An Effective Approach for Geolocation Prediction in Twitter Streams Using Clustering Based Discretization

Nghia Duong-Trung, Nicolas Schilling, Lucas Rego Drumond, Lars Schmidt-Thieme

Abstract Micro-blogging services, such as Twitter, have provided an indispensable channel to communicate, access, and exchange current affairs. Understanding the dynamics of users behavior and their geographical location is key to providing services such as event detection, geo-aware recommendation and local search. The geographical location prediction problem we address is to predict the geolocation of a user based on textual tweets. In this paper, we develop a clustering based discretization approach which is an effective combination of three well-known machine learning algorithms, e.g. K-means clustering, support vector machines, and K-nearest neighbor, to tackle the task of geolocation prediction in Twitter streams. Our empirical results indicate that our approach outperforms previous attempts on a publicly available dataset and that it achieves state-of-the-art performance.

Nghia Duong-Trung
University of Hildesheim, Universitätsplatz 1,
✉ duongn@ismll.uni-hildesheim.de

Nicolas Schilling
University of Hildesheim, Universitätsplatz 1,
✉ schilling@ismll.uni-hildesheim.de

Lucas Rego Drumond
University of Hildesheim, Universitätsplatz 1,
✉ ldrumond@ismll.uni-hildesheim.de

Lars Schmidt-Thieme
University of Hildesheim, Universitätsplatz 1,
✉ schmidt-thieme@ismll.uni-hildesheim.de

ARCHIVES OF DATA SCIENCE (ONLINE FIRST) DOI 10.5445/KSP/1000058749/13
KIT SCIENTIFIC PUBLISHING ISSN 2363-9881
Vol. 2, No. 1, 2017

1 Introduction

Given the increasing use of mobile devices to access social networks, people post and update statuses, follow and discuss events within a local and global spectrum, comment and share their daily lives with friends, and even publicize individual expression on local political and social trends. The contents are very rich and dynamic; consequently, representing people's thoughts at a given time and vicinity. Understanding the current user's geographical location, e.g. latitude and longitude pairs of physical coordinates, enables providing services such as local search, advertisements and event detection.

From the observation of many users of Twitter supplying exact geographical coordinates from GPS-enabled devices, Eisenstein et al (2010) concatenate users tweets into a single representative *document* that is used to predict a user's geolocation. The authors propose a multi-level generative model with the aim of predicting the user's geographical location based on raw text alone. The idea is to analyze lexical variation by topics and geography interaction influenced by latent regional indicators that lately adapt to location indicative words (Bo and Baldwin, 2012; Han et al, 2014). Then, based on the lexical interactions, the model splits the geographical space into coherent linguistic communities, and finally, infers the user's physical coordinates.

Kinsella et al (2011) investigate inferring language models to predict the location of an individual tweet and/or location of a user at varying levels of granularity, from zip code to the country level. Wing and Baldridge (2011) develop a geodesic grid representation, a grid of square cells of equal degree, at different levels by using a vocabulary distribution. Hong et al (2012) model the tweets' geolocation based on the topical diversity, the geographical diversity, and the interest distribution of a user. Chandra et al (2011) and Jurgens (2013) exploit the idea of inferring locations based on observable social relationships and a small amount of ground truth locations.

In this paper, we took those aforementioned works under scrutiny and focus on the geolocation prediction by developing a clustering based discretization method. Unlike many previous approaches that employed different kinds of language models, we investigate a combination of well-known machine learning algorithms, e.g. K-means clustering (K-means), support vector machines (SVM) and K-nearest neighbor (k-nn) to effectively predict the geographical location of users based on their textual tweets. The results of our proposed approach have outperformed state-of-the-art approaches on the same dataset.

The paper is organized as follows. In Sect. 2, we are going to briefly discuss related work on the user geolocation prediction. Then, we present the problem formulation, the data preprocessing and the proposed approach in Sect. 3. In Sect. 4, we present two evaluation metrics, the experimental results and compare our results with previous approaches. Finally, we conclude our work and discuss future improvements in Sect. 5.

