Learning Tag Relevance by Neighbor Voting for Social Image Retrieval

Xirong Li, Cees G.M. Snoek, and Marcel Worring
ISLA, Informatics Institute, University of Amsterdam
Kruislaan 403, 1098 SJ, Amsterdam, The Netherlands
{x.li, cgmsnoek, m.worring}@uva.nl

ABSTRACT

Social image retrieval is important for exploiting the increasing amounts of amateur-tagged multimedia such as Flickr images. Since amateur tagging is known to be uncontrolled, ambiguous, and personalized, a fundamental problem is how to reliably interpret the relevance of a tag with respect to the visual content it is describing. Intuitively, if different persons label similar images using the same tags, these tags are likely to reflect objective aspects of the visual content. Starting from this intuition, we propose a novel algorithm that scalably and reliably learns tag relevance by accumulating votes from visually similar neighbors. Further, treated as tag frequency, learned tag relevance is seamlessly embedded into current tag-based social image retrieval paradigms.

Preliminary experiments on one million Flickr images demonstrate the potential of the proposed algorithm. Overall comparisons for both single-word queries and multiple-word queries show substantial improvement over the baseline by learning and using tag relevance. Specifically, compared with the baseline using the original tags, on average, retrieval using improved tags increases mean average precision by 24%, from 0.54 to 0.67. Moreover, simulated experiments indicate that performance can be improved further by scaling up the amount of images used in the proposed neighbor voting algorithm.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.2.4 [**Database Management**]: Systems—*Multimedia databases*

General Terms

Algorithms, Design, Experimentation

Keywords

Social image retrieval, Tag relevance, Neighbor voting

MIR'08, October 30–31, 2008, Vancouver, British Columbia, Canada.

1. INTRODUCTION

Social multimedia sharing systems have successfully motivated amateur users around the world to tag and share their content on the web. A good example is Flickr which hosts more than two billion images, and receives around three million new uploaded photos per day [1]. Apart from its usage for general purpose retrieval, this rich multimedia database is triggering many innovative research scenarios in area as diverse as personalized multimedia retrieval [24], landmark recognition [16], visual query construction [30], and automatic image labeling [29]. For all these scenarios, one would expect tag-based retrieval approaches to be a good starting point for search.

Despite the success of amateur tagging, tags are known to be ambiguous, limited in terms of completeness, and overly personalized [10, 21]. This is not surprising because of the uncontrolled nature of social tagging and the diversity of knowledge and cultural background of its users. Note that the relevance of a tag given the visual content is subjective. Relevance is indeed a relationship between an image and a user. Nonetheless, to find images relevant to a majority of users, an objective criterion of tag relevance is required. We define a tag and an image as relevant if the tag accurately describes objective aspects of the visual content, meaning the content can be easily and consistently recognized by common knowledge. Consider the examples in Figure 1. When we seek an airplane object and submit an "airplane" query to Flickr, the result typically includes inside scenes like persons in seats or aerial views from airplane windows depicting clouds and grounds. The airplane tag for Figure 1-c,d is subjective because the airplane concept is not easily and consistently captured by common users. Apart from the fact that tags can be subjective, individual tags are mostly used once per image. This implies that within an image, relevant tags and irrelevant ones are not distinguishable by their occurrence frequency. Hence, given the fact that tags are ambiguous, noisy, and limited, a fundamental problem in social image retrieval is how to reliably learn the relevance of a tag with respect to the visual content it is describing.

Existing methods to automatically predict tag relevance with respect to the visual content often heavily rely on complicated machine learning algorithms [2, 4, 17]. In general, the methods boil down to learning a mapping between low-level visual features and high-level semantic concepts. Compared to a potentially unlimited vocabulary existing in social tagging, currently only a very limited number of visual concepts can be effectively modeled using small-scale datasets. Moreover, uncontrolled visual content contributed by ama-

Figure 1: Social tagging examples. Images (a) and (b) are objectively tagged as "airplane", while images (c) and (d) are subjectively tagged as "airplane".

teurs creates a broad domain environment having significant diversity in visual appearance even for the same concept [25]. The scarcity of training examples and the significant diversity of visual appearance might make the learned models unreliable and hardly generalizable.

Intuitively, if different persons label visually similar images using the same tags, these tags are likely to reflect objective aspects of the visual content. The intuition implies that the relevance of a tag with respect to an image might be inferred from tagging behavior of visual neighbors of that image. Starting from this intuition, we propose a novel neighbor voting algorithm for learning tag relevance. The key idea is, by propagating common tags through visual links induced by visual similarity, each tag accumulates its relevance credit by receiving neighbor votes. To demonstrate the viability of the proposed algorithm, we provide a systematic evaluation on one million Flickr images.

The rest of the paper is organized as follows. Section 2 reviews related work. Learning tag relevance is described in Section 3. The experimental setup is in Section 4, followed by results in Section 5. The paper is concluded in Section 6.

2. RELATED WORK

Many methods have been developed to improve multimedia search where textual descriptions of visual content are vague, e.g., web images [8, 23] and news videos [31, 11]. These methods might also be used to improve social image retrieval. We divide them according to their query-dependence into two types of approaches, i.e. query-dependent and query-independent approaches.

2.1 Query-dependent Approaches

Given a user query, query-dependent approaches try to improve search results either by reranking search results using pseudo relevance feedback algorithms [31, 23, 8, 11] or by expanding the original query [20, 5, 3].

