Twitter Geolocation Prediction using Neural Networks

Philippe Thomas and **Leonhard Hennig**
Deutsches Forschungszentrum für Künstliche Intelligenz, Germany
firstname.lastname@dfki.de

Abstract

Knowing the location of a user is important for several use cases, such as location specific recommendations, demographic analysis, or monitoring of disaster outbreaks. We present a bottom up study on the impact of text- and metadata-derived contextual features for Twitter geolocation prediction. The final model incorporates individual types of tweet information and achieves state-of-the-art performance on a publicly available test set. The source code of our implementation, together with individual models, is freely available at github-url.blinded.for.review.

1 Introduction

Data from social media platforms is an attractive real-time resource for data analysts. It can be used for a wide range of use cases, such as monitoring of fire- (Paul et al., 2014) and flue-outbreaks (Power et al., 2013), provide location-based recommendations (Ye et al., 2010), or is utilized in demographic analyses (Sloan et al., 2013). Although some platforms, such as Twitter, allow users to geolocate posts, Jurgens et al. (2015) reported that less than 3 % of all Twitter posts are geotagged. This severely impacts the use of social media data for such location-specific applications.

The location prediction task can be either tackled as classification problem, or alternatively as a multi-target regression problem. In the former case the goal is to predict city labels for a specific tweet, whereas the latter case predicts latitude and longitude coordinates for a given tweet. Previous studies showed that text in combination with metadata can be used to predict user locations (Han et al., 2014). Liu and Inkpen (2015) presented a system based on stacked denoising auto-encoders (Vincent et al., 2008) for location prediction. State-of-the-art approaches, however, often make use of very specific, non-generalizing features based on web site scraping, IP resolutions, or external resources such as GeoNames. In contrast, we present an approach for geographical location prediction that achieves state-of-the-art results using neural networks trained solely on Twitter text and metadata. It does not require external knowledge sources, and hence generalizes more easily to new domains and languages.

The remainder of this publication is organized as follows: First, we provide an overview of related work for Twitter location prediction. In Section 3 we describe the details of our neural network architecture. Results on the test set are shown in Section 4. Finally, we conclude the paper with some future directions in Section 5.

2 Related Work

For a better comparability of our approach, we focus on the shared task presented at the 2nd Workshop on Noisy User-generated Text (WNUT'16) (Han et al., 2016). The organizers introduced a dataset to evaluate individual approaches for tweet- and user-level location prediction. For tweet-level prediction the goal is to predict the location of one specific message, while for user-level prediction the goal is to predict the user location based on a variable number of user messages. In the following, we focus on tweet-level prediction as it is more practical in real world applications (Han et al., 2016). The organizers evaluate team submissions based on accuracy and distance in kilometers. The latter metric allows to account for wrong, but geographically close predictions, for example, when the model predicts Vienna instead of Budapest.

We focus on the five teams who participated in the WNUT shared task. Official team results for tweet- and user-level predictions are shown in

Table 1. Unfortunately, only three participants provided systems descriptions, which we will briefly summarize:

Team *FujiXerox* (Miura et al., 2016) built a neural network using text, user declared locations, timezone values, and user self-descriptions. For feature preprocessing the authors build several mapping services using external resources, such as GeoNames and time zone boundaries. Finally, they train a neural network using the fastText n-gram model (Joulin et al., 2016) on post text, user location, user description, and user timezone.

Team *csiro* (Jayasinghe et al., 2016) used an ensemble learning method built on several information resources. First, the authors use post texts, user location text, user time zone information, messenger source (e.g., Android or iPhone) and reverse country lookups for URL mentions to build a list of candidate cities contained in GeoNames. Furthermore, URL mentions were scraped and the website metadata was screened for geographic coordinates. The authors implemented custom scrapers for websites which are frequently used in Twitter and sometimes provide latitude and longitude in their metadata. Second, a relationship network is built from tweets mentioning another user. Third, posts are used to find similar texts in the training data to calculate a class-label probability for the most similar tweets. Fourth, text is classified using the geotagging tool pigeo (Rahimi et al., 2016). The output of individual stages is then used in an ensemble learner.

