A Simple Scalable Neural Networks based Model for Geolocation Prediction in Twitter

Yasuhide Miura
Fuji Xerox Co., Ltd. / Japan
yasuhide.miura@fujixerox.co.jp

Motoki Taniguchi
Fuji Xerox Co., Ltd. / Japan
motoki.taniguchi@fujixerox.co.jp

Tomoki Taniguchi
Fuji Xerox Co., Ltd. / Japan
taniguchi.tomoki@fujixerox.co.jp

Tomoko Ohkuma
Fuji Xerox Co., Ltd. / Japan
ohkuma.tomoko@fujixerox.co.jp

Abstract

This paper describes a model that we submitted to W-NUT 2016 Shared task #1: Geolocation Prediction in Twitter. Our model classifies a tweet or a user to a city using a simple neural networks structure with fully-connected layers and average pooling processes. From the findings of previous geolocation prediction approaches, we integrated various user metadata along with message texts and trained the model with them. In the test run of the task, the model achieved the accuracy of 40.91% and the median distance error of 69.50 km in message-level prediction and the accuracy of 47.55% and the median distance error of 16.13 km in user-level prediction. These results are moderate performances in terms of accuracy and best performances in terms of distance. The results show a promising extension of neural networks based models for geolocation prediction where recent advances in neural networks can be added to enhance our current simple model.

1 Introduction

The growth of social media has brought demands to develop techniques to automatically extract a variety of information from it. A geolocation predictor is one such technique that has been widely studied to strengthen business activities such as advertisement and marketing. This paper describes a model that we developed to tackle this geolocation prediction task. The model is a neural networks based model that uses various user metadata along with message texts. We participated W-NUT 2016 Shared task #1: Geolocation Prediction in Twitter (Rahimi et al., 2016) with this model.

The architecture of our model is designed by following previous researches that have targeted geolocation prediction in Twitter. Cheng et al. (2010) proposed a probabilistic model based on local (location indicative) words to estimate city-level geolocation prediction of Twitter users. Han et al. (2014) implemented a city-level geolocation prediction system for Twitter users, and constructed a Naive Bayes learner using location indicative words and user metadata. We decided to use both message texts and user metadata which have shown effectiveness in these past studies.

Another characteristic of our model is that we decided to use neural networks as a machine learning method to train a geolocation predictor. Neural networks are being extensively applied to tasks of image processing, speech processing, or text processing and have shown the state of the performances (Goodfellow et al., 2016). In a context of geolocation prediction, Liu and Inkpen (2015) have showed that a model based on stacked denoising auto-encoders achieves a comparable performance to the state of the arts models. Although neural networks with deep complex layers have recently received attention as achieving high performance, we designed our model to be a simple scalable neural networks based model considering the following two reasons:

1. A city-level geolocation prediction of Twitter is a large scale task that requires a scalable machine learning method which can handle millions of tweet with thousands of classes (Han et al., 2014).
2. A model based on shallow simple neural networks has recently marked close performance to deep models with high scalability on several tasks (Joulin et al., 2016).

In the following sections of this paper, we first describe the specification of datasets to train and to test our model in Section 2. The detail of our neural networks based model is explained in Section 3,

Name	#tweet	#user
Training (tweet/user)	9026924	781889
Validation (tweet)	7260	4725
Validation (user)	94872	7826
Training10 (user)	7809590	622133
Validation10 (user)	87084	6807

Table 1: The number of tweet and the number of user we collected for each dataset.

Name	#tweet	#user
Experimental Training (tweet)	8846145	766251
Pseudo Test (tweet)	7819	7819
Experimental Training10 (user)	7653295	609691
Pseudo Test10 (user)	78173	6221

Table 2: A summary of the experimental training datasets and the pseudo test datasets.

accompanying Section 4 that explains experiments and the test run which were performed to evaluate the model. Finally, Section 5 summarizes and concludes the paper with some future directions.

