

On Predicting Geolocation of Tweets using Convolutional Neural Networks

Binxuan Huang and Kathleen M. Carley

School of Computer Science, Carnegie Mellon University,
5000 Forbe Ave., Pittsburgh, United States
{binxuanh,kathleen.carley}@cs.cmu.edu

Abstract. In many Twitter studies, it is important to know where a tweet came from in order to use the tweet content to study regional user behavior. However, researchers using Twitter to understand user behavior often lack sufficient geo-tagged data. Given the huge volume of Twitter data there is a need for accurate automated geolocating solutions. Herein, we present a new method to predict a Twitter user’s location based on the information in a single tweet. We integrate text and user profile meta-data into a single model using a convolutional neural network. Our experiments demonstrate that our neural model substantially outperforms baseline methods, achieving 52.8% accuracy and 92.1% accuracy on city-level and country-level prediction respectively.

1 Introduction

Recently, there is growing interest in using social media to understand social phenomena. For example, researchers have shown that analyzing social media reveals important geospatial patterns for keywords related to presidential elections[23]. People can use Twitter as a sensor to detect earthquakes in real-time[22]. Recent research also has demonstrated that Twitter data provides real-time assessments of flu activity[1].

Using Twitter’s API¹, a keyword search can be done and we can easily get tweet streams from across the world containing keywords of interest. However, we cannot conduct a fine-grained analysis in a specific region using such a keyword-based search method. Alternatively, using the same API tweets with geo-information can be collected via a bounding box. Since less than 1% of tweets are tagged with geo-coordinates[7], using this location-based search means we will lose the majority of the data. If we can correctly locate those ungeotagged tweets returned from a keyword search stream, that would enable us to study users in a specific region with far more information.

With this motivation, we are aiming to study the problem of inferring a tweet’s location. Specifically, we are trying to predict on a tweet by tweet basis, which country and which city it comes from. Most of the previous studies rely on rich user information(tweeting history and/or social ties), which is time-consuming to collect because of the Twitter API’s speed limit. Thus those methods could not be directly applied to Twitter streams. In this paper, we study a

¹ <https://dev.twitter.com/docs>

global location prediction system working on each single tweet. One data sample is one tweet JSON object returned by Twitter’s streaming API. Our system utilizes location-related features in a tweet, such as text and user profile meta-data. We summarize useful features that can provide information for location prediction in Table 1.

Table 1. Feature table

Feature	Type
Tweet content	Free text
User personal description	Free text
User name	Free text
User profile location	Free text
Tweet language(TL)	Categorical
User language(UL)	Categorical
Timezone(TZ)	Categorical
Posting time(PT)	UTC timestamp

Recent research has shown that using bag-of-words and classical machine learning algorithms such as Naive Bayes can provide us a text-based location classifier with good accuracy[9]. Different from previous research, we intend to use the convolutional neural network(CNN) to boost prediction power. Inspired by the success of convolutional neural network in text classification[12], we are going to use CNN to extract location related features from texts and train a classifier that combines high-level text feature representations with these categorical features. To benchmark our method, we compared our approach with a stacking-based method. Experimental results demonstrate that our approach achieves 92.1% accuracy on country-level prediction and 52.8% accuracy on city-level prediction, which greatly outperforms our baseline methods on both tasks.

2 Related work

Identifying demographic details of Twitter users[16] has been widely studied in previous literature including inferring users’ attributes like age, gender[5], and personalities like openness and conscientiousness[18]. Among these research, there is increasing interest in inferring Twitter user’s location, which is largely driven by the lack of sufficient geo-tagged data[7]. In many situations, it is important to know where a tweet came from in order to use the information in the tweet to effect a good social outcome. Key examples include: disaster relief[14], earthquake detection[6], and predicting flu trends[1].

A majority of previous works either focus on a local region e.g. United States[4], Sweden[2], or using rich user information like a certain number of tweets for each user[4], user’s social relationship[11,17]. Different from these works, this paper works on worldwide tweet location prediction. We only utilized features in one single tweet without any external information. Thus this method could be easily applied to real-time Twitter stream.