2 Related Work

Eisenstein et al (2010) introduce a geographic topic model that incorporates two sources of lexical variation: topic and geographical regions. The model uses the correlations between global and local topics. The local topics are derived from global topics to generate terms. Terms are conditional on the latent region variables. They concatenate all messages of a user into a single *document* in order to predict that user's geolocation. Cascading topic models are used to generate base topics that are selected by per-token hidden variables as in latent Dirichlet allocation (Blei et al, 2003; Wang and Grimson, 2008). The geographic topic model then generates region variation by selection of latent variables per-user. This unsupervised methodology assumes that topics and regions interact to generate the observed lexical frequencies.

The Sparse Additive Generative model (SAGE) has been introduced by Eisenstein et al (2011) and is an improvement upon their previous work. In their paper, the authors propose a model for discrete distributions via natural parameters and log-frequency differences. They address some problems with Dirichlet-multinomial generative models, e.g. the inference cost, the over-parametrization and the lack of exploiting sparsity. Wing and Baldridge (2011) investigate the construction of a discrete grid representation of the earth's surface to automatically identify the location of a document based only on its text. Their approach finds a grid cell whose word distribution has the smallest Kullback-Leibler divergence (Zhai and Lafferty, 2001). In each grid cell, the model calculates a word distribution. Then the similarity between a test document's words and that of each cell is computed using Naive Bayes. The predicted location is the center of the most similar cell.

Hong et al (2012) propose a generative model that uses both statistical topic models and sparse coding techniques by addressing the problem of modeling geographical topical patterns. Their model is a significant improvement upon the SAGE model, by additionally adopting personal preferences. They argue that a tweet depends on both location and topic: language variations differ among regions; locations affect on topics mentioned and words used; users tend to appear in a handful geographical locations. The authors assume that each tweet is heavily influenced by three types of language models simultaneously. These are the background language model, the per-region language model, and the topical language model. After a number of latent regions are clustered, the model generates locations, topics and terms used in the tweet. The authors also demonstrate that the model can discover the relationship between interesting topics and users' geographical patterns. The prediction takes two steps. At the first step, a region that maximizes the likelihood of a test tweet is drawn. In the second step, the mean location of the selected region is assigned as the predicted geolocation of that test tweet. They manually set a fixed number of topics and latent regions.

3 Our Proposed Approach

In Sect. 3.1, we first discuss about the geolocation prediction that we will address in this paper as well as present the problem formulation and notation that is used throughout the paper. Then in Sect. 3.2, we discuss the data preprocessing techniques. Following in Sect. 3.3, we present the proposed method and how it works.

3.1 Problem Formulation and Notation

As we have already mentioned in the above section, the geolocation prediction that we want to address here is to predict a user's geolocation based on his textual tweets. But instead of exploiting every tweet, all of a users' tweets are concatenated into one single representative document. Then this document is used as the input and the geographical location of it is the output of the system. We have m, v and w documents in the training, test and validation data respectively, and n features/tokens/words describing each document. Each document is annotated with a geolocation $\mathbf{y} \in \mathbb{R}^2$, $\mathbf{y} = (y^{lat}, y^{lon})$ where $y^{lat} \in \mathbb{R}$ is the latitude and $y^{lon} \in \mathbb{R}$ is the longitude. Given some training data $X^{train} \in \mathbb{R}^{m \times n}$, and the respective labels $Y^{train} \in \mathbb{R}^{m \times 2}$, we aim to find a model $f : \mathbb{R}^n \rightarrow$

$\mathbb{R}^2$ such that for some test data $X^{test} \in \mathbb{R}^{v \times n}$, the error

$$\sum_{i=1}^{v} d(f(X_i^{test}), Y_i^{test}) \tag{1}$$

is minimal, where $Y^{test} \in \mathbb{R}^{v \times 2}$ is the true geolocation matrix and d is a distance metric which is either the Euclidean distance (see Equation 2) or the Haversine distance (see Equation 3).

