Location Prediction using Language Variation

Meenakshi Sundaram Viswanathan, Gogula Krishnan Saravanan, Muthiah Nachiappan, Preetha Udhayakumar, Vipul Sharma

Abstract

Our project investigates the feasibility of geographically locating Twitter users based solely on tweet content. We are trying to locate a user using their tweet content by understanding the dialect differences across geographies through deep learning techniques. We are not using any other external information to locate the user. This project provides an approach to augment existing systems that locate users.

1 Introduction

This project explores the workability of multiple deep learning models in predicting the user's location based on their tweet content. We intend to classify the tweets into different geographical regions using the dialect differences. Input will be the tweet content. And the output will be the location of that tweet's user. For example, the system will predict "California" or "Texas" based on the type of dialect expressed in the user's tweet.

Currently, fewer than 3% of current tweets are configured to include geographical information (Liu et al., 2015), and this project provides an approach to elevate existing user locating systems. This project is trying to predict the user's location without the use of external information such as user-specified “hometown” data, location names from Named Entity Recognizer, or Twitter social graph data. The ability to locate Twitter users based on tweet content has many practical applications including improving our understanding of language and dialect differences across geographies, detecting anomalies to track down travelers, and predominantly to segment users by geography for marketing or emergency purposes.

2 Related Work

Many prior studies have been conducted over the last few years focusing on locating Twitter users based on dialect patterns. Most experiments relied heavily on statistical Natural Language Processing (NLP) techniques. (Eisenstein et al., 2010) conducted the first significant study attempting to locate Twitter users predictively. (Eisenstein et al., 2010) tried multiple approaches like topical modeling, k-nearest neighbors, and several statistical methods including LDA and regression. (Liu et al., 2015) expanded upon Eisenstein's previous work through Stacked Denoising Autoencoder (SDA) feed-forward neural network. (Eisenstein et al., 2010) (Liu et al., 2015) predicted the user's location using tweet as a 4-way regional and 48-way state level classification task. Table 1 provides the result of previous works.

Model		**Accuracy (%)**
Eisenstein (2010)	Geo topic	58.0
	Unigram	53.0
	LDA	39.0
	Regression	41.0
	kNN	37.0
Liu (2015)	SDA-1	61.1

Table 1: Results of prior works for the 4-way regional classification task

3 Dataset

In order to compare the performance of our model with prior works, we choose a publicly available GeoText[1] dataset from (Eisenstein et al., 2010). Several other researchers have used this dataset.

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

The GeoText dataset contains the latitude and longitude value of the place where a user created the account, the tweet text of all the tweets by that users. It includes about 380,000 tweets from 9,500 users from the contiguous United States (i.e., the U.S. excluding Hawaii, Alaska and all off-shore territories) collected during a seven day period in March 2010. Average tweet length is 14 words. Again to make the comparisons fair, we split the GeoText dataset in the same way as (Eisenstein et al., 2010), i.e., 60% for training, 20% for validation and 20% for testing. Table 2 provides the distribution of tweets across different regions. Metadata like user's profile and time zone will not be used.

Region	**Count**	**Percent(%)**
Midwest	43,170	11.71
Northeast	139,132	37.73
South	140,291	38.05
West	46,157	12.52
	368,750	100.00

Table 2: Number of tweets across each region in GeoText

4 Data Preprocessing

A preprocessing pipeline is constructed to prepare and standardize the tweet data. Figure 1 shows the overall flow of preprocessing pipeline. Initial cleaning will make the raw dataset more suitable for our task. Geocoding and filtering are two significant steps in initial cleaning. We are using the reverse geocoding technique[2] to convert latitude, longitude values to a particular region in the United States. After geocoding, we will have a few anomaly data that will have regions outside the United States. Filtering is done to remove these anomalies.

Since we are experimenting with multiple deep learning models, model specific preprocessing is performed after the initial preprocessing. Model specific preprocessing includes tweet cleaning, tokenization, and word vectorization. Tweet cleaning is carried out similar to GloVe's preprocess twitter approach[3]. NLTK's Twitter-aware tokenizer[4] (Liu et al., 2015) or Keras's standard tokenizer[5] is used to tokenize the tweets based on the word vectorizer. The baseline multilayer perceptron (MLP) model uses TfidfVectorizer whereas GloVe[6] embedding based on 27B tweets is used to vectorize the words in recurrent and convolutional models. GLoVe vectors of length 25, 50, 100, 200 are leveraged in this project. Glove vector of length 100 seems to perform better in our case.

