the KdCentroid method was faster than other methods. While UnifCentroid took nearly 19 hours to complete the test run on GeoWiki (approximately

[4] `https://bitbucket.org/utcompling/textgrounder/wiki/WingBaldridge2011`

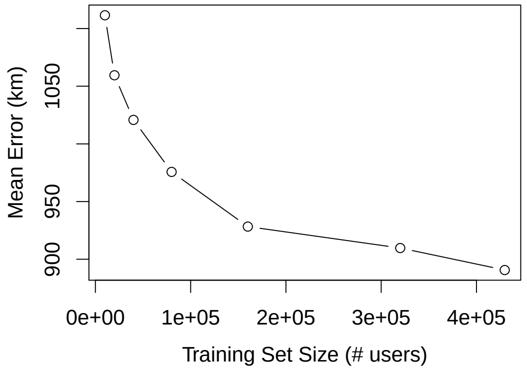


Figure 4: Learning curve of KdCentroid on the UTGeo2011 development set.

1.38 seconds per test document), KdCentroid took only 80 minutes (.078 s/doc). Similarly, UnifCentroid took about 67 minutes to run on UTGeo2011-Large (0.34 s/doc), but KdCentroid took only 27 minutes (0.014 s/doc). Generally, the KdCentroid partitioning results in fewer cells, and therefore fewer KL-divergence comparisons. As expected, the UnifKdCentroid model needs as much time as the two together, taking roughly 21 hours for GeoWiki (1.52 s/doc) and 85 minutes for UTGeo2011-Large (0.36 s/doc).

7 Discussion

7.1 Error Analysis

We examine some of the greatest error distances to better understand and improve our models. In many cases, landmarks in Australia or New Zealand are predicted in European locations with similarly-named landmarks, or vice versa — e.g. the *Theatre Royal, Hobart* in Australia is predicted to be in London's theater district, and the *Embassy of Australia, Paris* is predicted to be in the capital city of Australia. Thus, our model may be inadvertently capturing what Clements et al. (2010) call *wormholes*, places that are related but not necessarily adjacent.

Some of the other large errors stem from incorrect gold labels, in particular due to sign errors in latitude or longitude, which can place documents 10,000 or more km from their correct locations.

Word	Error	Word	Error
paramus	78	6100	130
ludlow	79	figueroa	133
355	99	dundas	138
ctfuuu	101	120th	139
74th	105	mississauga	140
5701	105	pulaski	144
bloomingdale	122	cucina	146
covina	133	56th	153
lawrenceville	122	403	157
ctfuuuu	124	428	161

Table 3: The 20 words with least average error (km) in the UTGeo2011 development set, trained on the UTGeo2011-Small training set, using the KdCentroid approach with our best parameters. Only words that occur in at least 10 documents are shown.

Word	Error	Word	Error
seniorpastor	1.1	KS01	2.4
prebendary	1.6	Keio	2.5
Wornham	1.7	Vrah	2.5
Owings	1.9	overspill	2.5
Londoners	2.0	Oriel	2.5
Sandringham	2.1	Holywell	2.6
Sheffield's	2.2	\'vr&h	2.6
Oxford's	2.2	operetta	2.6
Belair	2.3	Supertram	2.6
Beckton	2.4	Chanel	2.7

Table 4: Top 20 words with the least average error (km) in the GeoWiki development set, using the UnifKdCentroid approach with our best parameters. Only words occurring in at least 10 documents are shown.

7.2 Most Predictive Words

Our approach relies on the idea that the use of certain words correlates with a Twitter user or Wikipedia article's location. To investigate which words tend to be good indicators of location, we computed, for each word in a development set, the average error distance of documents containing that word. Table 3 gives the 20 words with the least error, among those that occur in at least 10 documents (users), for the UTGeo2011 development set, trained on UTGeo2011-Small.

Many of the best words are town names (*paramus*, *ludlow*, *bloomingdale*), street names (*74th*, *figueroa*, *120th*), area codes (*403*), and street numbers (*5701*, *6100*). All are highly locatable terms, as we would expect. Many of the street addresses are due to check-ins with the location-based social networking service Foursquare (e.g. the tweet *I'm at Starbucks (7301 164th Ave NE, Redmond Town Center, Redmond)*), where the user is literally broadcasting his or her location. The token *ctfuuu(u)*—an elongation of the internet abbreviation *ctfu*, or *cracking the fuck up*—is a dialectal or stylistic feature highly indicative of the Washington, D.C. area.

Similarly, several place names (*Wornham*, *Belair*, *Holywell*) appear in GEOWIKI. *Operetta*s are a cultural phenomenon largely associated with France, Germany, and England and particularly with specific theaters in these countries. However, other highly specific tokens such as *KS01* have a very low average error because they occur in few documents and are thus highly unambiguous indicators of location. Other terms, like *seniorpastor* and *\'vr&h*, are due to extraction errors in the dataset created by W&B, and are carried along because of a high correlation with specific documents.

