Twitter Photo Geo-Localization Using Both Textual and Visual Features

Shin Matsuo Wataru Shimoda Keiji Yanai
Department of Informatics, The University of Electro-Communications, Tokyo
1-5-1 Chofugaoka, Chofu-shi Tokyo 182-8585 JAPAN
Email: {matsuo-s,shimoda-k,yanai}@mm.inf.uec.ac.jp

***Abstract*—In this paper, we propose a method to add geotags to Twitter photos which have no geotags. Our objective is localizing a Twitter photo using both textual features and visual features. For localization from texts, we use GeoNLP which estimates location from location names and location-dependent named entity. For localization from visual features, we use image search for a geotagged photo database. As visual features, we use local features and deep neural features and compare them. In the experiments, combining texts and deep neural feature based image search achieved the best results.**

***Keywords*-Twitter photo, photo localization, bag-of-features, SIFT, DCNN**

I. Introduction

Because microblogs such as Twitter and Weibo has unique characteristics which are different from other social media in terms of timeliness and on-the-spot-ness, they include much information on various events in the real world. By mining photos from microblogs, we can get to know and understand what happens in the world visually and intuitively. To do that, locations where photo were taken and tweets were posted are useful information. However, the ratio of "geotagged photo tweets" to all the photo Tweets is very limited.

Then, in this paper, we propose a method to add geotags to Twitter photos which have no geotags. Our objective is localizing a Twitter photo using both textual features and visual features. For localization from texts, we use GeoNLP [1] which estimates location from location names and location-dependent named entity. For localization from visual features, we use image search for a geotagged photo database. As visual features, we use local features and deep neural features and compare them. As local features, we use SIFT features and bag-of-feature representation, while we use Deep Convolutional Neural Network (DCNN) features.

In the experiments, combining texts and deep neural feature based image search achieved the best results.

[1] http://agora.ex.nii.ac.jp/GeoNLP/

II. Related work

Watanabe et al. [1] proposed a method to estimate locations of Tweets from Tweet message texts. IM2GPS by Hays et al. [2] showed that image retrieval for a large-scale geotagged image database enabled us to estimate location of a single image. As a benchmark for location estimation of images, the Placing Task at MediaEval exists. In the task, most of the participants adopts multi-modal location estimation approach by combining visual features and textual features in the same way as our work.

Recently, as a method for object recognition, Deep Neural Network is much paid attention due to its high performance. To train a DCNN directly, we need a large-scale image data such as the ILSVRC dataset which contains more than one million images. If a large-scale training data is always needed, applicable problems of a DCNN is very limited. To avoid such situation and to make a DCNN effective even for small-scale data, it was proposed to use a pre-trained DCNN with a large-scale dataset such as the ILSVRC dataset as a feature vector extractor for a small-scale data. By extracting activation signals from the intermediate layer of the DCNN after an image is provided into the first layer of the pre-trained DCNN and its signals are propagated into the upper layers, the extracted signal can be regarded as image features. This DCNN features are commonly extracted from the output signals of the previous layer of the last one in the pre-trained DCNN. Donahue et al. [3] confirmed the effectiveness of DCNN features. Then, we use DCNN features for visual representation of Twitter photos. In addition, Babenko et al. [4] showed DCNN features was effective for image retrieval, and PCA-based compression did not harm the performance.

III. Proposed Method

A. Overview

In this work, we estimate locations for Twitter photos in the following steps.

1) Location estimation by visual features.
2) Location estimation by Twitter messages texts.

3) Integration of the locations estimated by the two kinds of features.

We estimation locations by visual and textual features independently, and then we integrate two results to obtain final results. When integrating, we use a reliable score of location estimated by visual features as a combining weight.

B. Grid-based location estimation

In this work, we estimate grid-based location rather than a pair of longitude and latitude. Figure 1 shows an example of location grids. We evaluate possible grids by giving scores, and select the grid with the best score as the final estimated location. To explain it more concretely, we estimate the probability that an image I is taken at the location grid L_i, $P(L_i|I)$, for each grid. This probability is estimated by integrating the visual-feature-based location probability, $P_v(L_i|I)$, and the textual-feature-based location probability, $P_t(L_i|I)$.