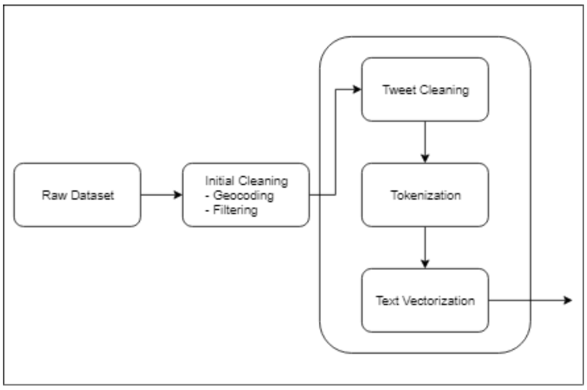

Figure 1: Data Preprocessing Pipeline

5 Model

Based on (Liu et al., 2015) we define our work as a classification task where each tweet is classified into a region. All our models are trained on the training dataset, with each tweet and its corresponding region label is considered as a training example. For a baseline model, we carried out MLP very similar to (Liu et al., 2015)'s SDA-1. Our final MLP model after hyperparameter tuning (Table 3) has three dense layers with 32, 16, 8 number of units in each layer respectively. Gaussian noise is added after each dense layer to avoid overfitting. And the output layer is a softmax layer.

We then decided to determine the effectiveness of recurrent neural network (RNN) architectures like long short-term memory (LSTM), gated recurrent unit (GRU). RNN architectures have proven to capture better features for NLP problems (Yin et al., 2017), and for this classification task, significant prior works didn't experiment with RNN architectures.

[2] Reverse geocoded using services provided by http://www.mapquest.com

[3] https://nlp.stanford.edu/projects/glove/preprocess-twitter.rb

[4] https://www.nltk.org/api/nltk.tokenize.html

[5] https://keras.io/preprocessing/text/

[6] https://nlp.stanford.edu/projects/glove/

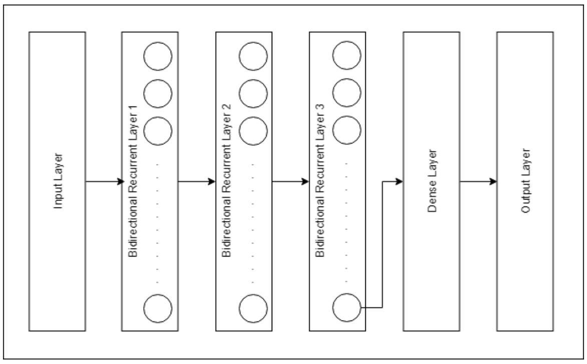
Input Layer
Bidirectional Recurrent Layer 1
Bidirectional Recurrent Layer 2
Bidirectional Recurrent Layer 3
Dense Layer
Output Layer

Figure 2: Illustration of our best GRU/LSTM model

Our GRU and LSTM recurrent models have two major parts. Initially, we have a number of bidirectional recurrent layers to capture the implicit features. Following that, we have dense layers to do the classification task. We performed hyperparameter tuning using scikit-learn's RandomizedSearchCV[7]. We tuned across several hyperparameters like number of recurrent layers, number of dense layers, number of units in each layer, etc. Refer to Table 3 for more information related to hyperparameter tuning. Figure 1 shows the illustration of our best LSTM and GRU model. Our best GRU model has three bidirectional recurrent layers and one dense layer. The number of units for each recurrent layer and dense layer is 10. Gaussian noise is added after the dense layer to avoid overfitting. Our best LSTM model is very similar to our GRU model except for the fact that it has dropout for each recurrent layer with a dropout rate of 0.2.

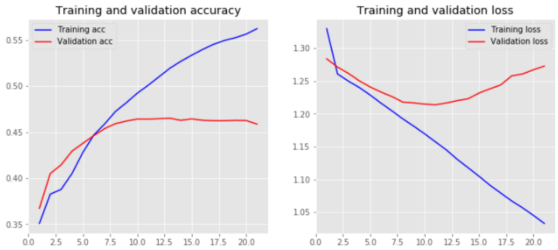

Figure 3: Training history of LSTM model

Given that tweets contain a maximum of up to 140 characters with an average tweet length being 14 in our dataset, we had a reason to try out convolutional neural network (CNN). Also, CNN performs better in general for NLP classification task as it captures interesting n-gram features (Yin et al., 2017). After the hyperparameter optimization (Table 3) step, our best CNN model has five layers of convolution each consisting of 64 kernels with filter size 5. Convolution layers have a decaying dropout with an initial dropout rate of 0.3. Following that is a dense layer with ten units and a softmax output layer.