3.2 Data Preprocessing

As we are working with textual data, we propose the data preprocessing as follows. All given documents are converted from sparse vectors of tokens into sparse vectors of bag-of-words representation with term frequency - inverse document frequency (TF-IDF) weights (Sparck Jones, 1972; Robertson, 2004). In this way, we discard language grammar structure, token's order, and part-of-speech. It is intuitive that the frequency with which a token appears in a document could indicate the extent that the document pertains to that token. The TF-IDF score reflects how important a token is to a document. The more common a token is to many documents, the more penalization it gets. The training and test sets' TF-IDF weights are computed separately to respect their own tokens' scores.

3.3 Proposed Method

From the observation that users tend to appear in a handful geographical locations, Hong et al (2012) examine the influence of language models over geolocation. However, we find this approach is complicated and computationally expensive, instead we can directly capture users' appearance by clustering their geographical location. Consequently, we develop a clustering based discretization (CBD) approach because of the following reasons: Firstly, it effectively captures the distribution of physical locations. Secondly, it is more effective than the geodesic grid representation proposed by Wing and Baldridge (2011), as one disadvantage of their grid representation is that they apply equally sized grid cells. Therefore, a cell may capture a dense of users' appearances but another cell is empty. In other words, from the observation of the geographical appearance of users, we first learn from the training set to identify a geolocation distribution. Then for a new data point, we need to estimate to which location it belongs. Finally, the geolocation prediction is conducted with a distance measurement given the predicted location.

K-means clustering

The target geolocation $\mathbf{y}$ consists of two labels y^{lat} and y^{lon}. The basic idea in the first step is to transform a multi-target prediction task into a multi-class classification task. The idea of an equally squared grid is not effective (Wing and Baldridge, 2011) although we know in advance that the geographic coordinates of documents are spread within the USA for the implemented dataset. In order to find regions of interest, we cluster the documents in the training set using K-means clustering (Hartigan and Wong, 1979) that dynamically captures the geographical location distribution. At the end of this step, we have a cluster assignment vector $\mathbf{c} \in \{1,\ldots,K\}^m$, where the i-th element c_i contains the cluster assigned to the i-th instance based on its geo-location $\mathbf{y}_i$.

Support vector machines

Now that we have identified clusters, we need to learn a model on X^{train} and $\mathbf{c}$ in order to map the test instances to those clusters. For that reason, we use a classifier which has $\mathbf{c}$ as the target and X^{train} as the predictors domain. From now on, the task of geolocation prediction can be treated as a multi-class classification problem. The SVM $g : \mathbb{R}^n \rightarrow \{1,\ldots,K\}$ with L2 regularization (Joachims, 1998) is trained on the dataset associated with corresponding clusters $\mathbf{c}$. The SVM is chosen because it is one of the state-of-the-art algorithms for text classification.

K-nearest neighbor

Once we have estimated to which cluster c_i a test instance X_i^{test} should belong, there are a couple of strategies for predicting the geolocation. We could take the mean or the median location of the cluster c_i as the prediction. Otherwise, we could also compare the similarity between the test instance X_i^{test} and all training instances in the cluster c_i. During experiments, we find all strategies achieve good results and the latter strategy is the most effective. The predicted geolocation $\hat{\mathbf{y}}_i^{test}$ is that of the nearest neighbor in the same cluster $g(X_i^{test})$. The distance between two instances is basically the Euclidean distance between two vectors X_i^{test} and X_j^{train}. The physical coordinates of X_i^{test} are predicted using k-nn regression (Peterson, 2009) on all the training instances X_j^{train} belonging to $g(X_i^{test})$. Consequently, depending on which strategy is chosen, we develop three variants of our proposed method. We denote them as CBD_Mean, CBD_Median, and CBD_k-nn in case of taking the mean location, the median location and, the k-nn distance as prediction, respectively.

Those aforementioned steps yield the pseudocode (see Algorithm 1).