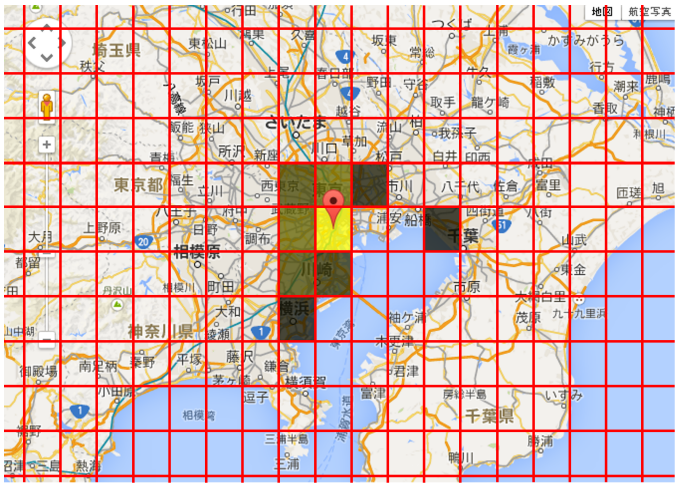

Figure 1. Location grid which are units of location estimation in this work.

C. Twitter photo localization by visual features

Following IM2GPS [2], we estimation photo locations with image retrieval for a large-scale geotagged image database. To build the database, we use several millions of geotagged photo tweets.

As visual features for image retrieval, we use and compare two methods. One is a local-feature-based method, while the other is DCNN-feature-based image search.

For a local-feature-based method, we use SIFT [5] as a local feature, and Vocabulary Tree proposed by Nister et al. [6] as an index for fast search.

For a DCNN-based method, to extract DCNN features [3], we use Overfeat [2] which extracts 4096-d activation features from the previous layer of the last layer, and compress it to 64-d vector with PCA following Babenko et al. [4].

To estimate locations of photos by image retrieval for a geotagged image database, we retrieve the top M similar images for a given image. The visual-feature-based score of i-th grid for a given image I is calculated as follows:

$$S_v(L_i|I) = \sum_{j=1}^{M} \frac{1}{\sqrt{j}} \phi(E_j - i) \tag{1}$$

where $\phi(x) = 1(x = 0), 0(x \neq 0)$, and E_j represents the location grid index of j-th retrieved images. The visual-feature-based location probability, $P_v(L_i|I)$, is a normalized value of $S_v(L_i|I)$.

$$P_v(L_i|I) = \frac{S_v(L_i|I)}{\sum_i S_v(L_i|I)} \tag{2}$$

D. Text-based location estimation

To estimate tweet location, it is possible to use Twitter geotagged text database like Watanabe et al. [1]. For simplicity, in our current work, we use off-the-shelf text-based location estimator, GeoNLP [3]. GeoNLP extracts place names such as Tokyo and New York and location-dependent named entity such as Tokyo Disneyland, and estimate locations based on the dictionary of association between location names and longitude/latitude. GeoNLP outputs multiple possible locations on a given text. Then, in the same as the case of visual-feature-based location estimation, we estimate the textual-feature-based location probability, $P_t(L_i|I)$.

Given N estimated locations, the textual-feature-based score of i-th grid for a given image I is calculated as follows:

$$S_t(L_i|I) = \sum_{j=1}^{N} \phi(E_j - i) \tag{3}$$

where $\phi(x) = 1(x = 0), 0(x \neq 0)$, and E_j represents the location grid index of j-th estimated location. Note that no ranking weighting is for the text-feature-based score, being different from the visual-feature-based score. The textual-feature-based location probability, $P_t(L_i|I)$, is a normalized value of $S_t(L_i|I)$.

$$P_t(L_i|I) = \frac{S_t(L_i|I)}{\sum_i S_t(L_i|I)} \tag{4}$$

E. Integration of estimated location

We define $P_t(L_i|I)$ and $P_v(L_i|I)$ as textual and visual scores of grid L_i with respect to image I. An integrated score $P(L_i|I)$ is calculated in the following equation:

$$P(L_i|I) = \frac{w_v P_v(L_i|I) + w_t P_t(L_i|I)}{\sum_{k=1}^{n} w_v P_v(L_k|I) + w_t P_t(L_k|I)} \tag{5}$$

[2]http://cilvr.nyu.edu/doku.php?id=software:overfeat:start

[3]http://agora.ex.nii.ac.jp/GeoNLP/

where w_t and w_v are weights of texts and image features, respectively. Note that the sum of both weights is 1.