Team *cogeo* (Chi et al., 2016) employ multinomial naïve Bayes and focus on the use of textual features (i.e., location indicative words, GeoNames gazetteers, user mentions, and hashtags).

3 Methods

We used the WNUT'16 shared task data consisting of 12,827,165 tweet IDs, which have been assigned to a metropolitan city center from the GeoNames database[1], using the strategy described in Han et al. (2012). As Twitter does not allow to share individual tweets, posts need to be retrieved using the Twitter API, of which we were able to retrieve 9,127,900 (71.2 %). The remaining tweets are no longer available, usually because users deleted these messages. In comparison, the winner of the WNUT'16 task (Miura et al., 2016) reported that they were able to successfully retrieve 9,472,450 (73.8 %) tweets. The overall training data consists of 3,362 individual class labels (i.e., GeoNames cities). In our subset of approximately 9 million tweets we only observed 3,315 different classes.

For text preprocessing, we use a simple whitespace tokenizer with lower casing, without any domain specific processing, such as unicode normalization (Davis et al., 2001) or any lexical text normalization (see for instance Han and Baldwin (2011)). The text of tweets, and metadata fields containing texts (user description, user location, user name, timezone) are converted to word embeddings (Mikolov et al., 2013), which are then forwarded to a Long Short-Term Memory (LSTM) unit (Hochreiter and Schmidhuber, 1997). In our experiments we randomly initialized embedding vectors. We use batch normalization (Ioffe and Szegedy, 2015) for normalizing inputs in order to reduce internal covariate shift. The risk of overfitting by co-adapting units is reduced by implementing dropout (Srivastava et al., 2014) between individual neural network layers. An example architecture for textual data is shown in Figure 1. Mentions of links in the post are handled slightly differently by building character embeddings and feeding them into a LSTM layer. Metadata fields with a finite set of elements (UTC time and source type) are directly represented as one-hot encodings.

We connect all eight individual neural architectures with a dense layer for classification using a softmax activation function. We use stochastic gradient descent over shuffled mini-batches with Adam (Kingma and Ba, 2014) and cross-entropy loss as objective function for classification. For parameter tuning we tested different properties on a randomly selected validation set consisting of 50,000 tweets. The final parameters of our model are shown in Table 3.

The WNUT'16 task requires the model to predict class labels and longitude/latiude pairs. To account for this, we predict the mean city longitude/latitude location given the class label. For user-level prediction, we classify all messages individually and predict the city label with the highest probability over all messages.

3.1 Model combination

The internal representations for all eight different resources (i.e., *text*, *user-description*, *user-location*, *user-name*, *user-timezone*, *links*, *UTC*, and *source*) are concatenated to build a final tweet represen-

[1] http://www.geonames.org/

Submission	Tweet			User		
	Acc	Median	Mean	Acc	Median	Mean
FujiXerox.2	0.409	**69.5**	**1,792.5**	0.476	**16.1**	1,122.3
csiro.1	**0.436**	74.7	2,538.2	**0.526**	21.7	1,928.8
FujiXerox.1	0.381	92.0	1,895.4	0.464	21.0	**963.8**
csiro.2	0.422	183.7	2,976.7	0.520	23.1	2,071.5
csiro.3	0.420	226.3	3,051.3	0.501	30.6	2,242.4
Drexel.3	0.298	445.8	3,428.2	0.352	262.7	3,124.4
aist.1	0.078	3,092.7	4,702.4	0.098	1,711.1	4,002.4
cogeo.1	0.146	3,424.6	5,338.9	0.225	630.2	2,860.2
Drexel.2	0.082	4,911.2	6,144.3	0.079	4,000.2	6,161.4
Drexel.1	0.085	5,848.3	6,175.3	0.080	5,714.9	6,053.3

Table 1: Official WNUT'16 tweet- and user-level results ranked by tweet median error distance (in kilometers). Individual best results for all three criteria are highlighted in bold face.

