Text-based Geolocation Prediction of Social Media Users with Neural Networks

Ismini Lourentzou*, Alex Morales*, ChengXiang Zhai
University of Illinois at Ubrana-Champaign
{*lourent2,amorale4,czhai*}*@illinois.edu*

***Abstract*—Inferring the location of a user has been a valuable step for many applications that leverage social media, such as marketing, security monitoring and recommendation systems. Motivated by the recent success of Deep Learning techniques for many other tasks such as computer vision, speech recognition, and natural language processing, we study the application of neural networks to the problem of geolocation prediction and experiment with multiple techniques to improve neural networks for geolocation inference based solely on text. Experimental results on three Twitter datasets suggest that choosing appropriate network architecture, activation function, and performing Batch Normalization, can all increase performance on this task.**

***Keywords*-Geolocation prediction; Text-based Geotagging; Neural Networks; Deep Learning**

I. Introduction

With the growth of social media, many novel applications require inference on user attributes, such as gender, topics of interest and geographical location. For example, most recommender systems on news articles (or products) rely on the user location to target potential readers (or buyers). Geotagging social media posts can become a valuable tool for predicting group behaviors and modeling populations for consumer intelligence; marketing is often guided by collecting and analyzing behavioral data regarding consumer preferences among different locations. Recently social networks have been leveraged to detect terrorism, track virus spread, predict elections, model linguistic differences between groups and inform users regarding natural disasters, as well as coordinate aid and resources in such cases [1]–[3].

Social media services allow users to declare their location by geotagging posts with GPS-based check-ins or by filling a text field in their profile description. However, such text-based descriptions are often missing, unstructured or non-geographical (e.g., "the moon") and thus unreliable and imprecise, with only a tiny proportion of users geotagging their posts [4]. Predicting user location is therefore essential for creating the necessary location annotation and has attracted increasing research interest.

Textual data such as user posts, as well as metadata, i.e. users' time zones or number of followers have been exploited for geolocation prediction [4]–[6]. Moreover, constructing user-friend networks or user-mentions networks has been studied for location inference, with hybrid approaches achieving state-of-the-art results [7]. However, the use of metadata limits the learned model on a specific corpus, since metadata availability depends highly on the provider and can vary among social media platforms (for example, Twitter metadata that are available, such as timezone or number of friends could be unavailable for other corpora, such as Wikipedia). Extracting network information for each user prevents real-time prediction, as it becomes time consuming for large social networks with many edges, for example Facebook, where most users have hundreds of "friends".

On the other hand, text-based supervised approaches can be easily adjusted to new datasets for real-time applications. The difficulty of text-based geotagging is that social media deviate from normal usage of language; emoticons, abbreviations and acronyms (e.g. "LOL"), the lack of conventional orthography (e.g. adding more vowels - "looooooool"), new words and meanings (e.g. "troll") make text-based geotagging a hard and complex task.

From a machine learning perspective, such a complex task would require more effective feature construction than simply combining surface features. In other words, it would potentially benefit from using Deep Learning approaches. Recently, Deep Learning has been demonstrating good performance on various natural language processing (NLP) tasks, such as language modeling, sentiment analysis, POS tagging, named entity recognition and many others. An attractive advantage of these methods is that they perform well without the need of incorporating domain knowledge in the form of time-consuming feature engineering or other external resources. Deep Learning has been revolutionizing both Computer Vision and NLP fields but Information Retrieval and Data Mining communities are only beginning in exploring such methods. Moreover, there has been increasing interest in both industry [8] and academia on applying Deep Neural Networks to social media [9], [10].

Given the success of Deep Learning in so many applications, it is not surprising that it has also been applied to geo-tagging recently [7], [11]. However, the performance of neural networks in related research (e.g., [11]) is not as good as other non-neural network approaches, leaving the question open whether neural networks is a good approach to solving this problem. Since the performance of Deep Learning is known to be sensitive to the architecture design, choice of activation, and other design decisions, in this paper we study how to apply Deep Learning more effectively to solve the

*Contributed equally

problem of text-based geotagging by systematically varying all the major decisions including the activation functions, layer and regularization choices with two different prediction task formulations (i.e., as classification vs. regression) to thoroughly study the impact of each component so as to assess the full potential of basic neural networks for text-based geo-tagging. Our experimental results show that by appropriately configuring the neural network and using Batch Normalization, we can indeed improve the performance substantially over the performance reported in existing work [11], [12], achieving comparable performance with the best method proposed so far for this task. Given that we have only explored optimizing configurations for a basic neural network, it is reasonable to believe that with the application of more advanced neural networks, the performance can be improved further, which would be a very interesting future direction for further exploration.

Specifically, we study the following questions:

1) Related work on the geolocation task improves performance either by building complex pipelines and hierarchies of regression models, or rigorous feature extraction. Can we improve the performance on the social media geolocation prediction with end-to-end architectures that alleviate the need of advanced feature extraction techniques and complex combinations of several components?
2) Batch normalization [13] has recently shown promise in vision. Can it be beneficial for geo-tagging? How effective is regularization, such as Dropout [14], in the social media geolocation task, where most of the datasets are highly imbalanced, as the majority of users live in big cities, such as New York, Los Angeles etc, leaving rural regions underrepresented? In summary, what parameter and regularization choices seem to work better for this task?