For fine-grained location prediction, there are several types of location representation methods existing in literature. One typical method is to divide earth

into small grids and try to predict which cell one tweet comes from. Wing and Baldridge introduced a grid-based representation with fixed latitude and longitude[24]. Based on the similarity measured by Kullback-Leibler divergence, they assign each ungeotagged tweet to the cell with most similar tweets. Because cells in urban area tend to contain far more tweets than the ones in rural areas, the target classes are rather imbalanced[8]. To overcome this, Roller et al. further proposed an adaptive grid representation using K-D tree partition[20]. Another type of representation is topic region. Hong et al. proposed a topic model to discover the latent topic words for different regions[10]. Such parametric generative model requires a fixed number of regions. However, the granularity of topic regions is hard to control and will potentially vary over time[9].

The representation we choose is city-based representation considering most tweets come from urban area. One early work proposed by Cheng et al. using a probabilistic framework to estimate Twitter user’s city-level location based on the content of tweets[4]. Their framework tries to identify local words with probability distribution smoothing. However, such method needs a certain number of tweets(100) for each user to get a good estimation. Han et al. proposed a stacking-based approach to predict user’s city[8]. They combine tweet text and meta-data in user profile with stacking[25]. Specifically, they train a multinomial naive Bayes base classifier on tweet text, profile location, timezone. Then they train a meta-classifier over the base classifiers. More recently, Han et al. further did extensive experiments to show that using feature selection method, such as information gain ratio[19] could greatly improve the classification performance.

3 Location prediction

In this section, we will introduce our location prediction approach. We first briefly describe the useful features in a tweet JSON object. After that, we will further explain how we utilize these features in our prediction model.

3.1 Feature Set

We have listed all useful information we want to utilize in Table 1. Tweet content, user personal description, user name and profile location are four text fields that we will use. Twitter users often reveal their home location in their profile location and personal description. However, location indicating words are often mixed with informal tweet text(e.g. chitown for Chicago). It is unrealistic to use a gazetteer to find these words. In this work, we choose to apply CNN on these four text fields to extract high-level representations.

In addition to these four text fields, there are another three categorical features: tweet language, user language, and timezone. Tweet language is automatically determined by Twitter’s language detection tool. User language and timezone are selected by the user in his/her profile. These three categorical features are particularly useful for distinguishing users at the country-level.

The last feature is UTC posting time. Using posting timestamp as a discriminative feature is motivated by the fact that people in a region are more active on Twitter at certain times during the day. For example, while people in United Kingdom start to be active at 9:00 am in UTC time, most of the people in United

States are still asleep. We transform the posting time in UTC timestamp into discrete time slots. Specifically, we divide 24 hours into 144 time slots each with a length of 10 minutes. Thus each tweet will have a discrete time slot number in the range of 144, which can be viewed as a categorical feature. In Figure 1, we plotted the probability distribution of an user posting tweets in each time slot in three different countries. As expected, there is a big variance between these three countries.

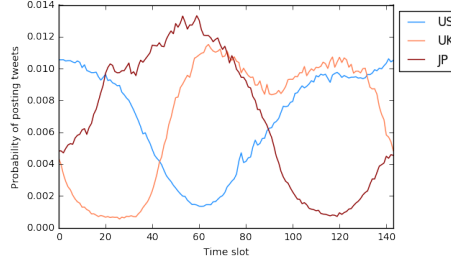


Fig. 1. The probability of an user posting a tweet in different time slot in three different countries: United States, United Kingdom, Japan.

3.2 Our Approach

Our approach is based on the convolutional neural network for sentence classification proposed by Kim[12]. Different from traditional bag-of-words method, such convolutional neural networks take the word order into consideration. Our model architecture is shown in Figure 2. We use this CNN architecture to extract high-level features from four text fields in a tweet. Let $x_i^t \in R^k$ be the k -dimensional word vector corresponding to the i -th word in the text t , where $t \in \{\text{tweet content, user description, profile location, user name}\}$. As a result, one text of length n can be represented as a matrix

$$X_{1:n}^t = x_1^t \oplus x_2^t \oplus \dots \oplus x_n^t \quad (1)$$

where \oplus is concatenation operator. In the convolutional layer, we apply each filter $w \in R^{hk}$ to all the word vector matrices, where h is the window size and k is the length of a word vector. For example, applying filter w to a window of word vectors $x_{i:i+h-1}^t$, we generated $c_i^t = f(w \cdot x_{i:i+h-1}^t + b)$. Here $b \in R$ is a bias term and we choose $f(x)$ as a non-linear ReLU function $\max(x, 0)$. Sliding the filter window from the beginning of a word matrix till the end, we generated a feature vector $c^t = [c_1^t, c_2^t, \dots, c_{n-h+1}^t]$ for each text t . If we have m filters in the convolutional layer, then we can produce m feature vectors for each text field and $4m$ vectors in total.