Reranking methods assume that the majority of search results are relevant to the query and relevant examples tend to have similar visual patterns such as color and texture. To find the dominant visual patterns, density estimation methods are often used, typically in the form of clustering [23, 11]. However, density estimation is known to be inaccurate when feature dimensionality is high and samples are insufficient, which is mostly the case in the reranking scenario. Besides, density estimation is computationally expensive. In [11] for example, the authors report an execution time of 18 seconds per search round, while a study on web users [22] shows the tolerable waiting time for web information retrieval is approximately 2 seconds. The difficulty in density estimation and the associated computational expense put the utility of reranking methods for social image retrieval into question.

Query expansion methods augment the original query by adding relevant terms [20, 5, 3]. The methods are either lexicon-driven or corpus-driven. In [5], for instance, the authors use synonyms from WordNet [7], whereas in [20] the authors select strongly related terms from snippets returned by Google web search. Another example is [3], where the authors rely on clustering techniques to find correlations between tags. Though the methods may recall more relevant results as more query terms are used, they leave the fundamental problem of noisy amateur tagging unaddressed.

2.2 Query-independent Approaches

The query-independent approaches improve tagging quality either by adding new annotations [2, 4, 17] or by removing existing noisy ones [14, 28]. Most of the approaches are model-based. That is, by treating tags as visual concepts, they first train concept classifiers for each tag and then use the learned classifiers to predict relevant image tags. Despite their success in small-scale image databases, however, such model-based approaches are not scalable to handle a massive amount of social-tagged images. The scarcity of learning examples and the significant diversity in visual appearance might make the learned classifiers unreliable and hardly generalizable. Furthermore, training such a large amount of classifiers is computationally prohibitive.

More recently, a new "model free" method is developed to learn visual concepts from web images, e.g. [29, 26]. The intuition is assuming there exists a well-annotated and unlimited-scale image database such that for any unlabeled image outside the database, we can find its visual duplicate. Then, annotating an unlabeled image can be done by first finding its duplicate from the database and then propagating tags from the duplicate to that image. In a more realistic case where the database is very large, e.g., images on the web, we can still find a set of visually similar images for a given image, and use tags of neighbor images to annotate the given image. However, due to the semantic gap problem [25], i.e., the inconsistency between visual similarity and semantic similarity, irrelevant tags may be also propagated.

Note the query-independent approaches optimize the image annotation problem, an intermediate step for retrieval. They might be suboptimal for the retrieval goal. Besides, to search the massive amount of social-tagged images, an effective and scalable retrieval framework is crucial. Consequently, how to effectively and flexibly leverage learned tag relevance information within the retrieval framework is an important and practical requirement. However, this problem is overlooked or at least understudied in previous work.

2.3 Contribution of This Work

Our neighbor voting algorithm is query-independent. Different from other query-independent approaches that target at addding new annotations or removing noisy ones [17, 14], our method preserves original tags but estimates tag relevance by votes from visually similar neighbors. The neighbor voting framework does not impose any model training for any visual concepts. It is thus scalable to handle a large amount of social-tagged images. Our method shares similar spirits with the "model-free" method [29, 26], since both methods can be regarded as propagating tags between neighbor images. Nonetheless, our method has two distinguish-

able novelties. First, only common tags shared by neighbors are propagated. We do not introduce new tags to an image. Such self-validation mechanism reduces the risk of incorrectly propagating irrelevant tags, which is caused by the inconsistence between visual similarity and semantic similarity. Second, the neighbor voting process directly optimizes the retrieval problem. When the neighbor search is better than random sampling, learned tag relevance is a good ranking criterion. Finally, tag relevance, treated as tag frequency, is flexibly integrated into a scalable tag-based retrieval framework. The scalability, reliability, and flexibility of our method make it promising for real applications.

3. LEARNING TAG RELEVANCE

3.1 Problem Formulation

3.1.1 Tag relevance for image retrieval

We introduce some notations first. Given an image collection $\mathcal{I}$, a tag vocabulary $\mathcal{T}$, and a user set $\mathcal{U}$, we define three indicator functions to describe relationships between images, tags, and users. That is, for image $I \in \mathcal{I}$, tag $w \in \mathcal{T}$, and user $u \in \mathcal{U}$,

$$\begin{cases} s(I,w) = 1, \text{ if } I \text{ is tagged as } w, \ 0 \text{ otherwise} \\ r(I,w) = 1, \text{ if } I \text{ and } w \text{ are relevant}, \ 0 \text{ otherwise} \\ h(I,u) = 1, \text{ if } I \text{ is tagged by user } u, \ 0 \text{ otherwise} \end{cases}$$

Let $\mathcal{L}_w = \{I \in \mathcal{I} | s(I,w) = 1\}$ be images tagged as w, $\mathcal{R}_w = \{I \in \mathcal{I} | r(I,w) = 1\}$ images relevant to w, and $\bar{\mathcal{R}}_w = \{I \in \mathcal{I} | r(I,w) = 0\}$ images irrelevant to w. We further denote $P(w|\mathcal{R}_w)$ as probability of correct tagging, i.e., an image relevant to w tagged as w. Similarity, $P(w|\bar{\mathcal{R}}_w)$ is probability of incorrect tagging, i.e., an image irrelevant to w tagged as w. Since $|\bar{\mathcal{R}}_w| \gg |\mathcal{R}_w|$ in general, $P(w|\bar{\mathcal{R}}_w) < P(w|\mathcal{R}_w)$.

Given query tag w, its search result set $\mathcal{L}_w$ consists of two distinct subset. One is $\mathcal{L}_w \bigcap \mathcal{R}_w$, the image set relevant to the query. The other is $\mathcal{L}_w \bigcap \bar{\mathcal{R}}_w$, the image set irrelevant to the query. $\bigcap$ is the intersection operator on image sets. A good ranking function ranks images from $\mathcal{L}_w \bigcap \mathcal{R}_w$ ahead of those from $\mathcal{L}_w \bigcap \bar{\mathcal{R}}_w$. Clearly, if we can accurately estimate tag relevance with respect to the visual content, we solve the retrieval problem. Since individual tags are typically used once per image, estimating tag relevance by tag frequency, a well-founded principle in document retrieval [15], is unfeasible here.