This is the first systematic analysis of advanced Deep Learning techniques for geolocation prediction. Through comprehensive experiments[1], we make the following findings:

- The choice of activation can affect performance significantly. (P)ReLUs are more robust to additional components and have stable performance across all tasks.
- Batch normalization is highly effective in stabilizing a neural geolocation model, speeding convergence and increasing robustness across all tasks.
- These two components, alongside with proper weight initialization produce state-of-the-art performance in geolocation prediction, with respect to the optimized metrics during training, i.e. accuracy and mean error distance.
- There is still work remaining on how to optimize additional evaluation components, that have not yet been included in a proper neural network loss function, such as median error distance.

[1]Our code is available at https://github.com/TIMAN-group/geoNN

II. Related Work

Here we focus on the methods that predict user location based solely on text from user posts. Quercini et al. [15] use the contextual evidence, identified by geo-specific gazetteers, in text to geo-tag news articles. Although the premature success, there are several issues in applying this approach to social networks. Taking Twitter as an example, the informal language makes it difficult to construct the gazetteers, let alone user tweets may not be solely about some homogeneous topic, thus complicating the importance of the contextual evidence and finally tweets are very short texts, limiting the information contained. A recent workshop on Twitter geolocation prediction focused on prediction of metropolitan cities, a classification problem [16], using both text and metadata, while our work focuses on prediction based solely on text. Metadata based approaches can improve accuracy but are very specific to the corpus and the types of metadata it makes available [17]. Text-based approaches generalize to all types of data; supplementary information from metadata can be incorporated to any text-based approach.

One of the early works in predicting user location is by Cheng et al. [4] who propose a generative model for city-level geolocation of U.S Twitter users that identifies words in tweets with a strong local geo-scope (location-indicative words). Their method calculates the posterior probability of a user being from a city given his/her tweets. They also experiment with different smoothing techniques.

Eisenstein et al. [5] create a geographic topic model by treating tweets as documents generated by two latent variables, i.e., topic and region. They formulate the problem as both a regression task that predicts geographical coordinates and a classification task, where labels are either the 48 contiguous U.S. states or Washington D.C. or a division between regions (West, Midwest, North-east and South). Due to the computational complexity and efficiency issues of generating topics, the model is limited to small datasets. An important contribution of this work is the creation of the first dataset available for the social media geolocation prediction task, however one has to deal with sparsity issues due to its relatively small size, for example some classes are not represented on the training set.

There have been much subsequent studies using generative models [18], discovering a fixed hierarchical structure over context, via a merging of global topics and regional languages, while Ahmed et al. [19] extends the idea by jointly modeling the location and user context, allowing for an adaptive hierarchical structure for different users.

Wing and Baldridge [20] divide the geographic surface of the Earth into uniform grids and then construct a pseudo-document for each grid. Document similarity based on language models and a nearest neighbors approach is used

for prediction. The granularity of the grids is controlled by a parameter. Uniform grids do not take into account the skewness of the pseudo-document distribution; for example metropolitan areas typically cover most of the twitter posts, while rural areas face the issue of sparsity. Roller et al. [6] address this issue by constructing grids using a k-d adaptive trees, creating more balanced pseudo-documents. They experiment on two datasets of geotagged tweets and one dataset of geotagged English Wikipedia articles. A limitation with this work is that it is unable to discover shared structures in a location, without explicitly controlling the grid sizes. More recently, Wing and Baldridge [17] showed the effectiveness of using logistic regression models on hierarchy of nodes in grids.

Han et al. [21] investigate several feature selection methods for identifying location-indicative words, such as Information gain ratio, geographic density and Ripley's K statistic, as well as the impact of several additional features, such as non-geotagged tweets and metadata on predicting the city of a Twitter user or the actual coordinates.

Cha et al. [22] leverage sparse coding, PCA whitening and dictionary learning for Twitter geolocation to create user representations and then a voting-based grid nearest neighbors approach. Their semi-supervised approach has shown state-of-the-art results for the GeoText dataset [5]. However, the performance increase is due to incorporating word order information, i.e. word sequences, and therefore cannot be applied to the already preprocessed datasets described below.

Liu and Inkpen [11] create the first Deep Neural Network architecture for the geolocation task. A three hidden layer (5000 neurons per layer) Stacked Denoising Auto-encoder, paired with great-circle distance as a loss function and early stopping is tested on two Twitter datasets. However, little insight was given on how the choice of different components affects performance, for example the activation function, the number of layers, pre-training or parameter tuning.

Rahimi at el. [12] use Mixture Density Networks [23] for Twitter user geolocation and lexical dialectology. They cast the problem in a classification task and showed that it outperforms regression models by a large margin. By sharing the parameters of the Gaussian mixtures they achieve competitive results with state-of-the-art classification models.

Our work advances the Deep Learning approach and shows that carefully designed architectures can achieve better performance than complex models. We provide the first comparison on how the choice of activation functions, number of neurons per layer, initialization and regularization affects performance on predicting the actual geographical user coordinates, as well as classifying users per state or region. As most applications of Deep Learning advance by improving known models, this work serves as a good starting point that could benefit DNN practitioners and researchers to identify areas for further improvements in "neural geotagging".

Datasets			
Dataset Name	Users	Sample Size	Region
GeoText	9.5K	380K tweets	Contiguous US
TwUS	450K	38M tweets	North America
TwWORLD	1.4M	12M tweets	English World Wide

Table I: Datasets summary

III. Datasets

We compare our performance with previous text-based systems built on three publicly available datasets from Twitter. To the best of our knowledge, these are the data sets used in previous related work on the twitter text-based geolocation task, while their differences in terms of number of users make them appropriate for comparison of architectures with respect to the availability (or lack) of social media data. See Table I for a summary.