In the max-pooling layer, we apply a pooling operation over each feature vector generated in the convolutional layer. Each pooling operation takes a feature vector as input and outputs the maximum value $\hat{c}^t = \max(c^t)$. \hat{c}^t can be viewed as the most representative feature generated by a filter on text t . Hence we finally got a long vector $\theta \in R^{4m}$ after the max-pooling layer. To avoid the co-adaptation of hidden units, we apply dropout on the max-pooling layer that randomly set elements in θ to zero in the training phase. After that, we append four categorical features tweet language(TL), user language(UL), timezone(TZ)

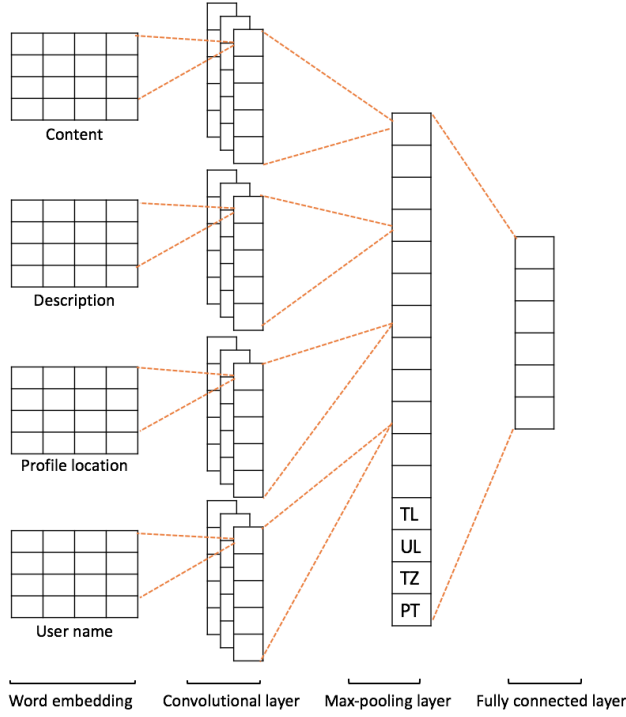


Fig. 2. A diagram of the architecture of our neural model.

and posting time(PT) with one-hot encoding at the end of θ and get $\hat{\theta}$. In the last fully connected layer, we use a softmax function over this long vector $\hat{\theta}$ to generate the probability distribution over locations. Specifically, the probability of one tweet coming from location l_i is

$$P(l_i|\hat{\theta}) = \frac{\exp(\beta_i^T \hat{\theta})}{\sum_{j=1}^L \exp(\beta_j^T \hat{\theta})} \quad (2)$$

where L is the number of locations and β_i are parameters in softmax layer. The output predicted location is just the location with highest probability.

The minimization objective in the training phase is the categorical cross-entropy loss. The parameters to be estimated include word vectors, weight vectors w for each filters, the weight vectors β in softmax layer, and all the bias terms. The optimization is performed using mini-batch stochastic gradient descent and back-propagation[21].

4 Data

We used geo-tagged tweets collected from Twitter streaming API² for training and evaluation. In this study, we set the geographic bounding box as [-180, -90, 180, 90] so that we could get these geo-tagged tweets from the whole world. Our collection started from January 7, 2017 to February 1, 2017. Because it is very common for one user to post tweets from the same city, we randomly chose one tweet for each city that one user has visited. This could ensure that there

² <https://dev.twitter.com/streaming/reference/post/statuses/filter>

is no strong overlap among our data samples. We only used tweets either with specific geo-coordinates or a geo-bounding box smaller than $[0.1, 0.1]$. For the latter case, we used the center of one tweet’s bounding box as its coordinates. No other filtering was done. There are 3,321,194 users and 4,645,692 tweets in total. For test data we used all tweets from 10% of the users who were randomly selected. For the remaining 90% users, we picked tweets from 50,000 of them as a development set and used the remaining tweets as training data.

There are two location prediction tasks we consider in this paper. The first task is country-level location prediction. We adopted the country code in the geo-tagged tweet as the label we want to predict. In our dataset, there are 243 countries and regions in total. The second task is city-level location classification. We adopt the same city-based representation as Han et al.[3]. The city-based representation consists of 3,709 cities throughout the world and was obtained by aggregating smaller cities with the largest nearby city. We assigned the closest city for each tweet based on orthodromic distance. Table 2 contains basic statistics about our dataset. It is worth mentioning that this dataset is rather imbalanced, where a majority of tweets are sent from a few countries/cities.