2 Data

2.1 Tweets and Users

We collected the training dataset and the validation dataset using the official download script. Table 1 shows the statistics of tweets and users that we successfully downloaded with the script. For the user geolocation prediction task, we only used users that have 10 or more tweets in the datasets. Training10 (user) and Validation10 (user) in Table 1 are the subsets of Training (tweet/user) and Validation (user) that fulfill this condition. We used Training10 (user) and Validation10 (user) in the experiments of user-level prediction systems which will be described in Section 4.

The training datasets are further split to experimental training datasets and pseudo test datasets since the test datasets were not available at the development period of the task. For each training dataset, 98% and 1% of the users were randomly sampled as an experimental training dataset and a pseudo test dataset respectively[1]. Furthermore, one tweet per user were selected to form Pseudo Test (tweet) so that the number of tweets in the dataset becomes close to Validation (tweet). Table 2 summarizes these experimental training datasets and pseudo test datasets.

2.2 Geo Coordinates of Tweet Cities

We realized that the tweet-level geo coordinates of the training data and the validation data are not the geo coordinates of the corresponding tweet cities. To obtain the geo coordinates of the tweet cities, we replaced the geo coordinates of the tweets with user city geo coordinates. For tweet cities that are not apparent in the user cities, we referred to GeoNames[2] to obtain geo coordinates of them.

3 Model

The training datasets consists of 8–9M tweets and the number of class is over 3K in the geolocation prediction task. Because of the large dataset size and the number of class, we decided to construct a simple and efficient model. Our model is a variant of Joulin et al. (2016), that is called fastText. Although the fastText model is simple, the accuracy of fastText is on par with deep complex models on several tasks. Moreover the fastText model trains much faster compared to typical deep complex models. The original fast text model is targeted for text classification that deals with one sentence. The Twitter

[1]The remaining 1% of the users were kept for an alternative pseudo test dataset but were not used in the experiments of our submission systems.

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

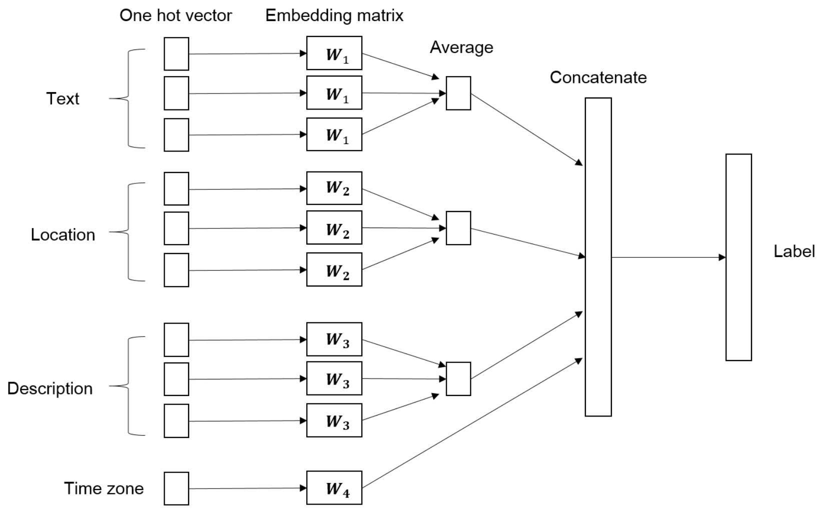

Figure 1: Model architecture.

Parameter	Message-level	User-level
Text word embedding dimension	300	300
Location word embedding dimension	300	300
Description word embedding dimension	300	300
Time zone embedding dimension	50	100
n-gram	2	2
Batch size	1500	160

Table 3: The parameters of the best performing models.

data in this shared task includes not only message text but also user metadata. Therefore we extend fast text model to deal with message text and user metadata.