8 Conclusion

We have shown how to construct an adaptive grid with *k*-d trees that enables robust text geolocation and scales well to large training sets. It will be interesting to consider how it interacts with other strategies for improving the IR-based approach. For example, the pseudo-document word distributions can be smoothed based on nearby documents or on the structure of the *k*-d tree itself. Integrating our system with topic models or Bayesian methods would likely provide more insight with regard to the most discriminative and geolocatable words. We also expect predicting locations based on multiple most similar documents (kNN) to be more effective in predicting document location, as the second and third most similar training documents together may sometimes be a better estimation of its distribution than just the first alone. Employing *k* Nearest Neighbors also allows for more sophisticated methods of location estimation than a single leaf's centroid. Other possibilities include constructing multiple *k*-d trees using random subsets of the training data to reduce sensitivity to the bucket size.

In this article, we have considered each user in isolation. However, Liben-Nowell et al. (2005) show that roughly 70% of social network links can be described using geographic information and that the probability of a social link is inversely proportional to geographic distance. Backstrom et al. (2010) verify these results on a much larger scale using geolocated Facebook profiles: their algorithm geolocates users with only the social graph and significantly outperforms IP-based geolocation systems. Given that both Twitter and Wikipedia have rich, linked document/user graphs, a natural extension to our work here will be to combine text and network prediction for geolocation. Sadilek et al. (2012) also show that a combination of textual and social data can accurately geolocate individual tweets when scope is limited to a single city.

Tweets are temporally ordered and the geographic distance between consecutive tweeting events is constrained by the author's movement. For tweet-level geolocation, it will be useful to build on work in geolocation that considers the temporal dimension (Chen and Grauman, 2011; Kalogerakis et al., 2009; Sadilek et al., 2012) to make better predictions for documents/images that are surrounded by others with excellent cues, but which are hard to resolve themselves.

9 Acknowledgments

We would like to thank Matt Lease and the three anonymous reviewers for their feedback. This research was supported by a grant from the Morris Memorial Trust Fund of the New York Community Trust.

References

Richard J. Anderson. 1999. Tree data structures for n-body simulation. *SIAM Journal on Computing*, 28(6):1923–1940.

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*, pages 61–70.

Jon Louis Bentley. 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517.

Chao-Yeh Chen and Kristen Grauman. 2011. Clues from the beaten path: Location estimation with bursty sequences of tourist photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1569–1576.

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 on Information and Knowledge Management*, pages 759–768.

Martin Clements, Pavel Serdyukov, Arjen P. de Vries, and Marcel J.T. Reinders. 2010. Finding wormholes with flickr geotags. In *Proceedings of the 32nd European Conference on Information Retrieval*, pages 658–661.

Sebastian Cobarrubias. 2009. *Mapping machines: activist cartographies of the border and labor lands of Europe*. Ph.D. thesis, University of North Carolina at Chapel Hill.

Dorin Comaniciu and Peter Meer. 2002. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619.

Junyan Ding, Luis Gravano, and Narayanan Shivakumar. 2000. Computing geographical scopes of web resources. In *Proceedings of the 26th International Conference on Very Large Data Bases*, pages 545–556.

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.

Jacon Eisenstein, Ahmed Ahmed, and Eric P. Xing. 2011. Sparse additive generative models of text. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1041–1048.

Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel. 1977. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3:209–226.

James Hays and Alexei A. Efros. 2008. im2gps: estimating geographic information from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Evangelos Kalogerakis, Olga Vesselova, James Hays, Alexei Efros, and Aaron Hertzmann. 2009. Image sequence geolocation with human travel priors. In *Proceedings of the IEEE 12th International Conference on Computer Vision*, pages 253–260.

Sheila Kinsella, Vanessa Murdock, and Neil O'Hare. 2011. "I'm eating a sandwich in Glasgow": Modeling locations with tweets. In *Proceedings of the 3rd International Workshop on Search and Mining User-generated Contents*, pages 61–68.

Brian Kulis and Kristen Grauman. 2009. Kernelized locality-sensitive hashing for scalable image search. In *Proceedings of the 12th International Conference on Computer Vision*, pages 2130–2137.

David Liben-Nowell, Jasmine Novak, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. 2005. Geographic routing in social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(33):11623–11628.

Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281.

Adam Sadilek, Henry Kautz, and Jeffrey P. Bigham. 2012. Finding your friends and following them to where you are. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, pages 723–732.

Jagan Sankaranarayanan, Hanan Samet, Benjamin E. Teitler, Michael D. Lieberman, and Jon Sperling. 2009. Twitterstand: news in tweets. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 42–51.

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*, pages 484–491.

Benjamin 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*, pages 955–964.

Chengxiang Zhai and John Lafferty. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 334–342.