Algorithm 1 the CBD_k-nn algorithm

INPUT: X^{train}, X^{test}, Y^{train}, cost s, number of clusters K, number of nearest neighbors k

```
{Step 1: K-means clustering}
$\mathbf{c} \leftarrow$ K-means$(Y^{train}, K)$
{Step 2: SVM}
$g \leftarrow$ SVM$(X^{train}, s, \mathbf{c})$
{Step 3: k-nearest neighbor}
for i = 1 ... v do
    $c_i \leftarrow g(X_i^{test})$
    $X, Y \leftarrow \{(X_j^{train}, Y_j^{train}) | g(X_j^{train}) = c_i\}$
    $\hat{\mathbf{y}}_i \leftarrow$ k-nnRegression$(X, Y, X_i^{test})$
end for
return $\hat{\mathbf{y}}_i$
```

4 Experimental Results

In Sect. 4.1, we first present the experimental dataset. Then in Sect. 4.2, we discuss two evaluation metrics to measure how good are the methods used in this paper and in previous approaches. We discuss experiment results and make comparison with related work in Sect. 4.3. Finally, we provide our preprocessed version of data as well as the employed hyperparameters to promote reproducibility of our work in Sect. 4.4.

4.1 Dataset

The training, the validation, and the test sets that we consider in our evaluation are originally described by Eisenstein et al (2010), called the CMU[1] dataset. We reuse the same raw training, validation, and test sets as well as the same dictionary. The dataset comprises the tweets gathered from the microblog website Twitter, via its official API, in the first week of March 2010 from the "Gardenhose" sample stream. Some important facts about the dataset are summarized as follows: only messages that are tagged with physical coordinates pairs, e.g. latitude and longitude, from a mobile client, and whose users wrote at least 20 messages over the observed period were collected. Emoticons, emoji, blocks of punctuation, @-mentioned words and other symbols as tokens are preserved. In total, the dataset comprises 4.7 million word tokens, 377.616 tweets, 9.475 users and 39,85 tweets/user. Figure $1(a)$ summarizes the top 30 tokens in the test set and Fig. $1(b)$ presents a histogram of documents' length represented by their number of tokens.

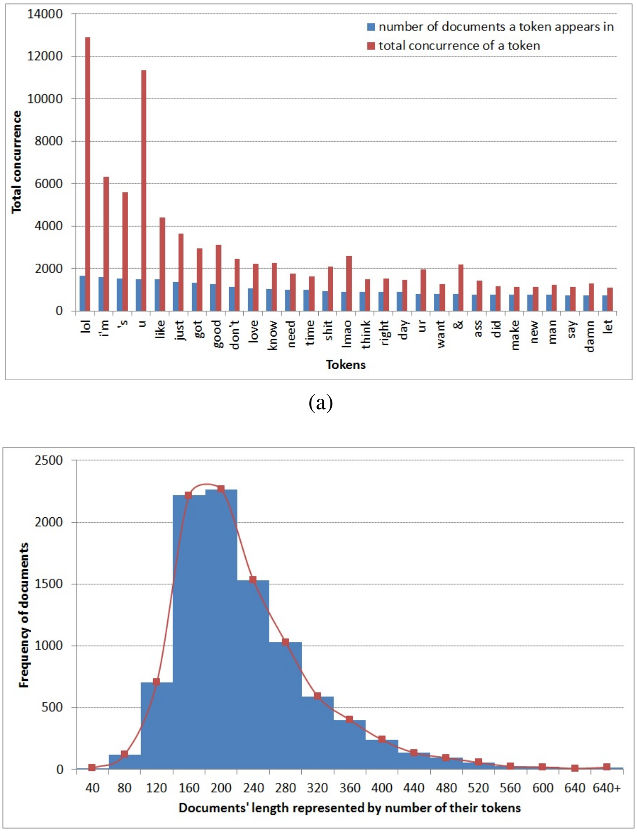
4.2 Evaluation Metrics

Given two points on the earth's surface represented by their latitude and longitude coordinates, the easiest way to calculate the physical distance between them is the Euclidean distance. That means we treat the earth's surface as a flat plane in two dimensional space. Hong et al (2012) report this metric in their work. Another metric is the Haversine distance that treats the earth's surface as an ellipsoidal shape. The other methods we compare to all use the Haversine distance.