Table 2. Summaries about the dataset. Numbers in brackets are standard deviation.

# of tweets	# of users	# of timezones	# of lang.	# of countries (or regions)	Tweets per country	# of cities	Tweets per city
4645692	3321194	417	103	243	19118.0 (99697.1)	3709	1252.5(4184.5)

5 Experiments

5.1 Evaluation Measures

Following previous work of tweet geolocation prediction[8], we used four evaluation measures listed below. One thing to note is that when we calculated the error distance we used distance between predicted city and the true coordinates in the tweet rather than the center of assigned closest city.

- Acc: The percentage of correct location predictions.
- Acc@Top5: The percentage of true location in our top 5 predictions.
- Acc@161: The percentage of predicted city which are within a 161km(100 mile) radius of the true coordinates in the original tweet to capture near-misses. This measure is only tested on city-level prediction.
- Median: The median distance from the predicted city to the true coordinates in the original tweet. This measure is only tested on city-level prediction.

5.2 Baseline Method

We compared our approach with one commonly used ensemble method in previous research works [8, 9]. We implemented an ensemble classifier based on stacking[25] with 5-fold cross validation. The training of stacking consists of two steps. First, five multinomial naive Bayes base classifiers are trained on different types of data(tweet content, user description, profile location, user name and the remaining categorical features). The outputs from the base classifiers are used to train a multinomial naive Bayes classifier in the second layer. We call such method STACKING in this paper. Same as [9], we also use information gain ratio to do feature selection on text tokens. We call STACKING with feature selection STACKING+.

5.3 Hyperparameters and Training

We used a tweet-specific tokenizer provided by NLTK³ to tokenize text fields. We built our dictionary based on the words that appeared in text, user description, and profile location. To reduce low-utility words and noise, we removed all words that had a word frequency less than 10. For our proposed approach, we used filter windows(h) of 3,4,5 with 128 feature vectors each, a dropout rate of 0.5 and batch size of 1024. We initialize word vectors using word2vec⁴ vectors trained on 100 billion words from Google News. The vectors have dimensionality of 300 and were trained using the continuous bag-of-words architecture[15]. For those words that are not included in word2vec, we initialized them randomly. We also performed early stopping based on the accuracy over the development set. Training was done through stochastic gradient descent using Adam update rule with learning rate 10^{-3} [13]. For our baseline models, we applied additive smoothing with $\alpha = 10^{-2}$, which is selected on the development set. For STACKING+ method, we first ranked these words by their information gain ratio value, then selected the top n% words as our vocabulary. The tuning of n is based on accuracy over the development set. We selected n as 40%, 55% for city-level prediction and country-level prediction respectively.

5.4 Results

The comparison results between our approach and the baseline methods are listed in Table 3 and Table 4. Our approach achieves 92.1% accuracy and 52.8% accuracy on country-level and city-level location prediction respectively. Our approach is consistently better than the previous model on the country-level location prediction task as shown in Table 3. It greatly outperforms our baseline methods over all the measures, especially on the city-level prediction task. It could assign more than half of the test tweets to the correct city and gain more than 20% relative improvement over the accuracy of the STACKING+ method.

Table 3. Country prediction results.

	Acc	Acc@Top5
STACKING	0.868	0.947
STACKING+	0.871	0.950
Our approach	0.921	0.972

Table 4. City prediction results.

	Acc	Acc@161	Acc@Top5	Median
STACKING	0.389	0.573	0.595	77.5 km
STACKING+	0.439	0.616	0.629	47.2 km
Our approach	0.528	0.692	0.711	28.0 km

Our approach performs better for countries with a large number of tweets. In Figure 3, we plotted the precision and recall value for each country as a scatter chart. The dot size is proportional to the number of tweets that come from that country. Turkey appears to be the country with highest precision and recall. These results suggest that our approach works better with more data samples.

The same graph is also plotted for city prediction in Figure 3. Because of the skewness of our data and the difficulty of city-level prediction, our classifier tends to generate labels towards big cities, which leads to high recall and low precision for cities like Los Angeles.