- GeoText is a dataset from Eisenstein et al. [5] that contains 380,000 tweets from 9,500 users with geographical coordinates for each user. All users come from the contiguous United States (i.e., the U.S. excluding Hawaii, Alaska and all off-shore territories).
- TWUS is a dataset of 38M tweets from 450K users located in North America compiled by Roller et al. [6]. Each training example is the collection of all tweets by a single user, where the earliest geotagged tweet determines the user's location. The dataset is already split in training, development and test sets, where 10,000 users are reserved for the development and test sets.
- TWWORLD is a dataset of tweets from 1.4M users (with 10,000 reserved for the development and test sets) compiled by Han et al. [21]. While TWUS is limited to the United States, this dataset covers the entire Earth. Non-English tweets and those not near a city were removed, in addition to filtering non-alphabetic, overly short and overly infrequent words.

IV. Task Definition

A. Models

To evaluate the sensitivity of our models to different tasks and compare our work with the previous literature, we predict the location of a Twitter user, either the exact coordinates (regression), U.S. state or region (classification), based only on the user posts. We first describe the input and output of our models:

B. Input Features

Wing and Baldridge [17] have already pre-processed and released the aforementioned datasets in the format

$$[M]\ [term_1][count]\ldots[term_N][count]$$

where $[M]$ is a Twitter user's id and the $[count]$ associated with each term is how many times that term appeared in the user's posts.

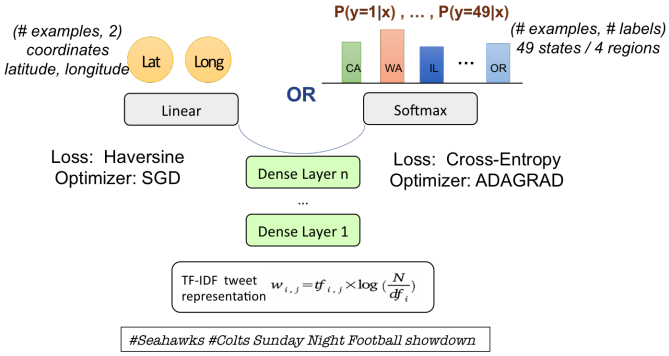


Figure 1: Our neural models for Geolocation

We therefore use a bag-of-words text representation and extract TF-IDF features, considering only the 50000 most frequent unigrams. We have also tried other textual representations, such as frequency counts, binary (presence or absence of a word) and averaged word embeddings, however our preliminary experiments indicated that TF-IDF produces the best results.

C. Model Output/Prediction tasks

Most of the previous literature only predicts the coordinates of the user location. Following Liu et al. [11], we provide results for the classification task, where each user is classified into a geographical region, either the 48 contiguous U.S. states or Washington D.C. (49 classes) or four classes, which represent the main four U.S. regions as defined by the Census Bureau[2]: the West, Midwest, South, and Northeast. We also report performance on the regression task, i.e. predicting the actual geographical coordinates.

All of our models therefore have the same output layers with respect to the task at hand:

- Linear layer with 2 neurons for the regression task, i.e. latitude and longitude coordinates
- Softmax layer with 49 or 4 neurons for the state and region classification tasks, respectively.

V. Architectures

We developed a three hidden layer network and vary all other several components. Below we present some of the parameter choices and discuss how these might affect performance. Figure 1 illustrates our proposed models[3].

A. Loss Functions

The objective function of our models depends on the task. We use the same loss functions as [11]. More specifically, for the classification tasks we used categorical cross entropy. When the output layer activation is the softmax function, categorical cross entropy can also be interpreted as the negative log likelihood or the KL-divergence between the output distribution and the target distribution, and is a typical loss function used in the Deep Learning literature.

For the prediction task, since the models produce location latitude and longitude coordinates, we can define the objective function as the great-circle distance between the estimated and actual coordinates, which can be calculated by the haversine formula:

$$d = 2r \arcsin\left(\sqrt{\alpha}\right) \tag{1}$$

$$\alpha = \sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right) \tag{2}$$

where

- r is the Earth radius
- φ_1,φ_2 and λ_1,λ_2 are the latitude and longitude of the predicted and true coordinates
- d is the final calculation of the distance, and consequently our error loss function.

B. Design choices: activation functions, weight initialization, regularization methods

Our selection of activation functions consists of one non-linear (sigmoid) and one linear non-parametric function (ReLU), as well as a parameterized linearity (PReLU). He at al. [26] propose a ReLU adjusted version of the Xavier weight initialization [27]. More specifically, initializing the weights of each neuron by drawing them from a distribution with zero mean and variance $Var(W) = \frac{2}{n_{in}}$ where W is the initialization distribution for the neuron (usually Gaussian or uniform) and n_{in} is the number of neurons from the previous layer that are passing a signal to this neuron, offers better guarantees in terms of gradient-based weight updates. Moreover, recent work [13] has shown that ensuring a stable distribution of non-linearity inputs during training could prevent the optimizer from getting stuck in a saturated regime, and the training would accelerate as the use of higher learning rates would not be an issue. Adding a linear layer before activation functions to perform batch-wise normalization (called "Batch Normalization") also solves the problem of ReLUs, in addition to regularizing the model and reducing the need for other regularization techniques, such as Dropout. To our best knowledge, *this is the first work that applies Batch Normalization to social media posts*. Dropout is a very simple regularization technique [14] that prevents over-fitting and speeds up training by randomly disabling neurons with a probability p (common choices are $p = 0.25$ or $p = 0.5$) in the learning phase. This prevent weights from converging to identical positions, as for each training example a different set of neurons is randomly "dropped", which results in robust feature representations that can generalize better to new data. We experiment with both regularization techniques; we will discuss our results later on.