3.1.2 Learning tag relevance from neighbors

Let f be a visual similarity function between two images. For each image $I \in \mathcal{I}$, we denote its k nearest neighbors found in $\mathcal{I}$ by f as $NN_f(I,k)$. We use $n_f(I,k,w)$ to represent the count of tag w in $NN_f(I,k)$, i.e.,

$$n_f(I,k,w) = |\{J \in \mathcal{I} | J \in NN_f(I,k), s(J,w) = 1\}| \quad (1)$$

where image J is a neighbor of image I. We argue that $n_f(I,k,w)$ is a good estimation of tag relevance.

For image $I \in \mathcal{L}_w$, its neighbor set $NN_f(I,k)$ is decomposed into two distinct subset. One subset consists of images from $\mathcal{R}_w$. The other consists of images from $\bar{\mathcal{R}}_w$. Specifically, for image $I' \in \mathcal{L}_w \bigcap \bar{\mathcal{R}}_w$, we assume that $NN_f(I',k)$ has β examples from $\mathcal{R}_w$ and $k-\beta$ examples from $\bar{\mathcal{R}}_w$, where $0 \le \beta \le k$. Similarly for image $I'' \in \mathcal{L}_w \bigcap \mathcal{R}_w$, $NN_f(I'',k)$ has $\beta+\varepsilon$ examples from $\mathcal{R}_w$ and $k-\beta-\varepsilon$ examples from $\bar{\mathcal{R}}_w$. ε is a variable indicating the quality of k-nn search using the similarity function f. If we use random sampling to find neighbors, $E[\varepsilon] = 0$. $E[\bullet]$ is the expectation operator. For a k-nn search using f,

$$E[\varepsilon] = \begin{cases} > 0, & \text{if it is better than random sampling} \\ = 0, & \text{if it is equal to random sampling} \\ < 0, & \text{otherwise} \end{cases}$$

Now for image $I' \in \mathcal{L}_w \bigcap \bar{\mathcal{R}}_w$, we have

$$E[n_f(I',k,w)] = E[\beta \cdot P(w|\mathcal{R}_w) + (k-\beta) \cdot P(w|\bar{\mathcal{R}}_w)] \quad (2)$$

For image $I'' \in \mathcal{L}_w \bigcap \mathcal{R}_w$, we have

$$\begin{aligned} & E[n_f(I'',k,w)] \\ = \ & E[(\beta+\varepsilon) \cdot P(w|\mathcal{R}_w) + (k-\beta-\varepsilon) \cdot P(w|\bar{\mathcal{R}}_w)] \\ = \ & E[\beta \cdot P(w|\mathcal{R}_w) + (k-\beta) \cdot P(w|\bar{\mathcal{R}}_w)] \\ & + E[\varepsilon](P(w|\mathcal{R}_w) - P(w|\bar{\mathcal{R}}_w)) \\ = \ & E[n_f(I',k,w)] + E[\varepsilon](P(w|\mathcal{R}_w) - P(w|\bar{\mathcal{R}}_w)) \end{aligned} \quad (3)$$

and

$$E[n_f(I'',k,w) - n_f(I',k,w)] = E[\varepsilon](P(w|\mathcal{R}_w) - P(w|\bar{\mathcal{R}}_w))$$

If $E[\varepsilon] > 0$, we get $E[n_f(I'',k,w) - n_f(I',k,w)] > 0$, meaning $\mathcal{L}_w \bigcap \mathcal{R}_w$ and $\mathcal{L}_w \bigcap \bar{\mathcal{R}}_w$ are expectedly distinguishable by $n_f(I,k,w)$. Therefore, when the k-nn search is better than random sampling, $n(I,k,w)$ is a good measure of the relevance of tag w with respect to image I.

3.2 A Neighbor Voting Algorithm

We have argued that learning tag relevance boils down to computing $n(I,k,w)$, i.e., the count of tag w in the k nearest neighbors of image I. Considering that each neighbor of I votes on w if itself is tagged as w, then $n(I,k,w)$ is the count of neighbor votes on w. Accordingly, we introduce a neighbor voting algorithm (Algorithm 1). Given a user-tagged image I, we first perform k-nn search to find its visual neighbors $NN_f(I,k)$. Then, for each neighbor image, we use its tags to vote on tags of image I. The key idea is illustrated in Figure 2.

Algorithm 1 Learning tag relevance by neighbor voting

Input: Image $I \in \mathcal{I}$ tagged by user $u \in \mathcal{U}$.
Output: $n_f(I,k,w)$, i.e., relevance value of each tag w in I.

```
Find NN_f(I,k), the k nearest neighbors of I.
for w ∈ {w' ∈ T | s(I,w') = 1} do
    n_f(I,k,w) = 0
end for
for image J ∈ NN_f(I,k) do
    if h(J,u) = 0, i.e., not the same user then
        for w ∈ {w' ∈ T | s(J,w') = 1, s(I,w') = 1} do
            n_f(I,k,w) = n_f(I,k,w) + 1
        end for
    end if
end for
```

Recall the intuition that if different persons use the same tags to label visually similar images, the tags are likely to be relevant to the visual content it is describing. We follow the intuition by imposing a user-related restriction in the algorithm: image $J \in NN_f(I,k)$ is excluded in the voting process if there exists user $u \in \mathcal{U}$ such that $h(I,u) = 1$ and $h(J,u) = 1$. In other words, neighbor images from the same user of the seed image are ignored in the voting process.

