How INRIA/IRISA identifies Geographic Location of Videos

Michele Trevisiol* trevi@yahoo-inc.com
Jonathan Delhumeau † jonathan.delhumeau@inria.fr
Hervé Jégou † herve.jegou@inria.fr
Guillaume Gravier ‡ guillaume.gravier@irisa.fr

*Web Research Group
Universitat Pompeu Fabra
Barcelona, Spain

†INRIA
Rennes, France

‡CNRS/IRISA
Rennes, France

ABSTRACT

In this paper, we describe our approaches and their results as part of the MediaEval 2012 Placing Task evaluation. We present two different tag-based techniques. Both first preselect one or several geographic area of interest and then perform a deeper analysis inside the selected area(s) to return the coordinates more likely to be related with the input tags. In addition, we also implement a content-based method that uses aggregated local images descriptors (VLAD) to find the video's visual nearest neighbors and infer its coordinates. In this work we do not use gazetteers or any other external information.

1. INTRODUCTION

The goal of the Placing Task of MediaEval 2012 Benchmark[1] is to retrieve as accurately as possible the location, in terms of latitude and longitude, of a set of Flickr videos. We can exploit information such as visual content, metadata, social information, users' social graph and users' previously uploaded content. For this purpose, a large training-set of geo-annotated Flickr images and videos is available.

2. METHODOLOGY

Our approach uses all the available information, however, our method relies mostly on tags analysis. Our idea is to identify tags that are geographically descriptive, discarding the ones likely to be irrelevant. First, we apply a preprocessing step in order to filter out the noise commonly present in the tags. Then, two different techniques were implemented: one based on simple text matching, and one based on a weighting scheme. We apply different methods in cascade: if a method does not find predictions, the following may be able to do it. *a*) system tags technique (Sec. 2.2 and 2.3), *b*) user's upload history, social information or home town (Sec. 2.4), *c*) content-based matching (Sec. 2.5), *d*) prior-location (Sec. 2.4).

2.1 Tags Pre-Processing

In order to work with a clean set of tags (Tc_{train}), we perform a few pre-processing steps. First standard steps such as removal of spaces, accents, *etc.* We also discard all the numeric tags (almost never relevant) and remove numeric characters from the alphanumeric tags. Then we built a basic stop-word dictionary containing common words (*e.g.* travel, birthday, cat, geotag) and names about the device with which the photos were taken (*e.g.* camera, iPhone, Canon). Some videos are annotated with *machine tags* (mtags)[1], *i.e.* one or more tags that Flickr recognized as a location (usually a country name, sometimes also the city name). We extract all of them in order to exploit their information separately in our tags techniques.

2.2 Basic Method (1st run)

For the first method, we filter the tags based on their geographic spread. For each tag t_i we compute the term frequency tf_{t_i} among all the tags, and the average Haversine distance $avgD_{t_i}$ among all the coordinates of the video/image in which it appears. We filter out all the tags that do not respect the following condition:

$$\forall t_i \in T_{train}, \quad t_i \in Tc_{train} \iff \begin{cases} tf_{t_i} \leq 50, \\ avgD_{t_i} \geq 200. \end{cases}$$

We consider the problem as a classic Word-Document task in information retrieval. We consider each training datum, image or video, as a geo-annotated document made up with tags instead that words (in this step we are considering only training data with tags). We merge all the documents associated with the same set of tags, collecting all the associated coordinates. This way we obtain for each document a list of coordinates with frequency. A test-set video is represented by $V_i = \langle \{m_i\}, \{t_i\}, u_i \rangle$ where $\{m_i\}$ is the set of mtags, $\{t_i\}$ is the set of tags and u_i the user id. If the video contains mtags we retrieve all the documents where there is at least one common mtag, otherwise we do the same with tags. Then, we count how many time individual (m)tags appear in each document. Finally, we obtain a ranked list of documents, each of them with a list of candidate coordinates. We select the document with the highest score (if there is more than one with this score, we pick all of them), and we choose the medoid of all the associated coordinates (*i.e.* the coordinates minimising the average distance to all the others).

2.3 Weighted Method (2nd run)

This method directly analyses the relationship between tags and coordinates, taking into account the weight of each tag. The idea is to first determine the approximate area in which the video is likely to belong, then to find the most

*Work done while visiting PhD student at INRIA Rennes

MediaEval 2012 Workshop, October 4-5, 2011, Pisa, Italy

[1] http://www.flickr.com/groups/api/discuss/72157594497877875/

probable coordinates from the known locations in that area. To do that, first we define the areas and divide up the training set by quantifying the coordinates on a *square* grid of 0.1°. We then compute a weighted co-occurrence matrix associating each tag with the areas it appears in (allowing us to select the most likely areas). For each area we also compute another weighted matrix associating the tags with coordinates (allowing us to select the most likely coordinates).