[2] https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf

[3] Our models are developed with Keras [24] and Theano [25] as backend.

C. Hyper-parameter choices

We tune the rest of the hyper-parameters with Tree-structured Parzen Estimator (TPE), a Bayesian sequential model-based optimization approach described in Bergstra et al. [28]. Learning rates we explore are: {0.1, 0.01, 0.001, 0.0001}. The maximum number of epochs is set to 1000, however we terminate training when the validation accuracy stops improving after a number of steps (in our case 20), known as Early Stopping [29]. This technique prevents over-fitting, with most of our configurations to finish training in approximately 100 epochs. Batch size was set to 64 for GeoText and 512 for TWUS and TWWOLRD.

Finally, we experiment with several optimizers and our final choices are ADAGRAD [30] for the classification task, with 10^{-8} learning rate decay over each update, which dynamically adapts the learning rate to the data and Stochastic Gradient Descent (SGD) for the regression tasks, with 10^{-6} learning rate decay over each update and 0.9 Nesterov momentum [31]–[33]. In our experiments we discovered that adaptive gradient-based optimization algorithms decreased the performance for the regression tasks, which is mainly due to our choice of cost function: our error range is of thousands of kilometers, and it seems that performing updates based on the slope of the error function as SGD does works better than adjusting the updates based on the feature frequencies.

VI. Experimental Results

A. Performance Metrics

We use the same performance metrics as previous work in text-based geolocation inference. For the classification tasks, we compare performance in terms of accuracy, which is defined as the proportion of users in the test set that are correctly classified. For the regression task, the performance metrics were introduced by Cheng et al. [4]: the mean error distance and the median error distance (in kilometers) between the predicted and the actual location, as well as accuracy within a 161km radius.

We present our results and compare with the related existing work that performs the same tasks, i.e. classification w.r.t to the U.S. state or region and prediction of the coordinates.

B. Discussion of the Results

For the GeoText dataset, Liu and Inkpen [11] provide the labels for the classification task. Using reverse geotagging, the authors were able to retrieve the city, state and country for each example in the dataset. We utilize these labels to perform the classification task. For the rest of the datasets, we follow previous literature and report the latitude and longitude predictions.

GeoText States		Dev(Test) Acc %		
#neurons per layer	Activaction Function	No Regularization	BatchNorm	BatchNorm + Drop(0.5)
128	PRELU	39.0 (38.8)	40.7 (39.4)	41.3 (41.7)
512		40.1 (39.7)	42.3 (42.8)	42.9 (43.3)
1024		37.5 (37.5)	43.2 (44.2)	43.7 (43.2)
4096		26.5 (26.2)	43.7 (43.0)	44.3 (44.3)
128	RELU	38.8 (37.3)	40.7 (40.4)	33.7 (33.4)
512		37.9 (37.7)	43.2 (42.6)	42.1 (42.9)
1024		32.9 (34.7)	43.3 (43.3)	43.9 (43.2)
4096		26.5 (26.2)	44.1 (43.2)	44.0 (44.4)
128	Sigmoid	32.8 (31.9)	40.9 (42.5)	41.0 (40.9)
512		32.5 (32.6)	42.0 (41.7)	43.2 (42.5)
1024		30.1 (30.5)	42.3 (42.6)	43.5 (43.3)
4096		4.2 (4.6)	40.8 (41.2)	42.8 (42.9)

Table II: Batch Normalization and Dropout effect on GeoText held-out development (test) set; U.S. states classification task

1) Classification tasks: We first note here that the two classification tasks have different levels of difficulty: in the states classification, the dataset is divided into a very skewed distribution that favors states with metropolitan areas, such as New York, whereas rural areas are underrepresented. This phenomenon and the number of classes in total (49) increases the complexity of the task. Regions classification can be considered easier than the states classification and as we will see it does not require complex models compared to the states classification, which is more prone to over-fitting.

Table II presents a comparison of activation functions across different hidden layer sizes, and how Batch Normalization and Dropout affects the performance in the states classification task. Without Batch Normalization, we see that the ReLU activation function works better than other options. In general, Batch Normalization seems highly valuable in improving performance, irrespective of the activation function choice. The sigmoid activation function has by far the highest performance increase and is producing the best results. However, performance differences among activation functions are diminished; the highest development accuracy with Batch Normalization is 44.00% with ReLU activation functions. We also report results for varying Dropout across different hidden layer sizes. In all cases, Dropout's affect on the performance is marginal. The best architecture is PReLUs with 4096 neurons and all regularization methods, with 44.3% accuracy in the development set.

In Table III we present the same comparison for the regions classification task. Batch Normalization helps in regularizing wider network architectures, but it's effect is overall limited. Dropout also adds a small improvement. The best accuracy is now produced with ReLUs at 67.4% and 68.5% without or with Dropout, respectively. Surprisingly, all of our experiments do not exceed the 67% accuracy level on the development set.