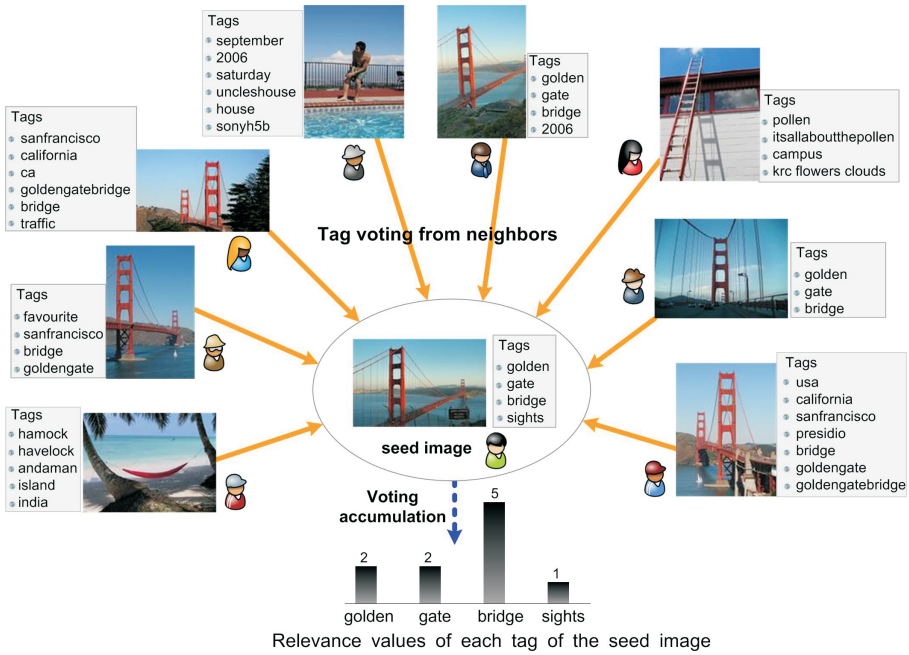


Figure 2: Learning tag relevance by neighbor voting. The relevance value of each tag is estimated by accumulating neighbor votes it receives from visually similar images of the seed image. For instance, five neighbors of the seed image are tagged as *bridge*. So the relevance value of *bridge* with respect to the seed image is 5.

3.3 Searching Visual Neighbors

Visual feature representation. Visual similarity between two images are measured by the similarity between corresponding visual features. Though numerous work have been done for visual feature representation, it is still a challenging problem for content-based image retrieval [25, 6]. Features efficient for extraction and effective for searching large-scale image sets are needed to handle the increasing amounts of social-tagged images. We choose a combined 64-dimensional global feature for its empirically success in searching millions of web images [18, 27]. The feature is calculated as follow. For each image, we extract 44-dimensional color correlgoram in the 44-bin HSV color space [12], 14-dimensional color texture moments [32], and 6-dimensional RGB color moments. We separately normalize the three features into unit length and concatenate them into the final 64-d feature. The dissimilarity between images are measured using the Euclidean distance between features.

Searching millions of images by content. To search millions or even billions of images by content, efficient indexing algorithms are imperative for speed up. Though high dimensional indexing has been studied over decades, it is still a difficult problem. We adopt K-means clustering based indexing methods for their empirical success in large-scale content-based image retrieval [9, 19]. First for indexing, we divide the whole dataset into smaller blocks by K-means clustering. Then for a query, we find neighbors within fewer blocks closest to the query. The search space is thus reduced. The advantages of the architecture are its efficiency for retrieval and its flexibility for updating existing indexes.

4. EXPERIMENTAL SETUP

We evaluate the proposed neighbor voting algorithm within a retrieval framework.

4.1 Tag-based Social Image Retrieval

We employ a general tag-based retrieval pipeline used in existing systems such as Flickr. The retrieval system indexes tags of Flickr images. We adopt Okapi BM25, a well-founded ranking function for text retrieval [15]. Given a query q containing keywords $\{w_1, \ldots, w_n\}$, the relevance score of an image I is computed as,

$$score(q, I) = \sum_{i=1}^{n} qtf(w_i) idf(w_i) \frac{tf(w_i) \cdot (k_1 + 1)}{tf(w_i) + k_1 \cdot (1 - b + b \cdot \frac{L_I}{L_{ave}})}$$

Here, $qtf(w_i)$ is the frequency of tag w_i in query q, $tf(w_i)$ the frequency of w_i in image I, L_I the total number of tags in I, and L_{ave} the average value of L_I over the whole collection. $idf(w_i) = \log \frac{N - |\mathcal{L}_w| + 0.5}{|\mathcal{L}_w| + 0.5}$ is the inverse document frequency (idf) weight of w_i, where N is the total number of images in the collection, and $|\mathcal{L}_w|$ the number of images tagged as w_i. Since individual tags are typically used once per image, original $tf(w)$ is 1. We substitute $tf(w)$ by $n_f(I, k, w) + tf(w)$. It is in this manner that we seamlessly embed learned tag relevance into the retrieval framework. The variable k_1 is a positive parameter for regularizing the effect of tag frequency. The other parameter b ($0 \leq b \leq 1$) determines the scaling by L_I. We refer to [15] for more discussion. In general, k_1 and b need to tune for specific datasets.

4.2 Datasets

We download one million tagged images from Flickr using its API service (`http://www.flickr.com/services/api/`). The images are of medium size with maximum width or height fixed to 500 pixels, and cost 110 GB disk storage in total. The whole dataset has 227,658 unique tags (after Porter stemming) and 96,410 unique user ids.