Model	**Hyperparameter**	**Values**
MLP	No. of hidden layers	1, 2, 3, 4, 5
	No. of units	16, 32, 64, 128
	Gaussian Noise (SD)	0, 0.1, 0.15,0.2
	Dropout (rate)	0, 0.1, 0.2
GRU/LSTM	No. of recurrent layers	1, 2, 3
	No. of recurrent units	10,15,20,25,30
	Dropout (rate)	0, 0.1, 0.2
	Gaussian Noise (SD)	0, 0.1, 0.15,0.2
	No. of dense layers	1, 2
CNN	No. of conv. layers	3,5,7
	No. of kernels	32, 64, 128
	Dropout (rate)	0.5, 0.3, 0.2

Table 3: Hyperparameter Tuning

MLP, GRU, LSTM and CNN models are implemented using Keras[8]. Adam optimizer is used for training the models. A major problem we faced while training our models is overfitting. Figure 3 shows the train and validation set accuracy with respect to the number of epochs for our LSTM model. It also shows the corresponding loss value. Training loss decreases exponentially as the number of epochs increases whereas validation loss increases after some time. To overcome overfitting, we tried multiple methods like dropout, regularization, adding Gaussian noise, making the model simpler. We also tried early stopping to avoid overfitting (Liu et al., 2015). Solving overfitting problem results in underfitting, it is more like a tradeoff between overfitting and underfitting.

6 Result

Correctly classifying tweets into 4-way regional categories is not an easy task even for humans. Performance of any human will be more or less similar to random classification. Given that, we

[7] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html

[8] https://keras.io/

now evaluate our model's performance with prior works.

Model	**Accuracy (%)**
CNN	57.43
GRU	56.35
LSTM	55.54
MLP	50.59

Table 4: Our model’s performance

In comparison with (Eisenstein et al., 2010) models, our convolutional and recurrent models are performing very much identical to their best model with an accuracy of around 57% (Table 1 and 4). Our baseline MLP model is still behind Geo topic and Unigram models (Eisenstein et al., 2010) but displays a superior performance when compared to other (Eisenstein et al., 2010) models. Whereas our models still need to be improved when compared to (Liu et al., 2015) SDA-1 which has an accuracy of about 61%. One significant difference between our work and (Liu et al., 2015) work is that (Liu et al., 2015) uses a much larger UTGEO2011[9] dataset along with GeoText dataset. Refer to Table 4 for the exact performance of our models. We also have to consider the fact that GeoText dataset is a smaller dataset captured over a period of seven days and it is distributed unevenly across regions with almost 80% of the dataset representing the Northeast and South regions (Table 2). This skewed nature of GeoText dataset likely to reduce the accuracy of the classification task.

7 Conclusion and Future Work

This project extends over several existing approaches leveraging the GeoText dataset to locate Twitter users using only their tweet content. We extensively studied and implemented a new approach using modern recurrent and convolutional network architectures.

Even with a considerably smaller dataset, recurrent and convolutional neural networks achieved a significant accuracy. Accuracy can be enhanced further by using massive datasets like UTGeo2011 by solving the overfitting problem. Twitter's doubling of character count in recent times will increase the average tweet length. Increase in tweet length[10] will likely aid LSTM, GRU models to capture the language variation better. Further development of the neural network with additional hyperparameter tuning will enhance the results.

8 References

Eisenstein J., O'Connor B., Smith N A., Xing E P. 2010. *A Latent Variable Model for Geographic Lexical Variation*. In Proceedings of the Conference on Empirical Methods in Natural Language Processing.

Liu J., Inkpen D. 2015. *Estimating User Location in Social Media with Stacked Denoising Auto-encoders*. Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing.

Yin W., Kann K., Yu M., Hinrich S. 2017. *Comparative Study of CNN and RNN for Natural Language Processing.* arXiv:1702.01923.

[9] http://www.cs.utexas.edu/~roller/research/kd/corpus/README.txt

[10] https://techcrunch.com/2018/10/30/twitters-doubling-of-character-count-from-140-to-280-had-little-impact-on-length-of-tweets/