Euclidean distance. We calculate the Euclidean distance $d_E : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}_+$ between two geographical pairs, for example the true geolocation $\mathbf{y}$ and the predicted geolocation $\hat{\mathbf{y}}$. We must convert $\mathbf{y}$ and $\hat{\mathbf{y}}$ into the Universal Transverse Mercator (Lampinen, 2001) coordinate system before calculating the Euclidean distance. The formula is stated as follows: for any particular two geographical pairs $\mathbf{y} = (y^{lat}, y^{lon})$ and $\hat{\mathbf{y}} = (\hat{y}^{lat}, \hat{y}^{lon})$ on the earth's surface, the Euclidean distance d_E is given by:

$$d_E(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{(\hat{y}^{lat} - y^{lat})^2 + (\hat{y}^{lon} - y^{lon})^2} \tag{2}$$

[1] http://www.ark.cs.cmu.edu/GeoText



(a)

(b)

Fig. 1: The graph (a) shows the top 30 tokens in term of their frequency in each document in the test set. The histogram (b) presents documents' lengths represented by number of their tokens.

Haversine distance. The Haversine distance $d_H : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}_+$ is the great circle distance between two geographical pairs. We compute the distance between two pairs by the Haversine formula (Robusto, 1957). The formula of the central angle between them is given by:

$$d_H(\mathbf{y}, \hat{\mathbf{y}}) = 2r\arcsin(\alpha), \tag{3}$$

where r is the radius of the earth. Because of the ellipsoidal shape of the earth, its radius varies from equator to poles. According to Decker (1986), the mean of the earth's radius $r = 6371$ km is taken. And α is defined as follows.

$$\alpha = \sqrt{\sin^2\left(\frac{|\hat{y}^{lat} - y^{lat}|}{2}\right) + \cos(y^{lat})\cos(\hat{y}^{lat})\sin^2\left(\frac{|\hat{y}^{lon} - y^{lon}|}{2}\right)} \qquad (4)$$

More specifically, the evaluation metrics are the median and the mean Haversine distances d_H and the median and the mean Euclidean distances d_E between the actual location $\mathbf{y}$ and the predicted location $\hat{\mathbf{y}}$. We apply a grid search mechanism to find the best value combination of all hyperparameters(see Sect. 4.4) that minimize the distance error $\sum_{i=1}^{w} d(f(X_i^{val}), Y_i^{val})$. For the reason of computational simplicity, the Euclidean distance is used for the model's optimization and the Haversine distance is used for calculating final distances.

4.3 Results and Comparison

We evaluate our proposed approach by conducting experiments on the `CMU` dataset and compare the results with Eisenstein et al (2010), Wing and Baldridge (2011), Eisenstein et al (2011) and Hong et al (2012). To the best of our knowledge, only the above papers have worked on the same training, validation and test splits and addressed the same geolocation prediction problem. We summarize our results with previous results in Table 1, reporting both the median and the mean Haversine errors.

From the results, we have shown that it is more effective to predict the user's geolocation by directly exploiting the location distribution instead of developing complex language models. Another advantage of the K-means clustering over the equally squared grid is that it takes less computation time because the number of clusters is much smaller than the number of the grid's cells. Additionally, it dynamically adapts to the location distribution despite of which dataset is evaluated.

In case of applying the k-nn strategy, our proposed approach works really well in terms of the median Haversine error as it outperforms previous approaches by achieving 326.47 km, while the median Euclidean error is 325.97 km. Admittedly, it does not work so well in terms of the mean Haversine error. With 852.94 km, it is not as good as the result of Eisenstein et al (2011) that achieved 845 km. One note is that Hong et al (2012) only reported the Euclidean error in comparison with previous approaches. They claimed that the difference between the two evaluation metrics is not significant. They also did not report the mean Haversine error.