Since no benchmark dataset is available, we construct a ground truth set for evaluation. Due to the expense of hu-

man labeling, we select 10 queries, consisting of 8 object-level concepts (airplane, bicycle, boat, bridge, car, dog, flower, and tiger) and 2 scene-level concepts (beach and mountain). For each concept, we adopt definitions from WordNet [7], and divide examples into two distinct groups in light of relevance and image quality, that is,

- *relevant examples*: non-blurred images with objects or scenes of the concept clearly visible, without any occlusion. Real world stuff, not toys, cartoon, painting, statues, etc. Should be an external view if the concept is an object.

- *irrelevant examples*: all images not meeting the *relevant examples* standards, e.g., objects or scenes of the concept are invisible, occluded, or rather insignificant in the whole image.

The definitions and examples of concepts are listed in Table 1. For each concept, we randomly select 1000 examples from images tagged as the concept keyword from the Flickr database, and label them according to our labeling criterion. The ground truth statistics is summarized in Table 2.

4.3 Evaluation Criteria

Since users often view fewer result pages [13], images relevant to user queries should be ranked as high as possible. Meanwhile, ranking quality of the whole list is important not only for user browsing, but also for applications using search results as a starting point. We evaluate both aspects by employing *Precision at top X* and *Average Precision*. Given a ranked list L with length n, *Precision at top* $X = \frac{\text{No. of relevant results in top } X}{X}$, where $X \ll n$. We choose $X = 5, 10, 20$. *Average Precision* is computed as $\frac{1}{R}\sum_{j=1}^{n}\frac{R_j}{j}I_j$, where R is the number of relevant instances in L, R_j the number of relevant instances among the top j ranked instances, $I_j = 1$ if the j-th instance is relevant and 0 otherwise. To evaluate the overall performance, we use *Mean Average Precision* (MAP), the mean value of AP scores over all queries.

4.4 Experiments

We index original image tags in a baseline system, called *Baseline* hereafter. Meanwhile, we index the same tags using learned tag relevance value as tag frequency in a new system, called *Neighbor* henceforth. By comparing search accuracy between the two systems, we demonstrate how the proposed algorithm improves social image retrieval.

In total, there are three system parameters to investigate. One is k, the number of neighbors for voting. The other two are k_1 and b in the BM25 ranking formula. Since k_1 does not affect ranking for single-word queries, we divide the evaluation into two settings, i.e., one using single-word queries and the other using multiple-word queries.

Experiment-1: Single-word query. We try various parameter combinations by choosing the neighbor number from {10, 50, 100, 200, 500, 1000, 2000, 3000, 5000, 10000, 20000} and ranging b from 0 to 1 with interval 0.1. Since k_1 does not affect ranking, we set k_1 to 2, a common choice in text retrieval [15].

Experiment-2: Multiple-word query. In this part, we study queries consisting of multiple keywords. We simulate the scenario by expanding single-word queries. That is, for each query, we first find synonyms from both WordNet [7] and an online English Synonym Dictionary (`http://dico.isc.cnrs.fr/dico/en/search`). We then manually select and add four most relevant terms to the original query. If less than four synonyms are found, we add them all. The

Table 1: Definition and visual examples of ten queries used for our experiments with one million Flickr images.

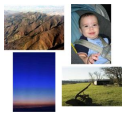
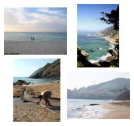
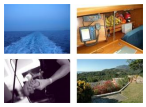
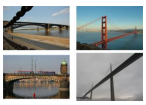
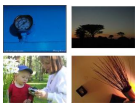
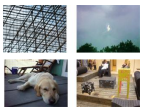
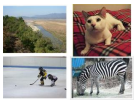
Query	Definition	Examples	
		Relevant	*Irrelevant*
airplane	A heavier than air, fixed-wing aircraft - gliders included. NOT balloons, helicopters, missiles, and rockets.		
beach	An area of sand sloping down to the water of a sea or lake, with both sand and water visible.		
bicycle	A wheeled vehicle that has two wheels and is moved by foot pedals.		
boat	A vessel for travel on water, e.g., canoe, rowboat, kayak, hydrofoil, hovercraft, aircraft carrier, and submarine.		
bridge	A structure that allows people or vehicles to cross an obstacle such as river, canal, or railway.		
car	A motor vehicle with four wheels; usually propelled by an internal combustion engine.		
dog	A member of the genus Canis that has been domesticated by man since prehistoric times.		
flower	a plant cultivated for its blooms or blossoms.		
mountain	A land mass that projects well above its surroundings, higher than a hill, with the slopes visible.		
tiger	Large feline of forests in most of Asia having a tawny coat with black stripes.		

Table 2: Ground truth statistics for our evaluation. Each query has 1000 manually labeled examples. Precision is No_of_relevant_images / 1000.

Query	Total no. of images in the database	Precision
airplane	5622	0.451
beach	60631	0.331
bicycle	4062	0.295
boat	16771	0.417
bridge	13968	0.471
car	31076	0.548
dog	41494	0.846
flower	63006	0.829
mountain	29068	0.502
tiger	1982	0.224

Table 3: Expanded query terms for Experiment-2.

Original query	New query
airplane	airplane, airbus, aircraft, aeroplane, plane
beach	beach, seashore, seaside, shore, strand
bicycle	bicycle, bike, cycle, pedal, wheel
bridge	bridge, connect, cross, link, span
boat	boat, barge, craft, ship, vessel
car	car, auto, automobile, machine, motorcar
dog	dog, canine, cur, mongrel, mutt
flower	flower, bloom, blossom, flourish, prosper
mountain	mountain, heap, stack
tiger	tiger, Panthera, tigris

expanded queries are listed in Table 3. In each new query, we empirically set the query term frequency of the original query term to 5 and that of new added terms to 1 such that the original query is not overwhelmed. We try k_1 from {1, 2, 4, 8, 16} and range b from 0 to 1 with interval 0.1.