The initial tag weights are determined using a method similar to the one in Section 2.2, but instead of discarding tags we assign them one out of four levels of importance. We manually defined the following thresholds identifying how *geo-descriptive* a tag is. $\forall t_i \in T_{train}$:

$$w_{t_i} = \begin{cases} -1 & \text{if } tf_{t_i} > 100K \text{ or } avgD_{t_i} < 0.2 \\ 10 & \text{if } tf_{t_i} \geq 200 \text{ and } 10 \leq avgD_{t_i} \leq 50 \\ 5 & \text{if } tf_{t_i} \geq 150 \text{ and } avgD_{t_i} \leq 70 \\ 1 & \text{otherwise} \end{cases}$$

To improve performance, we then re-weighted the tags with a feature weighting algorithm. After testing both TF-IDF and BM25, we chose the latter which performed better (confirming the results of Whissell *et al.*[6]). Those new weights are smoothed using signed SQRT (generalizing to textual features the results of Jégou and Chum[3] on visual features). Finally, we normalize the weights with L2-norm and apply a whitening step in order to make the data less redundant (also described in [3]).

Each query vector is first multiplied by the weighted tag-area matrix to find the most probable area (if several have the same score, we keep them all). Then the query vector is multiplied by the weighted tag-coordinates matrix for this area to find the most probable location. In case of tied scores at this point, we choose the medoid.

2.4 User-Based Location Estimation

For some test video no relevant tags were associated. To handle these cases we implemented different solutions. For each user with images/videos in the training-set, we pick a pre-computed *user location*, *i.e.* the most frequent location for his content (or the medoid of the most frequent ones). For the others, we go through all of his social connections to find the *user locations* of all of his contacts and choose the medoid. If this information is not available either, we use his home town. If all else fails, we choose the *prior location* which is the medoid of all the coordinates in T_{train}.

2.5 Content-Based Method (4th run)

This method is based solely on the visual content. For this purpose, and for each the provided keyframes, we first compute our own descriptors, selected for their good results for image retrieval. Those descriptors are based on normalised and whitened (local) SIFT descriptors[2] aggregated into a (global) VLAD descriptor[5], which are further dimensionally reduced, whitened and normalised[3]. The descriptors for each of the keyframes in the training videos are indexed using Product Quantization[4]. We then do an approximate nearest neighbours search for all the test video keyframes. From this frame-to-frame list of nearest neighbours we deduce a video-to-video list of potential matches by keeping only the best scores for each video. Finally, from this list of candidate videos we get the corresponding list of coordinates and keep the medoid.

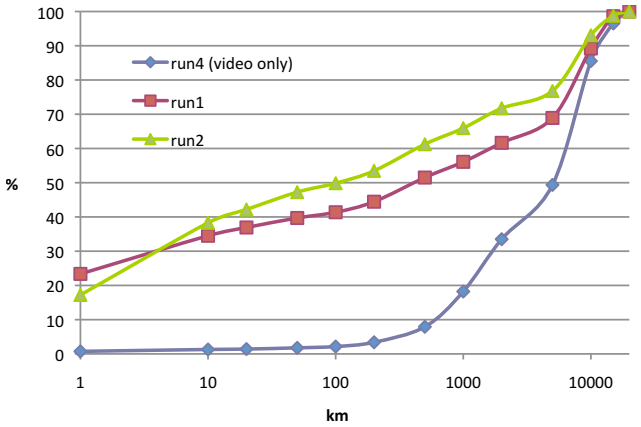

Figure 1: Cumulative correctly detected locations: rate of video founds in a radius of x km.

radius(km)	run1(%)	run2(%)	run4(%)
1	**23.36**	17.22	0.74
10	34.51	**38.28**	1.29
100	41.37	**49.86**	2.15
1000	56.10	**65.95**	18.22
10000	89.22	**93.07**	85.53

Table 1: Percentages of correctly detected locations.

3. EXPERIMENTS

The results of our experiments are shown in Figure 1 and in Table 1. The algorithm described in Section 2.3 is outperforming all the others in all the evaluation radius except for the first kilometer. Indeed, for the smaller radius the most accurate method is the one presented in Section 2.2.

4. CONCLUSIONS AND FUTURE WORK

One reason for the bad start of the method in *run2*, may be due to the coordinates partitioning method, as selecting the wrong area in the first step makes it hard to find a coordinates closer than 1km. Therefore performance in the smaller radius might be improved by partitionnig the coordinates by a clustering technique like K-Means. Integrating a gazetteer might also help improve overall performance.

5. ACKNOWLEDGMENTS

This work was partially funded by OSEO, French state agency for innovation, in the framework of the Quaero project and by Grant TIN2009-14560-C03-01 of the Ministry of Science and Innovation of Spain. Furthermore, we would like to thank Vincent Claveau for his helpful suggestions.

6. REFERENCES

[1] R. Adam and P. Kelm. Working notes for the placing task at mediaeval 2012, 2012.

[2] M. Jain, R. Benmokhtar, P. Gros, and H. Jégou. Hamming Embedding Similarity-based Image Classification. In *ICMR*, June 2012.

[3] H. Jégou and O. Chum. Negative evidences and co-occurrences in image retrieval: the benefit of PCA and whitening. In *ECCV*, Oct. 2012.

[4] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *PAMI*, 33(1):117–128, Jan 2011.

[5] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *PAMI*, Sept. 2012.

[6] J. Whissell and C. Clarke. Improving document clustering using okapi bm25 feature weighting. *IR*, 14:466–487, 2011.