As mentioned in Sect. 3.3, we also examine the effectiveness of k-nn by comparing its performance with those of other variants. We replace k-nn by simply taking the median and the mean geolocation of the predicted cluster as the prediction. We report them as CBD_Mean, CBD_Median and CBD_k-nn in case of taking the mean location, the median location and the k-nn distance as the prediction respectively. It is understandable that the CBD_k-nn is better because in this strategy, we semantically compare documents' content together to find the most similar one. We also try applying k-nn with different number of nearest neighbors $k \in \{1, 2, 3, 4, 5\}$, where 1-nn achieves the best results.

There are three hyperparameters that needed to be tuned, at first the number of clusters K, secondly the cost s and finally the number of nearest neighbors k. Figure 2 shows the hyperparameters search sensitivity analysis in terms of the median Haversine error against changes in the number of clusters K. For a smaller number of K, it is easier task for the SVM and harder task for the k-nn, and vice versa. For example, when the number of cluster equals $K = 1$, all instances belong to the same cluster which is equal to applying k-nn over the whole dataset. More precisely, Fig. 2 presents the mean and median error distances by applying the Euclidean and Haversine equations. The two upper lines are the mean error distances while the two lower lines are the median error distances. As we can see that there are no significant differences between applying Euclidean and Haversine equations.

Table 1: Comparison of models on the `CMU` dataset. All numbers are kilometers. We do not re-implement their models and only report Haversine errors from the corresponding papers.

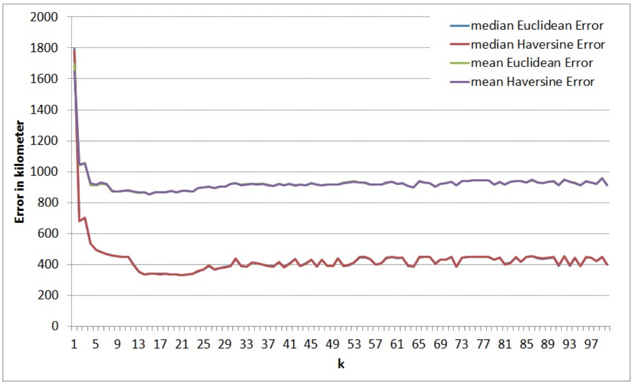
Models	median Haversine error	mean Haversine error
Eisenstein et al (2010)	494	900
Wing and Baldridge (2011)	479	967
Eisenstein et al (2011)	501	845
Hong et al (2012)	372.99	-
CBD_Mean	352.29	852.67
CBD_Median	332.37	858.76
CBD_k-nn	**326.47**	852.94

4.4 Reproducibility

The preprocessed dataset used in the paper is publicly available unconditionally[2]. Please refer to Eisenstein et al (2010) for the original datasets. A grid search mechanism is selected for searching the best hyperparameters combination. The number of clusters is selected among the range of $K \in \{1,2,3,\ldots,200\}$, while the value of the cost s is selected from $s \in \{0.1,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15\}$ and the value of the nearest neighbor k is selected from $k \in \{1,2,3,4,5\}$.

When each hyperparameter's value is selected, we evaluate the approach on the training set and make prediction on the validation set. The combination of hyperparameters' that achieve the smallest error on the validation set is then used by the approach to evaluate on both training-validation set and to make prediction on the test set. On the validation set, the lowest median Haversine and median Euclidean errors are achieved when the number of clusters is set to $K = 23$ for the K-means clustering, the cost equals $s = 6$ for the SVM and the number of nearest neighbors is set to $k = 1$, while the lowest mean Haversine and mean Euclidean errors are achieved when the number of clusters is set to $K = 16$ for the K-means clustering, the cost equals $s = 11$ for the SVM and the number of nearest neighbors is also set to $k = 1$.

Fig. 2: Hyperparameter search sensitivity analysis with respect to the number of clusters K.



[2] Available online at: http://fs.ismll.de/publicspace/ClusteringBasedDiscretization/

5 Conclusion and Outlook

In this paper, we have introduced an easy-to-implement clustering based discretization approach to tackle the geolocation prediction in Twitter streams. We have chosen a method that does not rely on language models. Instead, at first it manipulates the target space to learn an appropriate clustering. Secondly, it learns a mapping algorithm. And finally, it makes predictions by a distance measurement. The approach is experimentally straightforward and effective. Our empirical study on the CMU dataset and its geolocation task shows that the proposed approach outperforms a number of previous attempts developed in the past literature and achieves current state-of-the-art performance.