Experiment-3: The impact of database size. Since the amount of social-tagged images is rapidly increasing, an interesting problem is whether the neighbor voting algorithm will improve search accuracy as the database grows. We conduct a simulated experiment to gain insight into the problem. The experimental procedure is as follows. First, we construct a subset of size x by randomly select x images from the whole 1M database. x is ranged from 100K to 1M with interval 100K. Then, we use the selected subset to learn tag relevance for images in the evaluation set. The procedure is repeated 10 times for each x.

5. RESULTS

5.1 Experiment-1: Single-word Query

As shown in Figure 3, *Neighbor* clearly outperforms *Baseline* for all parameter settings. The result verifies the effectiveness of the proposed algorithm. One interesting observation is different behavior of the ranking parameter b in the two systems. *Baseline* tends to perform well when b approaches 1. By contrast, *Neighbor* improves as b approaches 0. Recall that b controls the impact of normalizing scores by the total number of tags within an image. Since tag frequency is not discriminative in original tagging, the baseline system heavily relies on the normalization factor. While in the new system, tag frequency becomes more discriminative to distinguish relevant examples from irrelevant ones after tag relevance learning. This observation further demonstrates the effectiveness of learning tag relevance by neighbor voting.

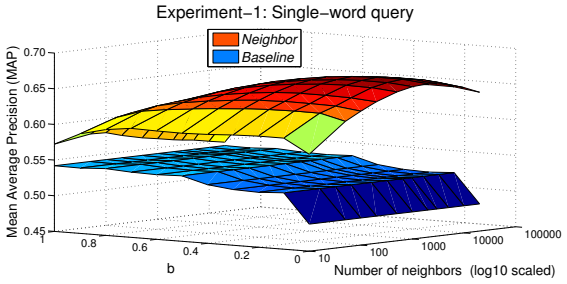


Figure 3: Experiment-1: Single-word query. We evaluate overall search performance by MAP.

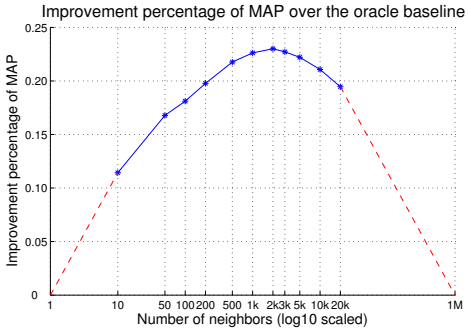


Figure 4: Improvement percentage of MAP over the oracle baseline with respect to the number of neighbors. The oracle baseline is reached at $b = 0.8$ with MAP=0.544.

The neighbor voting algorithm is also robust. As shown in Figure 4, *Neighbor* ($k_1 = 2, b = 0.1$) consistently outperforms the oracle *Baseline* ($k_1 = 2, b = 0.8$). Specifically, we reach at least 20% improvement of MAP when the neighbor number is widely ranging from 200 to 10000. In two extreme cases when the neighbor num is very small or very large, *Neighbor* degenerates to *Baseline*. We fix the number of neighbors to 1000 throughout the rest of experiments as a moderate trade-off between accuracy and efficiency.

We further provide a per-query comparison. The parameters for *Neighbor* are $k = 1000, k_1 = 2, b = 0.1$, and those for *Baseline* are $k_1 = 2, b = 0.8$. Evaluation results using *Precision at top 5,10,20* and *Average Precision* are summarized in Table 4. Top ten search results are presented in Figure 5. *Neighbor* improves *Baseline* in general. On average, we improve MAP by around 24%. An interesting result is the query "tiger". The top results of *Baseline* are either tigers of distance view or cats. While in top results of *Neighbor*, such examples are brought down. Instead, tigers of close-up view are ranked top. Note that irrelevant images of toy tigers or places named tigers are brought up as well. There might be two reasons for this. First, the "tiger" query is itself very ambiguous. Second, there are only 1982 images tagged as tiger in our Flickr database (see Table 2). The insufficient examples may weaken the estimation of tag relevance.

5.2 Experiment-2: Multiple-word Query

Similar to the single-word query scenario, *Neighbor* improves *Baseline* with a clear margin for all parameters, as shown in Figure 6. The result again shows the effectiveness of the proposed algorithm. We notice for both systems, the ranking parameter k_1 dose not affect the performance much.