GeoText Regions		Dev(Test) Acc %		
#neurons per layer	Activaction Function	No Regularization	BatchNorm	BatchNorm + Drop(0.5)
128	PReLU	68.1 (66.0)	66.1 (63.9)	66.5 (64.8)
512		66.6 (64.5)	67.2 (65.9)	68.3 (66.5)
1024		66.5 (65.1)	66.1 (65.5)	67.2 (66.3)
4096		38.3 (37.4)	65.8 (64.1)	68.5 (66.1)
128	RELU	67.8 (66.6)	66.4 (64.2)	67.5 (65.9)
512		66.6 (65.9)	67.4 (66.3)	68.5 (66.7)
1024		66.8 (65.8)	66.1 (65.5)	67.0 (67.3)
4096		38.3 (37.4)	65.9 (63.5)	68.4 (66.6)
128	Sigmoid	63.8 (63.5)	65.8 (64.0)	67.0 (65.9)
512		66.6 (65.1)	64.7 (63.8)	67.3 (66.8)
1024		36.4 (35.9)	64.9 (63.3)	65.7 (65.7)
4096		38.3 (37.4)	63.7 (61.3)	66.8 (64.9)

Table III: Batch Normalization and Dropout effect on GeoText held-out development (test) set; U.S. regions classification task

2) Regression task: For the regression task (Fig. 2), we see again that Batch Normalization is improving all configuration settings. We check whether the combination of both regularization methods would further improve results; Dropout has limited effect on the performance. In our experiments, we found that the best architectures are shallow and wide networks without Dropout, with PReLUs and Sigmoid activations for TWUS and TWWORLD, respectively. For GeoText, a denser ReLU network gives the best performance.

Overall, Dropout is extremely sensitive to the complexity of the task, while Batch Normalization is a robust technique that improves performance and speeds convergence. We are certain that Batch Normalization could continue to be one of the main components in improving the neural models for geolocation prediction. Interestingly, this method has also achieved better than human-level performance on the ImageNet visual recognition challenge.

3) Number of Layers vs. Number of Neurons: We also varied the number of neurons per layer as well as the number of hidden layers for all tasks. In Figure 3, we present the hidden layer size variation for the GeoText states classification task as well as the regression task in GeoText and TWUS, on our best performing models.

We can see that the optimal architecture with respect to the hidden layer size is task and data dependent, where "no solution fits all". For the classification task, there is a slight increase of performance as the number of neurons per layer increases. For the GeoText regression task, the performance is inversely proportional to the number of neurons per layer, however both TWUS and TWWORLD present the exact opposite case; performance increases by adding more neurons. GeoText is the smallest and most imbalanced dataset in our evaluation. Given that the availability of social media data has been increased in the past years, we argue that Deep Learning can further improve geolocation prediction; we leave the discovery of new architectures to future work.

	dropout	hidden	activation	layers
GEOTEXT states	0.5	4096	PReLU	3
GEOTEXT regions	0.5	512	ReLU	3
GEOTEXT regression	0.5	128	ReLU	3
TWUS	0	4096	PReLU	5
TWWORLD	0	4096	Sigmoid	3

Table IV: Best performing hyper-parameter settings of our proposed geolocation prediction models

Moreover, we varied the number of hidden layers for our best architetures. The same pattern appeared in both tasks, as shown in Figure 4, suggesting compact architectures for the regression task and shallow and wide architectures for classification tasks.

Our final hyper-parameter choices are summarized in Table IV.

C. Comparison with Related Work

With respect to previous related work on text-based geolocation, we present results from our best architectures (models chosen based on the performance on the appropriate development data sets and trained on full datasets per epoch). We achieve state of the art in the classification tasks (table V). We should note that for the GeoText regression task (table VI), Cha et al. [22] improve performance on GeoText by leveraging word sequences. As the rest of the prior work operates on TFIDF input representation, we choose the same for fair comparison, and report both results of Cha et al. on the regression task, i.e. using word counts and word sequences. We leave the addition of temporal information to future work. For TWUS and TWWORLD, our results on tables VII and VIII also show comparable performance with related work, with lower mean in all cases, including the recently proposed neural models [7], [11]. Since the mean error is the objective function used in our experiments, this is a good indication that Deep Neural Networks are well suited for geolocation prediction, leaving room for future improvement.

VII. Error Analysis

To further understand what types of mistakes the model makes we a used geo-coordinate visualization, Carto[4]. In Figure 5a we show the clusters of the ground truth data, i.e. the correct latitude and longitude, here we choose seven clusters to correspond to different regions of the United

[4]https://carto.com/

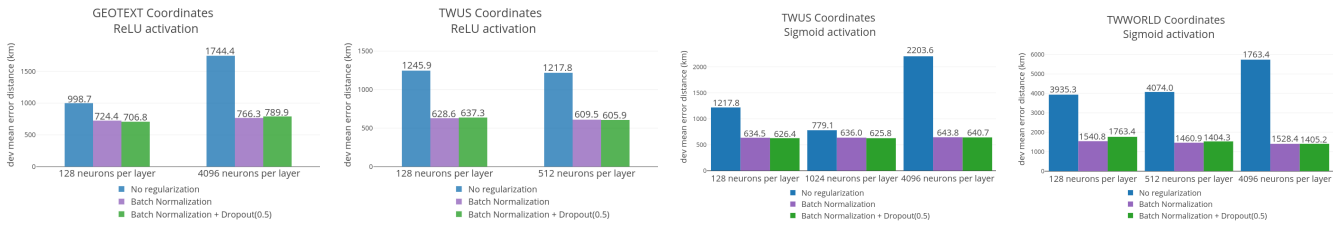


Figure 2: Batch Normalization and Dropout effect - regression tasks (coordinates prediction)