Parameter	Property
Description embedding dim.	100
Link embedding dim.	100
Location embedding dim.	50
Name embedding dim.	100
Text embedding dim.	100
Timezone embedding dim.	50
Batch-Size	256

Table 2: Selected parameter settings

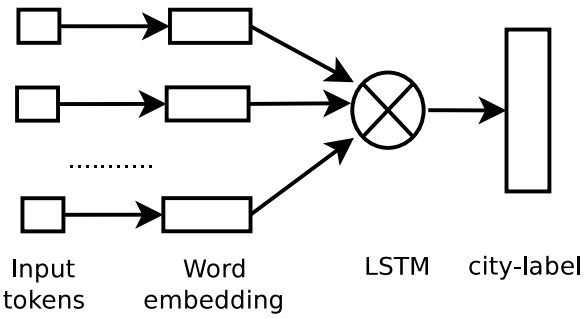

Figure 1: Example architecture used for textual data. Tokenized text is represented as word embeddings, which are then forwarded to a LSTM. Dropout and batch normalization is applied between individual layers.

tation. We then evaluate two training strategies: In the first training regime, we train the combined model from scratch. The parameters for all word- and character-level embeddings, as well as all network layers, are initialized randomly. The parameters of the full model including the softmax layer combining the output of the six individual LSTM models and the two metadata models are learned jointly. For the second strategy, we first train each LSTM model separately, and then keep their parameters fixed while training only the final softmax layer.

4 Results

The individual performance of our different models is shown in Table 4. As simple baseline, we predict the city label most frequently observed in the training data (Jakarta in Indonesia). According to our bottom-up analysis, the user-location metadata is the most productive kind of information for tweet- and user-level location prediction. Using the text alone, we can correctly predict the location for 19.3 % of all tweets with a median distance of 2,128 kilometers to the correct location. Aggregation of pretrained models also increases performance for all three evaluation metrics in comparison to training a model from scratch.

For tweet-level prediction, our best merged model outperforms the best submission (*FujiXerox.2*) in terms of accuracy, median and mean distance by 1.4 percentage points, 18.4 kilometers, and 392.1 kilometers respectively. The ensemble learning method (*csiro*) outperforms our best models in terms of accuracy by 1.3 percentage points,

Model	Tweet			User		
	Acc	**Median**	**Mean**	**Acc**	**Median**	**Mean**
location	0.362	209.4	4,535.7	0.441	45.9	3,841.8
text	0.193	2,128.4	4,404.3	0.322	266.4	2,595.0
description	0.087	3,806.7	6,048.9	0.097	3,407.9	5,896.8
user-name	0.059	3,942.5	5,990.1	0.058	4,153.4	6,116.0
timezone	0.062	6,504.1	7,144.1	0.062	6,926.3	7,270.9
UTC	0.050	6,610.3	7,191.9	0.050	6,530.9	7,211.7
links	0.033	7,593.4	6,978.6	0.045	6,732.0	6,554.3
source	0.044	8,029.0	7,528.2	0.045	6,950.8	6,938.5
full-scratch	0.417	59.0	1,616.4	0.513	17.8	1,023.9
full-fixed	**0.423**	**51.1**	**1,400.4**	**0.524**	**15.9**	**916.1**
baseline	0.028	11,723.0	10,264.3	0.024	11,771.5	10,584.4

Table 3: Tweet level results ranked by median error distance (in kilometers). Individual best results for all three criteria are highlighted in bold face. Full-scratch refers to a merged model trained from scratch, whereas the weights of the full-fixed model are only retrained where applicable. The baseline predicts the location most frequently observed in the training data (Jakarta).

but our model performs considerably better on median and mean distance with 23.6 and 1137.8 kilometers respectively. Additionally, the approach of *csiro* requires several dedicated services, such as GeoNames gazetteers, time zone to GeoName mappings, IP country resolver and customized scrapers for social media websites. The authors describe custom link handling for FourSquare, Swarm, Path, Facebook, and Instagram. On our training data we observed that these websites account for 1,941,079 (87.5 %) of all 2,217,267 shared links. It is therefore tempting to speculate that a customized scraper for these websites could further boost our results for location prediction.