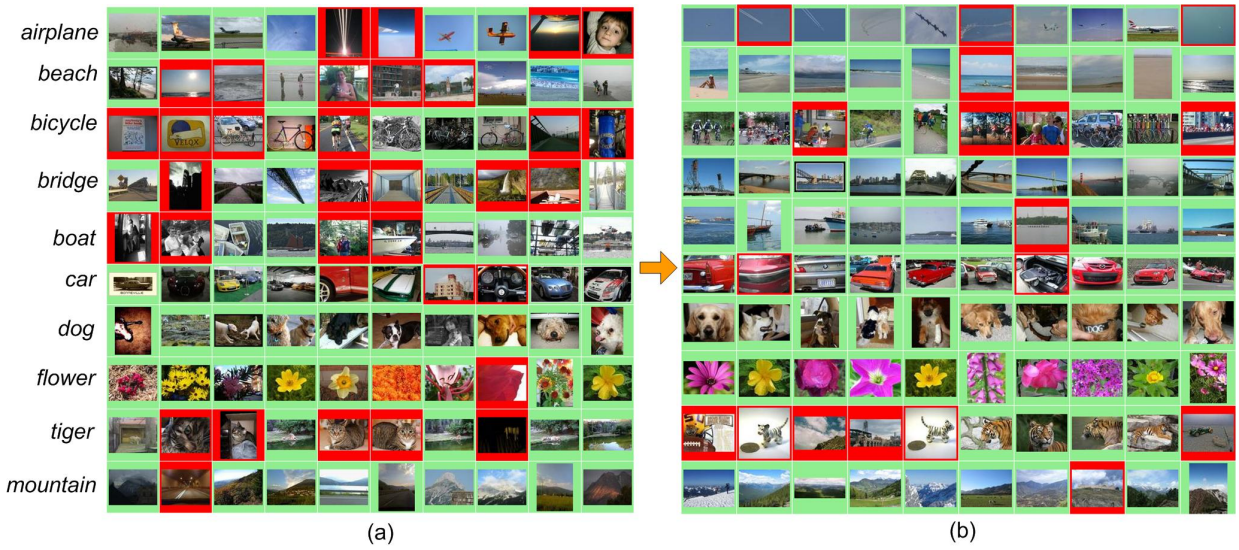


Figure 5: Top ten search results of ten queries. (a) is the baseline results, and (b) the results using learned tag relevance as tag frequency for retrieval. Green color denotes relevant examples, and red color for irrelevant ones.

Table 4: Per-query comparison. *Baseline* is the baseline system. *Neighbor* is the new system using learned tag relevance as tag frequency for retrieval. Bold numbers in the table indicate top performers.

	Precision at Top 5		Precision at Top 10		Precision at Top 20		Average Precision	
Query	Baseline	*Neighbor*	*Baseline*	*Neighbor*	*Baseline*	*Neighbor*	*Baseline*	*Neighbor*
airplane	0.80	0.80	0.60	**0.70**	0.45	**0.70**	0.42	**0.59**
beach	0.40	**1.00**	0.50	**0.90**	0.50	**0.85**	0.50	**0.63**
bicycle	0.20	**0.80**	0.40	**0.60**	0.25	**0.45**	0.28	**0.40**
boat	0.40	**1.00**	0.60	**0.90**	0.55	**0.75**	0.46	**0.57**
bridge	0.60	**1.00**	0.50	**1.00**	0.55	**0.85**	0.52	**0.60**
car	**1.00**	0.80	0.80	0.80	0.55	**0.90**	0.62	**0.72**
dog	1.00	1.00	1.00	1.00	1.00	1.00	0.91	**0.92**
flower	1.00	1.00	0.90	**1.00**	0.90	**1.00**	0.89	**0.96**
mountain	0.80	**1.00**	0.90	0.90	0.70	**0.85**	0.53	**0.76**
tiger	**0.40**	0.00	**0.50**	0.45	0.50	**0.60**	0.42	**0.50**
average	0.66	**0.84**	0.67	**0.82**	0.59	**0.80**	0.54	**0.67**

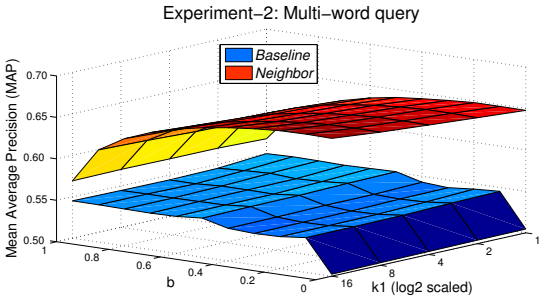


Figure 6: Experiment-2: Multi-word query. We evaluate overall search performance by MAP.

5.3 Experiment-3: The Impact of Database Size

The search performance, i.e., MAP, improves, as the database size x increases (see Figure 7). To model the relationship between x and MAP, we approximate the mean value curve by a log function $f(x) = 0.0113 \times \ln \frac{x}{100000} + 0.6407$. If the rules holds when one billion images are used, we will obtain an MAP of 0.74, meaning 37% improvement over the oracle baseline.

6. CONCLUSIONS

Since amateur tagging is known to be ambiguous, noisy, and personalized, a fundamental problem in social image retrieval is how to reliably learn the relevance of a tag with respect to the visual content it is describing. In this paper, we propose a neighbor voting algorithm as an initial step towards solving the problem. The key insight is to learn the relevance of a tag with respect to an image from tagging behavior of visual neighbors of that image. Specifically, estimating tag relevance boils down to counting neighbor votes on tags. For a query tag, we show that if the visual neighbor search is better than random sampling, the relevant set to the query and the irrelevant set are distinguishable by learned tag relevance. The advantage of the proposed algorithm are three-fold: 1) reliable, since only common tags are propagated between neighbors without introducing new tags to an image. Such self-validation scheme reduces the risk of incorrectly propagating irrelevant tags; 2) scalable, since the

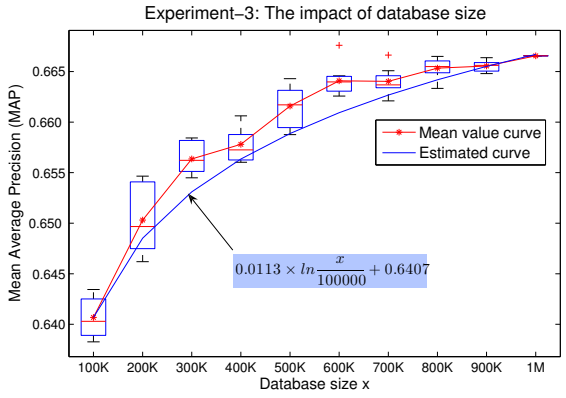


Figure 7: Experiment-3: The impact of database size. Search performance of the *Neighbor* system improves as the database size increases.

proposed method does not require any model training for any visual concepts; 3) flexible, since learned tag relevance information, treated as tag frequency, can be seamlessly embedded into current tag-based retrieval framework.