³ <http://www.nltk.org/api/nltk.tokenize.html>

⁴ <https://code.google.com/archive/p/word2vec/>

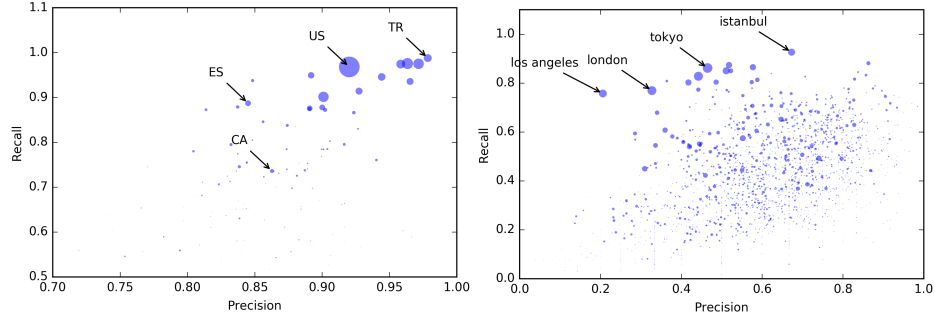


Fig. 3. Two scatter graphs that show the performances for each country and cities. The x-axis is precision, y-axis is recall. Each dot represents a country/city. The dot size is proportional to the number of tweets that comes from the corresponding location. Some tiny invisible country outside of the scope are not shown in the figure.

For real world applications, people may ask how we could set a threshold to get prediction results with high confidence. To answer this question, we further examined the relation between prediction accuracy and the output probability. Here the output probability is just the probability of our predicted location calculated by equation 2. Figure 4 shows the distribution of tweets in terms of output probability for two tasks. As expected, the prediction accuracy increases as the output probability increases. We get 97.2% accuracy for country-level prediction with output probability larger than 0.9. Surprisingly, the accuracy of city-level is as high as 92.7% for the 29.6% of the tweets with output probability greater than 0.9. However, the city-level accuracy for the remaining tweets with output probability less than 0.9 is only 48.4%. Unlike country-level prediction, the number of tweets decreases as output probability increases, unless the output probability is larger than 0.9.

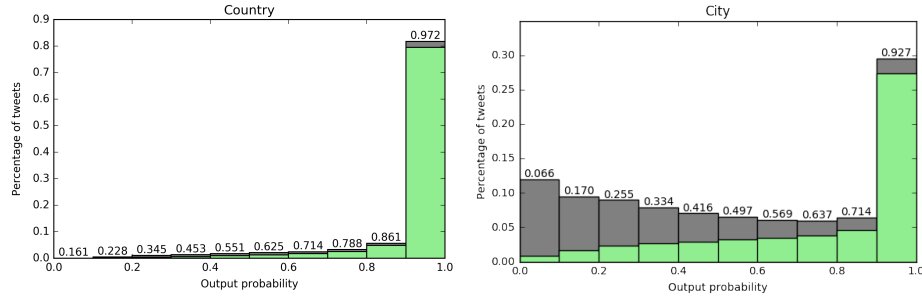


Fig. 4. Two bar charts that show the distribution of tweets in terms of the output probability. The x-axis is the output probability associated with each prediction, the y-axis is the percentage of tweets. The height of grey bar represents the percentage of test data that has certain output probability. The height of green bar represents the percentage of correctly predicted tweets in each probability range. We listed the accuracy for each probability range above the bar. Take the rightmost bar in country-level prediction for example, there are 81.8% tweets' country are predicted with output probability larger than 0.9. Among these 81.8% tweets, 97.2% are predicted correctly.

6 Discussion and Conclusion

These experiments demonstrate that our approach is consistently better than the prior method thus supporting more tweets to be accurately located by country, and city, of origin. At the country level, the more tweets that come from the country, the better the prediction. Regardless of the number of tweets per country, we can predict the country location for most tweets with extremely high confidence and accuracy. At the city level the results are more mixed. For a small fraction of tweets we can get greater than 90% accuracy, but for the rest of tweets the accuracy is less than 50%. For about half the tweets it is difficult to infer the city location. This result is partially due to the fact that we base the prediction on only a single tweet. Future work may consider using collection of tweets per user. This result is also partially due to the fact that the data is highly skewed toward a few cities. Future work should develop a training set that is more evenly distributed across cities. Despite these limitations, this approach shows promise.