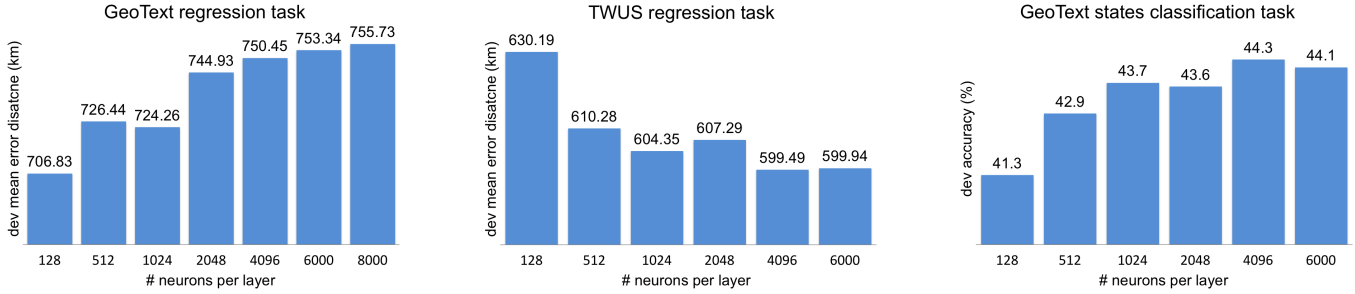


Figure 3: Varying number of neurons per layer

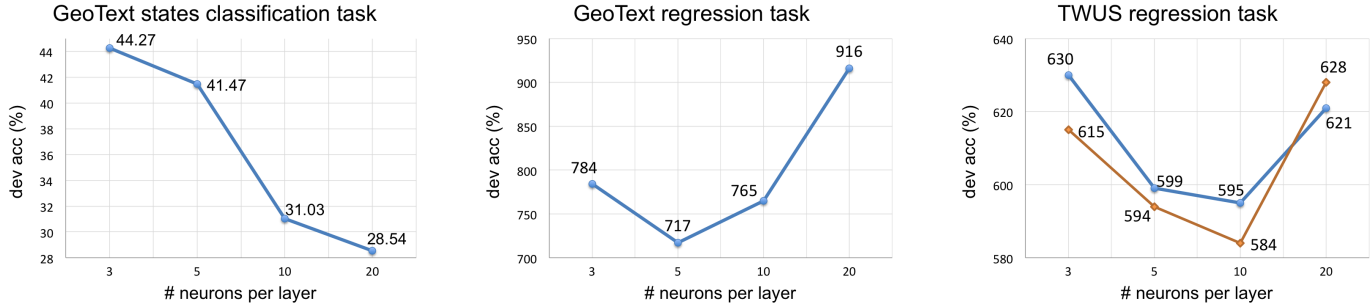


Figure 4: Varying number of layers

GEOTEXT	Accuracy (%)	
Models	States (49-way)	Regions (4-way)
Proposed method	44.3	67.3
Liu and Inkpen, 2015 (SDA)	34.8	61.1
Eisenstein et al., 2010 (Geo topic model)	24	58
Cha et al., 2015 (SC+all - including word sequences)	41	67

Table V: Performance comparison on GeoText held-out test set - states and regions classification tasks

States; each cluster here is represented by a different color. Using those clusters we show our predictions in Figure 5b. The visualization is also available online[5], and it also contains widgets for filtering by distance errors.

One observation we make from our predictions, is that in some cases we predict locations in regions consisting of water, using the spacial and other network information we could potentially further calibrate such predictions.

In the smallest dataset GeoText, we found several issues concerning imbalance of the data. For example, in the

[5]https://amorale4.carto.com/builder/a77e1130-e1dd-11e6-9d31-0e98b61680bf

GEOTEXT	Geolocation Error (km)		
Models	Mean	Median	Acc@161
Proposed method	747	448	29
Rahimi et al.,2017 (MDN-SHARED)	865	412	39
Liu and Inkpen, 2015 (SDA)	856	-	-
Cha et al., 2015 (SC+all - word counts)	926	497	-
Cha et al., 2015 (SC+all - including word sequences)	581	425	-
Roller et al.,2012 (UnifKdCentroid)	890	473	34
Roller et al., 2012 (KdCentroid)	958	549	35
Roller et al., 2012 (UnifCentroid)	897	432	36
Wing and Baldridge, 2011 (KL)	967	479	-
Eisenstein et al.,2011 (SAGE)	845	501	-
Eisenstein et al., 2010 (Geo topic model)	900	494	-

Table VI: Performance comparison on GeoText held-out test set - regression task (coordinates prediction)

TWUS	Geolocation Error (km)		
Models	Mean	Median	Acc@161
Proposed method	570	223	43
Rahimi et al.,2017 (MDN-SHARED)	655	216	42
Liu and Inkpen, 2015 (SDA)	733	377	24
Wing and Baldridge, 2014 (HierLR Uniform)	704	171	49
Wing and Baldridge, 2014 (HierLR k-d)	687	191	48
Han et al., 2014 (IGR)	-	260	45
Han et al., 2014 (LR)	-	878	23
Roller et al.,2012 (UnifKdCentroid)	913	532	33
Roller et al., 2012 (KdCentroid)	860	463	35
Roller et al., 2012 (UnifCentroid)	956	570	31

Table VII: Performance comparison on TwUS held-out test set (coordinates prediction)