For future work, we plan to make further experiments to improve our approach. There are some considerations worth investigating. At first, we would like to conduct an all-in-one model for geolocation prediction in general. It means that instead of combining different separated algorithms, we build a single model that competently captures the users' geolocation. More generally, we wish to apply our approach as well as the further improved model to another geolocation prediction problem: geolocation prediction on tweet level as opposed to user level.

References

Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. The Journal of Machine Learning Research 3:993–1022

Bo H, Baldwin PCT (2012) Geolocation prediction in social media data by finding location indicative words. Proceedings of COLING 2012: Technical Papers pp 1045–1062

Chandra S, Khan L, Muhaya FB (2011) Estimating twitter user location using social interactions–a content based approach. In: Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), pp 838–843, DOI 10.1109/PASSAT/SocialCom.2011.120

Decker BL (1986) World geodetic system 1984. Tech. rep., DTIC Document, URL http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA167570

Eisenstein J, O'Connor B, Smith NA, Xing EP (2010) A latent variable model for geographic lexical variation. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '10, pp 1277–1287

Eisenstein J, Ahmed A, Xing EP (2011) Sparse additive generative models of text. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp 1041–1048

Han B, Cook P, Baldwin T (2014) Text-based twitter user geolocation prediction. J Artif Int Res 49(1):451–500

Hartigan JA, Wong MA (1979) Algorithm AS 136: A k-means clustering algorithm. Applied Statistics 28(1):100–108, DOI 10.2307/2346830

Hong L, Ahmed A, Gurumurthy S, Smola AJ, Tsioutsiouliklis K (2012) Discovering geographical topics in the twitter stream. In: Proceedings of the 21st International Conference on World Wide Web, ACM, New York, NY, USA, WWW '12, pp 769–778, DOI 10.1145/2187836.2187940

Joachims T (1998) Text categorization with Support Vector Machines: Learning with many relevant features, Springer, Berlin, pp 137–142. DOI 10.1007/BFb0026683

Jurgens D (2013) That's what friends are for: Inferring location in online social media platforms based on social relationships. In: Seventh International AAAI Conference on Weblogs and Social Media, URL `http://www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/view/6067`

Kinsella S, Murdock V, O'Hare N (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, ACM, New York, NY, USA, pp 61–68, DOI 10.1145/2065023.2065039

Lampinen R (2001) Universal transverse mercator (utm) and military grid reference system (mgrs). URL `https://www.luomus.fi/en/utm-mgrs-atlas-florae-europaeae`, from the finnish museum of natural history

Peterson LE (2009) K-nearest neighbor. Scholarpedia 4(2):1883, DOI 10.4249/scholarpedia.1883

Robertson S (2004) Understanding inverse document frequency: on theoretical arguments for idf. Journal of documentation 60(5):503–520, DOI 10.1108/00220410410560582

Robusto C (1957) The cosine-haversine formula. American Mathematical Monthly 64(1):38–40, DOI 10.2307/2309088, URL http://www.jstor.org/stable/2309088

Sparck Jones K (1972) A statistical interpretation of term specificity and its application in retrieval. Journal of Socumentation 28(1):11–21, DOI 10.1108/eb026526

Wang X, Grimson E (2008) Spatial latent dirichlet allocation. In: Platt JC, Koller D, Singer Y, Roweis ST (eds) Advances in neural information processing systems, Curran Associates, Inc., pp 1577–1584, URL http://papers.nips.cc/paper/3278-spatial-latent-dirichlet-allocation.pdf

Wing BP, Baldridge J (2011) Simple supervised document geolocation with geodesic grids. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pp 955–964

Zhai C, Lafferty J (2001) Model-based feedback in the language modeling approach to information retrieval. In: Proceedings of the Tenth International Conference on Information and Knowledge Management, ACM, pp 403–410, DOI 10.1145/502585.502654