Finally the grid with the largest $P(L_i|I)$ is decided as the final location of the given image I.

F. Automatic weight estimation

In the experiments, we estimated w_t and w_v automatically in addition to use a fixed set of the weights. We assume that if the estimated locations among top-N images by visual-features are concentrated to a certain grid, the estimation is reliable. We calculate a reliable score $B(I)$ which represents how extent the estimated locations to image I concentrate to one grid as follows:

$$B(I) = \frac{e^{\frac{K}{N}} - 1}{e - 1} \tag{6}$$

$$w_v = B(I), w_t = 1 - B(I), \tag{7}$$

where K represents the number of the estimations in the grid which gathers the largest number of votes, and N represents the number of the all location candidates estimated by visual features.

IV. Experiments

A. Data and evaluation

In the experiments, as shown in Table I, we used about geotagged photo tweets we collected from 2011 to 2014 via Twitter Streaming API. Note that we collected only tweets written in Japanese, because GeoNLP can treat only Japanese texts. As a training data stored in a geotagged photo database, we used 2.4 million images, while 4000 images are used for evaluation of location estimation methods as test data.

In the experiments, we set 50 as M.

Table I
THE STATISTICS OF THE EXPERIMENTAL DATA.

	Training	Test
Number	240million	4000
Usage of Texts	-	Yes
Usage of Geotags	Yes	-

For evaluation, we calculate the ratio of correctly-estimated images to all the test images. We regard the location of the given image is correctly estimated if the difference between the ground-truth location and the estimated location of the image is less than the certain distance.

Note that we use a spherical distance between a ground-truth location j and a estimated location of image i in the following equation:

$$D_{i,j} = \cos^{-1}\big(\sin(lat_i)\sin(lat_j) + \cos(lat_i)\cos(lat_j)\cos(long_i - long_j)\big) \tag{8}$$

B. Experimental results

Table II shows the results by BoF and DCNN in case of $M = 50$. We examined the location estimation accuracy by varying the weights from 0.0 to 1.0 with 0.25 step (A..E) within the distance, 5km, 10km, 50km and 100km. In case of A, only text-based estimation is used, while only visual estimation is used in case of E. In case of B, C, and D, both estimations are mixed according to the weights. As results, DCNN showed the better results than bag-of-features (BoF). Then, we applied automatic weight estimation to DCNN.

Table II
LOCALIZATION ACCURACY (%) WITH $M = 50$.

	feature	w_t	w_v	5km	10km	50km	100km
A	BoF	1.00	0.00	36.0 (1440)	57.2 (2288)	65.9 (2636)	68.3 (2732)
B		0.75	0.25	35.8 (1432)	57.5 (2300)	67.3 (2692)	69.7 (2788)
C		0.50	0.50	35.3 (1412)	56.8 (2272)	66.6 (2664)	68.8 (2752)
D		0.25	0.75	31.8 (1272)	50.6 (2024)	58.6 (2344)	60.5 (2420)
E		0.00	1.00	2.6 (104)	6.0 (240)	13.7 (548)	16.0 (640)
A	DCNN	1.00	0.00	36.0 (1440)	57.2 (2288)	65.9 (2636)	68.3 (2732)
B		0.75	0.25	36.7 (1468)	58.7 (2348)	67.2 (2688)	69.9 (2796)
C		0.50	0.50	36.6 (1464)	58.3 (2332)	66.6 (2664)	69.4 (2776)
D		0.25	0.75	35.0 (1400)	55.2 (2208)	62.8 (2512)	65.4 (2616)
E		0.00	1.00	4.1 (164)	8.1 (324)	15.9 (636)	18.0 (720)
F		AUTO	AUTO	36.3 (1452)	59.1 (2364)	68.9 (2756)	71.4 (2856)

For BoF, 'A' achieved the best within 5km, which meant no visual features helped improve accuracy. On the other hand, for BoF with the other distance than 5km and DCNN with all the distances, 'B' achieved the best, which meant visual features helped raise accuracy.

Automatic method, 'F', achieved the best among all the combination except for 5km, which showed the effectiveness of the propose method.

C. Discussions

From the results, we confirmed that integration of textual features with visual features improved the accuracy, although the estimation accuracy by only visual features is lower than only textual features. In addition, we also confirmed that DCNN features outperformed conventional BoF based features.