TWWORLD	Geolocation Error (km)		
Models	Mean	Median	Acc@161
Proposed method	1338	495	21
Wing and Baldridge (2014) & HierLR Uniform	1715	490	33
Wing and Baldridge (2014) & HierLR k-d	1670	509	31
Han et al. (2014) & IGR	-	913	26
Han et al. (2014) & LR	-	640	23

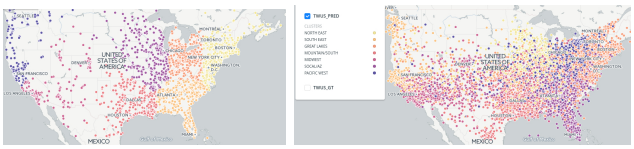
Table VIII: Performance comparison on TWWORLD held-out test set (coordinates prediction)

classification tasks there are were some states which were not at all represented in the training or development set but appeared in the test set. Moreover, for the regression task we found one point that was in fact in Europe. We did not remove these points to make our results comparable to the other approaches, however future research should be weary of using this dataset.

In Figure 6a we show the imbalance of the Geotext. For reference, the top four labels with the majority of labels correspond to New York (2), California (3), Georgia (20), and Florida (8) with 486, 195, 121, and 100 number of testing examples respectively. The normalized confusion matrix in Figure 6b, shows that most of the categories with low number of examples are miss-classified as one of the majority class. Despite such discovery of limitations, our experiments on three varying-sized datasets, GeoText, TWUS and TWWORLD, validate that Deep Learning techniques can be utilized for improving performance on the geolocation task.

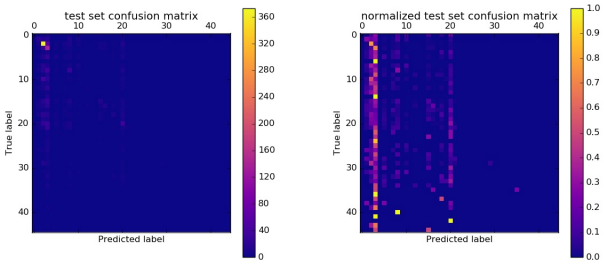
New York	Dallas	Los Angeles	San Francisco	Miami
cashmere	sundance	fitzpatrick	engineers	clemson
authenticity	bachmann	2pac	bot	preseason
trousers	immigrants	guste	gadgets	lansdowne
chadwick	administration	morningside	workflow	brewery
pearls	socialist	alvaro	ristorante	ginza
lakeshore	opposition	footlocker	execute	pike
afterparty	earthquake	dreads	sashimi	obsolete
fishbowl	occupywallstreet	cuffed	geniuses	jameson
wahlberg	brutality	afterhours	dinero	thunderstorm
mcqueen	bankrupt	calvary	unfriend	ethnic

Table IX: Selected words with smallest average median distance errors in selected areas (TWUS)



(a) Ground Truth, clusters (b) Our model predictions

Figure 5: TWUS dataset clusters and visualization



(a) Raw counts (b) Normalized counts

Figure 6: Confusion matrices for the Geotext data.

In Table IX, we show the words with the smallest average distance errors for different cities. Our model is able to distinguish several location-indicative words. It finds many restaurants local to a particular city, for example ‘Chadwick’ is a popular Brooklyn restaurant, ‘Ristorante Milano’ is a restaurant in San Francisco and ‘Ginza’ is the name of a highly rated Japanese Buffet in Miami. The model is also able to distinguish vernacular in twitter for those locations. In New York City there are frequent mentions of clothing brands and their quality, which makes sense since it is often described as the fashion capital of the U.S.; in San Francisco technology terms are also expected since it is located close to Silicon Valley; Miami has many terms associated to sports, for example ‘Lansdowne’ is a popular sports theme bar and ‘Clemson University’ corresponds to the sport rivals of ‘Miami University’.

It is surprising to see that Los Angeles and Dallas are very different in terms of language usage, having no clear location-specific topics. On the other hand, Miami describes the populous movement, “occupy wall street” as well as the problems associated with, such as police brutality and banking bailouts, while in Los Angeles the terms are more related to physical appearance. These terms show that tweets have some dynamic property to them and thus we could incorporate methods that utilize temporal aspects, such as event discovery, to learn better location-indicative terms.

VIII. Conclusion and Future Work

We experiment with neural network architectures for predicting the location of social media users. We explore which parameters affect the evaluation metrics and how our

careful choices can increase the performance. Experimental results show how each of these hyper-parameter changes impacts our models, and which modules increase the model performance, for example the initialization of the weights leads to better convergence, Batch Normalization leads to better regularization. We show that Batch Normalization has the highest performance increase and that Dropout seems to have an overall mixed effect with minor improvements.

Our models produce results that either advance the state-of-the-art or leave room for systematic improvements. Beyond highlighting the key components that improve robustness and the limitations of the smaller datasets, we move on to provide an error analysis and linguistic analysis of our models, while our prediction errors can be further analyzed through a visualization made available online. Our analysis provides valuable understanding on how to search for the optimal architecture, taking into account the task setup. This can be particularly useful in the case of transfer learning; for example when the classification task is refined, what options are available for keeping the performance at the same level.

Furthermore, additional error analysis can provide more guidance to building on this work, which we hope to do in the future. While our focus in this work was mainly a general text-based neural model for geolocation prediction, it would also be worth to investigate the effect of additional information beyond text, such as metadata or user information, as neural networks are also well suited for incorporating such features.