Experiments on one million Flickr images verify the proposed algorithm. We study retrieval performance for both single-word queries and multiple-word queries. Overall performance comparisons and per-query analysis show the effectiveness of the algorithm. Compared with the baseline using the original tags, on average, retrieval using improved tags increases *mean average precision* by 24%, from 0.54 to 0.67. Furthermore, simulated experiments predict that when one billion images are used, we might obtain an MAP of 0.74, meaning a 37% improvement over the baseline.

Acknowledgements

This work was supported by the EC-FP6 VIDI-Video project.

7. REFERENCES

[1] E. Auchard. Flickr to map the world's latest photo hotspots, November 2007. Reuters, `http://www.reuters.com/article/technologyNews/idUSHO94233920071119?sp=t%rue`, retrieved on 2008-06-16.

[2] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *JMLR*, pages 1107–1135, 2003.

[3] G. Begelman, P. Keller, and F. Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *WWW Collaborative Web Tagging Workshop*, 2006.

[4] E. Chang, G. Kingshy, G. Sychay, and G. Wu. Cbsa: content-based soft annotation for multimodal image retrieval using bayes point machines. *TCSVT*, 13(2):26–38, 2003.

[5] T.-S. Chua, S.-Y. Neo, K.-Y. Li, G. Wang, R. Shi, M. Zhao, and H. Xu. TRECVID 2004 search and feature extraction task by NUS PRIS. In *TRECVID Workshop*, 2004.

[6] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Survey*, 40(2):1–60, 2008.

[7] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

[8] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *ECCV*, pages 242–256, 2004.

[9] H. Ferhatosmanoglu, E. Tuncel, D. Agrawal, and A. E. Abbadi. Approximate nearest neighbor searching in multimedia databases. In *ICDE*, pages 503–511, 2001.

[10] S. A. Golder and B. A. Huberman. Usage patterns of collaborative tagging systems. *Jour. Information Science*, 32(2):198–208, 2006.

[11] W. Hsu, L. Kennedy, and S.-F. Chang. Video search reranking via information bottleneck principle. In *ACM Multimedia*, pages 35–44, 2006.

[12] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. In *CVPR*, pages 762–768, 1997.

[13] B. J. Jansen and A. Spink. How are we searching the world wide web?: a comparison of nine search engine transaction logs. *JIPM*, 42(1):248–263, 2006.

[14] Y. Jin, L. Khan, L. Wang, and M. Awad. Image annotations by combining multiple evidence & Wordnet. In *ACM Multimedia*, pages 706–715, 2005.

[15] K. S. Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments - part 2. *Jour. Information Processing and Management*, 36(6):809–840, 2000.

[16] L. Kennedy, M. Naaman, S. Ahern, R. Nair, and T. Rattenbury. How flickr helps us make sense of the world: Context and content in community-contributed media collections. In *ACM Multimedia*, pages 631–640, 2007.

[17] J. Li and J. Z. Wang. Real-time computerized annotation of pictures. *TPAMI*, 30(6):985–1002, 2008.

[18] X. Li, L. Chen, L. Zhang, F. Lin, and W.-Y. Ma. Image annotation by large-scale content-based image retrieval. In *ACM Multimedia*, pages 607–610, 2006.

[19] X. Li, X.-J. Wang, C. Wang, and L. Zhang. SBIA: search-based image annotation by leveraging web-scale images. In *ACM Multimedia*, pages 467–468, 2007.

[20] W.-H. Lin and A. Hauptmann. Web image retrieval re-ranking with relevance model. In *Web Intelligence*, pages 242–248, 2003.

[21] K. Matusiak. Towards user-centered indexing in digital image collections. *OCLC Systems and Services*, 22(4):283–298, 2006.

[22] F. F.-H. Nah. A study on tolerable waiting time: how long are Web users willing to wait? *Jour. Behaviour and Information Technology*, 23(3):153–163, 2004.

[23] G. Park, Y. Baek, and H.-K. Lee. Majority based ranking approach in web image retrieval. In *CIVR*, pages 499–504, 2003.

[24] N. Sebe and Q. Tian. Personalized multimedia retrieval: the new trend? In *ACM MIR Workshop*, pages 299–306, 2007.

[25] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *TPAMI*, 22(2):1349–1380, 2000.

[26] A. Torralba, R. Fergus, and W. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *TPAMI*, 2008 (in press).

[27] C. Wang, F. Jing, L. Zhang, and H.-J. Zhang. Scalable search-based image annotation of personal images. In *ACM MIR Workshop*, pages 269–278, 2006.

[28] C. Wang, F. Jing, L. Zhang, and H.-J. Zhang. Content-based image annotation refinement. In *CVPR*, pages 1–8, 2007.

[29] X.-J. Wang, L. Zhang, X. Li, and W.-Y. Ma. Annotating images by mining image search results. *TPAMI*, 2008 (in press).

[30] Y. Wu, J.-Y. Bouguet, A. Nefian, and I. V. Kozintsev. Learning concept templates from web images to query personal image database. In *ICME*, pages 1986–1989, 2007.

[31] R. Yan, A. Hauptmann, and R. Jin. Multimedia search with pseudo-relevance feedback. In *CIVR*, pages 649–654, 2003.

[32] H. Yu, M. Li, H. Zhang, and J. Feng. Color texture moment for content-based image retrieval. In *ICIP*, pages 929–932, 2002.