This paper presents a method for geo-locating a single tweet based on the information in a tweet JSON object. The proposed approach integrates tweet text and user profile meta-data into a single model. Compared to the previous stacking method with feature selection, our approach substantially outperforms the baseline method. We developed the approach for both city and country level and demonstrated the ability to classify tweets at both levels of granularity. The results demonstrate that using a convolutional neural network utilizes the textual location information better than previous approaches and boosts the location prediction performance substantially.

Acknowledgments. This work was supported in part by the Office of Naval Research (ONR) N000140811186, and the National Science Foundation(NSF) 00361150115291. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ONR or the NSF. We want to thank tutors in the Global Communication Center at Carnegie Mellon for their valuable advice.

References

1. Achrekar, H., Gandhe, A., Lazarus, R., Yu, S.H., Liu, B.: Predicting flu trends using twitter data. In: Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on. pp. 702–707. IEEE (2011)
2. Berggren, M., Karlgren, J., Östling, R., Parkvall, M.: Inferring the location of authors from words in their texts. arXiv preprint arXiv:1612.06671 (2016)
3. Bo, H., Cook, P., Baldwin, T.: Geolocation prediction in social media data by finding location indicative words. In: Proceedings of COLING. pp. 1045–1062 (2012)
4. Cheng, Z., Caverlee, J., Lee, K.: 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. pp. 759–768. ACM (2010)
5. Culotta, A., Kumar, N.R., Cutler, J.: Predicting the demographics of twitter users from website traffic data. In: AAAI. pp. 72–78 (2015)
6. Earle, P.S., Bowden, D.C., Guy, M.: Twitter earthquake detection: earthquake monitoring in a social world. *Annals of Geophysics* 54(6) (2012)

7. Hale, S., Gaffney, D., Graham, M.: Where in the world are you? geolocation and language identification in twitter. *Proceedings of ICWSM 12*, 518–521 (2012)
8. Han, B., Cook, P., Baldwin, T.: A stacking-based approach to twitter user geolocation prediction. In: *ACL (Conference System Demonstrations)*. pp. 7–12 (2013)
9. Han, B., Cook, P., Baldwin, T.: Text-based twitter user geolocation prediction. *Journal of Artificial Intelligence Research* 49, 451–500 (2014)
10. Hong, L., Ahmed, A., Gurumurthy, S., Smola, A.J., Tsioutsoulis, K.: Discovering geographical topics in the twitter stream. In: *Proceedings of the 21st international conference on World Wide Web*. pp. 769–778. ACM (2012)
11. Jurgens, D.: That’s what friends are for: Inferring location in online social media platforms based on social relationships. *ICWSM 13*, 273–282 (2013)
12. Kim, Y.: Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014)
13. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)
14. Landwehr, P.M., Carley, K.M.: Social media in disaster relief. In: *Data mining and knowledge discovery for big data*, pp. 225–257. Springer (2014)
15. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*. pp. 3111–3119 (2013)
16. Mislove, A., Lehmann, S., Ahn, Y.Y., Onnela, J.P., Rosenquist, J.N.: Understanding the demographics of twitter users. *ICWSM 11*, 5th (2011)
17. Qian, Y., Tang, J., Yang, Z., Huang, B., Wei, W., Carley, K.M.: A probabilistic framework for location inference from social media. *arXiv preprint arXiv:1702.07281* (2017)
18. Quercia, D., Kosinski, M., Stillwell, D., Crowcroft, J.: Our twitter profiles, our selves: Predicting personality with twitter. In: *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom)*, 2011 IEEE Third International Conference on. pp. 180–185. IEEE (2011)
19. Quinlan, J.R.: *C4. 5: programs for machine learning*. Elsevier (2014)
20. Roller, S., Speriosu, M., Rallapalli, S., Wing, B., Baldrige, J.: 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*. pp. 1500–1510. Association for Computational Linguistics (2012)
21. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. *Cognitive modeling* 5(3), 1 (1988)
22. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: *Proceedings of the 19th international conference on World wide web*. pp. 851–860. ACM (2010)
23. Tsou, M.H., Yang, J.A., Lusher, D., Han, S., Spitzberg, B., Gawron, J.M., Gupta, D., An, L.: Mapping social activities and concepts with social media (twitter) and web search engines (yahoo and bing): a case study in 2012 us presidential election. *Cartography and Geographic Information Science* 40(4), 337–348 (2013)
24. Wing, B.P., Baldrige, J.: 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*. pp. 955–964. Association for Computational Linguistics (2011)
25. Wolpert, D.H.: Stacked generalization. *Neural networks* 5(2), 241–259 (1992)