As team *cogeo* uses only the text of a tweet, the results of *cogeo.1* are comparable with our text-model. The results show that our text-model outperforms this approach in terms of accuracy, median and mean distance to the gold standard by 4.7 percentage points, 1296 kilometers, and 934 kilometers respectively.

For user-level prediction, our method performs on a par with the individual best results collected from the three top team submissions (*FujiXerox.2*, *csiro.1*, and *FujiXerox.1*).

5 Conclusion

We presented our neural network architecture for the prediction of city labels and geo-coordinates for tweets. We focus on the classification task and derive longitude/latitude information from the city label. We evaluated models for individual Twitter (meta)-data in a bottom up fashion and identified highly location indicative fields. The proposed combination of individual models requires no customized text-preprocessing, specific website crawlers, database lookups or IP to country resolution while achieving state-of-the-art performance on a publicly available data set. For better comparability, source code and pretrained models is freely available to the community.

As future work, we plan to incorporate images as another type of metadata for location prediction using the approach presented by Simonyan and Zisserman (2014).

Acknowledgments

This research was partially supported by the German Federal Ministry of Economics and Energy (BMWi) through the projects SD4M (01MD15007B) and SDW (01MD15010A) and by the German Federal Ministry of Education and Research (BMBF) through the project BBDC (01IS14013E).

References

Lianhua Chi, Kwan Hui Lim, Nebula Alam, and Christopher J. Butler. 2016. Geolocation prediction in twitter using location indicative words and textual features. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 227–234, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Mark Davis, Ken Whistler, and Martin Dürst. 2001. Unicode Normalization Forms. Technical report, Unicode Consortium.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 368–378, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bo Han, Paul Cook, and Timothy Baldwin. 2012. Geolocation prediction in social media data by finding location indicative words. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India*, pages 1045–1062.

Bo Han, Paul Cook, and Timothy Baldwin. 2014. Text-based twitter user geolocation prediction. *J. Artif. Int. Res.*, 49(1):451–500, January.

Bo Han, Afshin Rahimi, Leon Derczynski, and Timothy Baldwin. 2016. Twitter geolocation prediction shared task of the 2016 workshop on noisy user-generated text. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 213–217, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167.

Gaya Jayasinghe, Brian Jin, James Mchugh, Bella Robinson, and Stephen Wan. 2016. Csiro data61 at the wnut geo shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 218–226, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759.

David Jurgens, Tyler Finethy, James McCorriston, Yi Tian Xu, and Derek Ruths. 2015. Geolocation prediction in twitter using social networks: A critical analysis and review of current practice. In *ICWSM*, pages 188–197.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Ji Liu and Diana Inkpen. 2015. Estimating user location in social media with stacked denoising autoencoders. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 201–210, Denver, Colorado, June. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

Yasuhide Miura, Motoki Taniguchi, Tomoki Taniguchi, and Tomoko Ohkuma. 2016. A simple scalable neural networks based model for geolocation prediction in twitter. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 235–239, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Michael J Paul, Mark Dredze, and David Broniatowski. 2014. Twitter improves influenza forecasting. *PLOS Currents Outbreaks*.

Robert Power, Bella Robinson, and David Ratcliffe. 2013. Finding fires with twitter. In *Australasian language technology association workshop*, volume 80.

Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. 2016. pigeo: A python geotagging tool. In *Proceedings of ACL-2016 System Demonstrations*, pages 127–132, Berlin, Germany, August. Association for Computational Linguistics.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

Luke Sloan, Jeffrey Morgan, William Housley, Matthew Williams, Adam Edwards, Pete Burnap, and Omer Rana. 2013. Knowing the Tweeters: Deriving sociologically relevant demographics from Twitter. *Sociological Research Online*, 18(3).

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1096–1103, New York, NY, USA. ACM.

Mao Ye, Peifeng Yin, and Wang-Chien Lee. 2010. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 458–461, New York, NY, USA. ACM.