Photo Set Refinement and Tag Segmentation in Georeferencing Flickr Photos

Jiewei Cao
South China University of Technology
GuangZhou, China
jonbakerfish@gmail.com

ABSTRACT
In this paper, we describe our approach as part of the MediaEval 2013 Placing Task evaluation. We use language model and similarity search as baseline approach, and improve the accuracy by two techniques: photo set refinement and tag segmentation. The first technique takes advantage of geo-location correlation among test photos and the second one exploits the textual similarity between tags.

1. INTRODUCTION
The MediaEval 2013 Placing Task requires participants to assign geographical coordinates (latitude and longitude) to each provided test image, we refer to [1] for a detailed description. A framework proposed by [2] is used as our baseline approach. The main contributions of this paper are two techniques to improve the accuracy of georeferencing. Firstly, we noticed that Flickr users can organize their photos by assigning them to different sets and collections[1]. Intuitively, photos in the same set are highly correlated, and we can exploit these relations when estimating the geo-location of given images. The outcome of our submitted runs justifies this assumption. Secondly, when only training data provided by the task organizers can be used, the unseen tags - tags only existing in test data - are useless for geo-referencing. However, we tried to exploit these tags by applying tag segmentation. This is similar to the word segmentation pre-processing for language that written without spaces between words, such as Chinese. Both proposed techniques can be applied to other existing systems with little changes.

2. METHODOLOGY
2.1 Data Pre-processing
A total of 8,539,050 geo-referenced photos from Flickr were provided as training data. Following [2], we carried out two preliminary filter steps on this training set. First, photos without tags are removed. Second, we removed the duplicated photos in a slightly different approach: photos uploaded by the same user, and with an identical tag set, and the Haversine distance among these photos is less than $\tau = 10m$ are treated as duplicates and only one instance is retained. Here we use a distance threshold τ instead of identical latitude and longitude in order to relax the restriction of filtering, and we can remove more or less duplicates according to the τ we selected. Smaller distance threshold means more photos with identical tag set and different location can be retained, and identical geo-location is a special case when $\tau = 0m$. Finally, this resulted in a pre-processed training set with 4,538,784 photos when the $\tau = 10m$. There are five different test sets and we chose test3 whose size is 53,000. We didn't use any external resource for georeferencing except run 5, in which we geocoded the home location of users in the test set, using the Google Geocoding API[2].

2.2 Baseline Approach
The framework proposed by [2] applies a two steps approach to estimate the location of test photos. First, the location of the training data are clustered into 500, 2500 and 10000 clusters which could be referred to as C_{500}, C_{2500} and C_{10000}. Given a clustering, a Naïve Bayes classifier is used to find the most likely cluster to contain the location of a given test photo. Second, within the found cluster, they use a similarity search to find the training items whose tags are the closest to the ones of test photo. In [3], they proposed an improved spatially aware feature ranking method which is based on Ripley's K statistic. Therefore, we use this framework with Ripley's K feature selection as our baseline approach.

2.3 Photo Set Refinement
Photos within the same set or collection would be highly geo-location correlated. For example, a user can upload his photos which were taken on during a trip into a new set created by him. However, not every photo in the same set is well tagged because a user only tags the photos he loved or interested in, and leaving others un-tagged or poorly tagged. This will result in photos with completely different tag sets or visual content could be considered as taken in the same location or nearby, if they were within the same photo set.

A test photo with poor tags will result in a bad estimation. However, if this photo belongs to a photo set which contains one or more photos with well estimated location (usually well tagged), then we can use the centroid location of these photos as the estimation for the bad one. This is the intuition of our proposed photo set refinement, and there are two problems here: 1. Given a photo, how to find its neighbors within the same photo sets? 2. How to distinguish between the well estimated photo and bad one? Although we didn't handle the Placeability sub-task of Placing Task at MediaEval 2013, our solution for the second problem may be considered as a naive approach for error estimation.

To handle the first problem, it seems we can simply break down the test data into different sets according to the original photo sets created by users. However, a photo set in this user scenario can be

MediaEval 2013 Workshop, October 18-19, 2013, Barcelona, Spain

[1] http://www.flickr.com/help/collections/

[2] https://developers.google.com/maps/documentation/geocoding/

changed from time to time, whether it's adding new photos or deleting the old ones. And the geo-location correlation between these photos will become weaker. Therefore we need a different approach: Given a photo, we find its neighbors in the test data by comparing their user id, the timestamp of the photo was taken on and uploaded. If a photo has an identical user id with the given photo, and the time interval between their taken dates is less than $\tau_{taken} = 7\ days$, and their uploaded dates interval is less than $\tau_{uploaded} = 7\ days$, then we consider these two photos belong to the same photo set. Here both thresholds (τ_{taken} and $\tau_{uploaded}$) are set to 7 days because we consider a week-long vacation is common for most people, and photos taken and uploaded during these days can be consider as a photo set.