Our model for message-level prediction is shown in Figure 1. We use the text field of tweet and three metadata fields of user: location, description, and time zone. In the text, the location, and the description fields, the word embeddings of words in a field are averaged to form a representation of the field. The weights of word embeddings are not shared across the fields because the unique weights resulted to better performance than the shared weights. Similar to fastText, we also use bag of n-gram features to capture word orders. According to Guo and Berkhahn (2016), the embeddings of categorial variables can reduce the network size and capture the intrinsic properties of the categorical variables. Therefore, we decided to embed the value of the time zone field into a fixed size vector as a time zone representation. All representations are then concatenated as a tweet representation. Finally the tweet representation is fed into a linear classification layer. The objective function of our model is defined as cross-entropy loss. We minimize the objective function through stochastic gradient descent over shuffled mini-batches with Adam (Kingma and Ba, 2014).

We basically used the same model for user-level prediction except that the text fields of tweets in a user instance are concatenated into one long text. The three metadata fields are also used in the user level predictor.

4 Experiments

4.1 Model Configurations

Text Processors

We used Twokenizer (Owoputi et al., 2013) to segment words from texts. Prior to word segmentations, all upper case characters in texts were converted to lower case correspondants and Unicode Normalization

	Validation	Pseudo Test	Test		
	Acc. (%)	Acc. (%)	Acc. (%)	Median Dist. Err.	Mean Dist. Err.
Message-level	39.1	39.6	40.91	69.50	1792.53
User-level	45.0	45.2	47.55	16.13	1122.26

Table 4: Evaluation results of the test run with the accuracies of the validation datasets and the pseudo test datasets.

in form NFKC[3] is applied to texts. We also normalized Twitter user names to "USER" and URLs to "URL" using regular expressions.

Pre-training of Word Embeddings

In our model, words are input to the model as n-dimensional word embeddings. We pre-trained word embeddings using the text, the location, and the description fields of the full training data (Training (tweet/user) in Section 2) and the method of Bojanowski et al. (2016)[4] with skip gram algorithm to speed up a training process of the user-level prediction task. For the parameters of skip gram, we used learning rate=0.025, window size=5, negative sample size=5, and epoch=5.

Model Parameters

We prepared three user-level systems and two message-level systems for the submission. These systems are all variants of the model explained in Section 3 with different parameters. Table 3 lists the parameters that are used in the user-level prediction system 2 and the message-level prediction system 2 which were our best systems in the test run. These parameters were decided prior to the submission with grid search using the validation datasets and the pseudo test datasets.

4.2 Evaluations

The message-level prediction system 2 and the user-level prediction system 2 were evaluated on the test run with classification accuracy, median distance error, and mean distance error. Table 4 shows the result of the test run along with the accuracies of the validation datasets and the pseudo test datasets. The performances of our model was moderate in terms of accuracies compared to the best systems (tweet-level-best=43.62% and user-level-best=52.65%) and best in terms of median distance errors.

5 Conclusion

We developed a simple neural networks based model for geolocation prediction in Twitter. In the test run, the model performed the accuracy of 40.91% and the median distance error of 69.50 km in message-level prediction and the accuracy of 47.55% and the median distance error of 16.13 km in user-level prediction. These results indicate that a neural networks based model has high scalability and is a competitive approach in geolocation prediction.

As future work, we plan to expand our model to have a deeper complex structure which has shown effectiveness in other tasks. We found out that the current model is quite simple only taking several hours of training time with a use of a single modern GPU. Some expansions using current advances in neural networks can possibly boost the performances of the model without losing the scalability of the model.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments to improve this paper.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606.*

[3]http://unicode.org/reports/tr15/

[4]https://github.com/facebookresearch/fastText

Zhiyuan Cheng, James Caverlee, and Kyumin Lee. 2010. 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*, CIKM '10, pages 759–768.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. Deep learning. Book in preparation for MIT Press (http://www.deeplearningbook.org).

Cheng Guo and Felix Berkhahn. 2016. Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737*.

Bo Han, Paul Cook, and Timothy Baldwin. 2014. Text-based Twitter user geolocation prediction. *Journal of Artificial Intelligence Research*, 49(1):451–500.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ji Liu and Diana Inkpen. 2015. 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*, pages 201–210.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–390.

Afshin Rahimi, Bo Han, 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 (W-NUT)*. ACL.