Our exploration is by no means exhaustive. A more interpretable variation should be able to better capture linguistic similarities among Twitter users and jointly learn "user embeddings" alongside with word embeddings. There is additional potential for further improvement by exploring unlabeled social media data with unsupervised techniques, such as pre-training with autoencoders or adding social network and word order information with architectures, such as siamese [34], convolutional or recurrent networks [35]–[37] and network embeddings [38] that would facilitate such user/word representations and might advance the (neural) geolocation task.

REFERENCES

[1] K. Cohen, F. Johansson, L. Kaati, and J. C. Mork, "Detecting linguistic markers for radical violence in social media," *Terrorism and Political Violence*, vol. 26, no. 1, pp. 246–256, 2014.

[2] M. E. Ireland, H. A. Schwartz, Q. Chen, L. H. Ungar, and D. Albarracín, "Future-oriented tweets predict lower county-level hiv prevalence in the united states." *Health Psychology*, vol. 34, no. S, p. 1252, 2015.

[3] E. M. Cody, A. J. Reagan, P. S. Dodds, and C. M. Danforth, "Public opinion polling with twitter," *arXiv preprint arXiv:1608.02024*, 2016.

[4] Z. Cheng, J. Caverlee, and K. Lee, "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*. ACM, 2010, pp. 759–768.

[5] J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing, "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, 2010, pp. 1277–1287.

[6] S. Roller, M. Speriosu, S. Rallapalli, B. Wing, and J. Baldridge, "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*. Association for Computational Linguistics, 2012, pp. 1500–1510.

[7] A. Rahimi, D. Vu, T. Cohn, and T. Baldwin, "Exploiting text and network context for geolocation of social media users," *arXiv preprint arXiv:1506.04803*, 2015.

[8] FacebookEngineeringBlog, "Introducing deeptext: Facebook's text understanding engine," https://code.facebook.com/posts/181565595577955, 2016, accessed: 2016-10-02.

[9] S. Vosoughi, P. Vijayaraghavan, and D. Roy, "Tweet2vec: Learning tweet embeddings using character-level CNN-LSTM encoder-decoder," *CoRR*, vol. abs/1607.07514, 2016. [Online]. Available: http://arxiv.org/abs/1607.07514

[10] M. Korpusik, S. Sakaki, F. Chen, and Y. Chen, "Recurrent neural networks for customer purchase prediction on twitter," in *Proceedings of the 3rd Workshop on New Trends in Content-Based Recommender Systems co-located with ACM Conference on Recommender Systems (RecSys 2016), Boston, MA, USA, September 16, 2016.*, 2016, pp. 47–50. [Online]. Available: http://ceur-ws.org/Vol-1673/paper9.pdf

[11] J. Liu and D. Inkpen, "Estimating user location in social media with stacked denoising auto-encoders," in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, NAACL*, 2015, pp. 201–210.

[12] A. Rahimi, T. Baldwin, and T. Cohn, "Continuous representation of location for geolocation and lexical dialectology using mixture density networks," in *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP2017)*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017. [Online]. Available: http://people.eng.unimelb.edu.au/tcohn/papers/emnlp17geomdn.pdf

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[15] G. Quercini, H. Samet, J. Sankaranarayanan, and M. D. Lieberman, "Determining the spatial reader scopes of news sources using local lexicons," in *proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*. ACM, 2010, pp. 43–52.

[16] B. Han, A. Hugo, A. Rahimi, L. Derczynski, and T. Baldwin, "Twitter geolocation prediction shared task of the 2016 workshop on noisy user-generated text," *WNUT 2016*, p. 213, 2016.

[17] B. Wing and J. Baldridge, "Hierarchical discriminative classification for text-based geolocation." in *EMNLP*, 2014, pp. 336–348.

[18] J. Eisenstein, A. Ahmed, and E. P. Xing, "Sparse additive generative models of text." in *ICML*, L. Getoor and T. Scheffer, Eds. Omnipress, 2011, pp. 1041–1048.

[19] A. Ahmed, L. Hong, and A. J. Smola, "Hierarchical geographical modeling of user locations from social media posts," in *Proceedings of the 22nd international conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2013, pp. 25–36.

[20] B. P. Wing and J. Baldridge, "Simple supervised document geolocation with geodesic grids," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 955–964.

[21] B. Han, P. Cook, and T. Baldwin, "Text-based twitter user geolocation prediction," *Journal of Artificial Intelligence Research*, pp. 451–500, 2014.

[22] M. Cha, Y. Gwon, and H. Kung, "Twitter geolocation and regional classification via sparse coding," in *Proceedings of the 9th International Conference on Weblogs and Social Media (ICWSM 2015)*, 2015, pp. 582–585.

[23] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995.

[24] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

[25] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[27] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[28] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in Neural Information Processing Systems*, 2011, pp. 2546–2554.

[29] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 437–478.

[30] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[31] Y. Nesterov, "A method of solving a convex programming problem with convergence rate o (1/k2)," in *Soviet Mathematics Doklady*, vol. 27, no. 2, 1983, pp. 372–376.

[32] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th international conference on machine learning (ICML-13)*, 2013, pp. 1139–1147.

[33] J. Moody, S. Hanson, A. Krogh, and J. A. Hertz, "A simple weight decay can improve generalization," *Advances in neural information processing systems*, vol. 4, pp. 950–957, 1995.

[34] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 539–546.

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[36] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[37] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[38] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding." in *WWW*. ACM, 2015.