There are three clusterings of the training data, namely C_{500}, C_{2500} and C_{10000}, and a given test photo P_i can be classified to three different medoids respectively, which we referred to as M^i_{500}, M^i_{2500} and M^i_{10000}. Intuitively, these three medoids are not far from each other if P_i is well estimated and vice versa. So given a photo set $S = \{ P_i | i = 1 \dots N \}$, we consider P_i as well estimated if all the Haversine distances among M^i_{500}, M^i_{2500} and M^i_{10000} are less than 1000km, otherwise P_i is marked as badly estimated. Finally, we use the centroid location of well estimated photos as the finial estimation for the poorly estimated ones, and if no well estimated photo is found, we use the home location of the user (in run 5 only) or simply leave it unchanged.

2.4 Tag Segmentation

Consider the tag T_a = 'southchinauniversityoftechnology' and tag T_b = 'southchinauniversityoftechnologylibrary'. If T_b was an unseen tag, it will be ignored even though we can assume that T_a and T_b are correlated because of their textual similarity. However, we can split T_b into two terms 'southchinauniversityoftechnology' and 'library', then the first term is identical to T_a and can be used for georeferencing. Our approach for tag segmentation is to model the distribution of the segmentation output. First, we assume all tags are independently distributed, and the relative frequency of all tags in the training data was calculated. We created a tag dictionary sorted in descending order with size 2,080,618. We also assume that the tags in the training data follow Zipf's law [4], which means that the tag with rank n has probability $\frac{1}{n \log N}$, where N is the number of tags in the dictionary. Then we use dynamic programming to infer the position of the cut point. The most likely segmentation is the one that maximizes the product of the probability of each individual split term. Instead of directly using the tag probability, we use a cost defined as the logarithm of the inverse of the probability to avoid overflows.

Given a test photo, all the tags in this photo are preprocessed by tag segmentation before georeferencing. For each tag, we select its longest split term and assign it to this photo as a new tag. The remaining terms (such as 'library') are discarded because these terms are usually not spatially relevant.

3. RESULTS AND DISCUSSION

We submitted five runs and the results of our experiments are shown in Table 1.

run1: is the baseline approach

run2: uses visual features only and K-nearest neighbor search.

run3: corrects poorly estimated photos in run1 by photo set refinement proposed in section 2.3.

Table 1: Percentage of correctly detected locations and median error of each run in kilometer.

	1 km	*10 km*	*100 km*	*500 km*	*1000 km*	*ME km*
run1	20.7	43.0	55.3	62.8	66.3	37.65831
run2	0.0	0.0	0.0	0.1	0.6	10026.17
run3	21.1	44.2	57.1	65.2	69.2	28.01581
run4	21.2	44.2	57.5	65.5	69.6	27.0791
run5	20.9	46.1	61.7	71.8	76.5	16.73021

run4: is similar to run3 but tag segmentation is used to preprocess the test data before georeferencing.

run5: uses the user home location in the photo set refinement step. Note that this location is also used when estimating the prior probability in language model framework, we refer to [2] for more details.

The result of run3 justifies our assumption and we can estimate test photos jointly to improve the accuracy. In our experiment, the number of different estimated photos between run1 and run3 is 4,963, and this is the number of photos changed during the photo set refinement step. After comparing the georeferencing result of run1 and run3 with the ground truth, among these 4,963 photos, we found that 4,390 photos' estimated location in run3 became closer to the real location in comparison with run1, and the rest of 573 photos had a larger error distance in run3 compared with run1. This is mainly caused by the incorrectness of differentiating well estimated photo and the bad one. For some well estimated photos, the Haversine distances among their M^i_{500}, M^i_{2500} and M^i_{10000} could be far from each other. Therefore, we need a much more robust way to find out the error estimation.

Run4 doesn't show a promising improvement compared with run3. The reason is that unseen tags are not always segmentable, but the proposed technique did improve the performance slightly and the extra time and computational costs are low. However, other than tag segmentation which only exploits the textual similarity between unseen tags and training tags, we can also try to find out the semantic similarity between them by utilizing external resource or machine learning technique.

Run5 indicates that the home location of the user is very important for georeferencing for most photos, which is consistent with previous research findings. In run2, we simply used the extracted visual features provided by task organizers and ran a K-nearest neighbor search to find the most similar photo in the training set. However, we didn't get a reasonably geo-location prediction and more intensive study is needed in our future work.

4. REFERENCES

[1] C. Hauff and B. Thomee and M. Trevisiol. Working Notes for the Placing Task at MediaEval 2013. In *MediaEval 2013 Workshop*, 18-19 October 2013, Barcelona, Spain.

[2] O. Van Laere, S. Schockaert, and B. Dhoedt. Georeferencing Flickr resources based on textual meta-data. *Information Sciences*, 2013, http://dx.doi.org/10.1016/j.ins.2013.02.045.

[3] O. Van Laere, J. Quinn, S. Schockaert, B. Dhoedt. Spatially-Aware Term Selection for Geotagging. *IEEE TKDE* 2013. http://doi.ieeecomputersociety.org/10.1109/TKDE.2013.42

[4] G. K. Zipf. Human Behaviour and the Principle of Least-Effort. *Addison-Wesley*, Cambridge MA, 1949