Figure 3 and Figure 4 shows examples of successful cases and failure cases, respectively. In all the cases, we used

only visual features. As shown in Figure 3, landmarks and characteristics landscape tend to bring successful location estimation, while indoor scene, foods and sky which are expected to independent of location tend to be estimated as being in Tokyo area where the number of geotagged Japanese tweets are the largest. as shown in Figure 4.

Although location estimation of the tweet photos shown in Figure 4 failed using only visual features, textual features successfully estimated their locations because the tweet message included some nouns related to the true locations. When using automatic weight estimation, w_t became around 0.8, and w_v became around 0.2 for both the images. As results, the effects of failure estimation by visual features were reduced greatly. Instead, nearly true locations were estimated mainly thanks to estimation by textual features.

Figure 2 shows the relation between the automatically estimated weights and the errors of the locations estimated by visual features, which confirmed that the estimated weights reflected the reliability of the estimated locations by visual features almost correctly. These facts mean that integration by automatic weight estimation was effective for Twitter photo geo-localization.

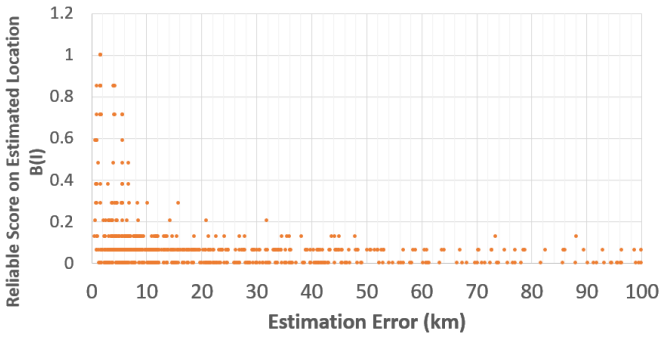

Figure 2. Relation between the automatically estimated weights and the errors of the locations estimated by visual features.

V. Conclusions

In this paper, we proposed a method to localize Twitter photos which integrate textual information of Twitter messages and visual information extracted from Twitter photos. The experimental results shows integration of both features improved localization accuracy compared to using only single modality.

For future work, we like to integrate our data source related to geotagged photos such as Flickr and Panoramio.

References

[1] K. Watanabe, M. Ochi, M. Okabe, and R. Onai, "Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs," in *Proc. of the 20th ACM international conference on Information and knowledge management*, pp. 2541–2544, 2011.

[2] J. Hays and A. A. Efros, "IM2GPS: estimating geographic information from a single image.," in *Proc.of IEEE Computer Vision and Pattern Recognition*, 2008.

[3] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," 2014.

[4] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *Proc.of European Conference on Computer Vision*, 2014.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[6] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc.of IEEE Computer Vision and Pattern Recognition*, 2006.

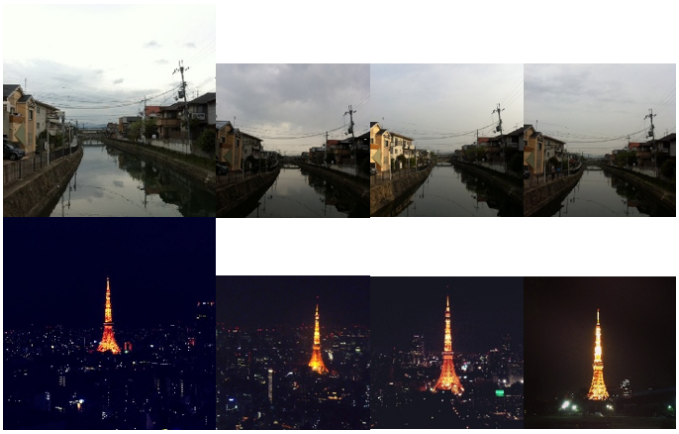

Figure 3. Examples of successful cases (Right: test photos, Left: training photos): (Upper) Kurozumi Station in Fushimi, Kyoto (Error: 1.76km), (Lower) Tokyo Tower in Minato, Tokyo (Error: 1.70km).

Figure 4. Examples of failure cases (Right: test photos, Left: training photos): (Upper) Sendai Station in Sendai, Miyagi (Error: 315km (Estimation: Tokyo)), (Lower) Nara Station in Nara (Error: 373km (Estimation: Tokyo)).