An Adaptive Query Prototype Modeling Method for Image Search Reranking

Hong Lu, Guobao Jiang, Bohong Yang, Xiangyang Xue
Shanghai Key Lab of Intelligent Information Processing
School of Computer Science, Fudan University, Shanghai, China
{honglu,guobaojiang,13210240072,xyxue}@fudan.edu.cn

Abstract

As more and more images with user free tags are appearing on the Internet, image search reranking has received considerable attention to help users obtain images relevant to the query. In this paper, we propose to rerank the initial image search results using an adaptive query modeling method based on local features. For a query and its initial rank list, we construct a visual word dictionary using the randomly selected images in the initial rank list. Then the top N images are incrementally used to evaluate the importance/score of the words for the query. The words with higher scores are selected to model the query. The relevance scores of the words to the query are also kept. Then the model is used to measure the relevance of each image in the image list to the query and rerank the image list according to the measured relevance. This method can obtain optimal local prototype for a query from the top N images (N is not large) without considering many images. Also, the relevances of the top N images to the query are not considered equally by weighting according to their positions in the initial image list. Using this model, the influences from some wrongly returned images at the top of the initial rank list can be filtered out. This method is also query independent and can be used to any other query. Experimental results on a large scale image dataset of WebQueries demonstrate the efficacy of the proposed method.

1. Introduction

As more and more images with user free tags appearing on the Internet, it is necessary to help a user to effectively obtain images relevant to the user submitted query. On one hand, existing web image search engines normally return images for a query focusing on associated texts. However, keyword based image search is not good enough to satisfy users' requirements. For example, it is reported in [20] that Google's image search engine can have as low as 32% precision and 39% average precision for keyword based search. Also, it needs to be note that there are some wrongly returned images also appearing on the top of the rank list, which may be caused by the noisy tag/text surrounding the image.

On the other hand, content based image retrieval (CBIR) [21] considers the visual features. It returns the images with higher visual similarity to the query image on measured visual features such as color, texture, shape, etc. and the metrics such as Euclidean distance, Mahalanobis distance [18], etc. However, it would be reluctant or difficult for many users to provide the query image or the query image is difficult to be given. And some applications also need to harvest images for one query from the web [20, 24].

Combining the two mechanisms is shown to be a reasonable solution for keyword based image retrieval. That is, images are first retrieved by text based search engine. The search results are then reranked using the initial rank images' visual content. And the image visual reranking aims to keep the relevant images to be higher ranked in the new reranked image list. The query modeling based on visual content has been widely used not only in image reranking [1, 3, 5, 9, 28], but also in tag ranking[15], harvesting images [6, 13, 20], and video retrieval [11].

From the result of image retrieval, it can be observed that compared with the bottom ranked images, the top ranked (although with few irrelevant images) have higher relevance with the query. Thus, for image reranking, pseudo relevance feedback (PRF) assumption is widely use. For one query, the assumption regards the top-N returned images as positive samples to learn the model of the query [25][26] [12]. And in the model, the top images are usually considered weighted equality without considering the position influence. Motivated by the information retrieval measure of Normalized Normalized Discounted Cumulative Gain (NDCG) [10] to measure the ranking result, we propose to adaptively weight the image visual feature with the position information. Specifically, the idea of NDCG is that highly relevant documents are more useful when appearing earlier in a search engine result list, i.e. having higher ranks. Then we propose to give higher importance to the images with higher ranks in the initial rank list. Another assumption is visual consistency. This assumption means that relevant-relevant image pairs share higher visual similarity compared with that of the relevant-irrelevant and irrelevant-irrelevant image pairs. Then the clustering, classification, and learning to rerank methods can be applied. For example, in [9], the frequently occurred images are selected as the prototypes of the query. On the other hand, clustering methods are used to cluster the initial returned images. The representative images are selected from each cluster to represent the variation of the images relevant to the query [1, 3, 5, 28]. In this paper, we also consider the visual consistency of the initial returned images.

The rest of the paper is organized as follows. Section 2 briefly introduces the related work on image search reranking on visual features. Our proposed model is discussed in Section 3. Experimental results and some discussions are given in Section 4. And we conclude our work in Section 5.

2. Related work

The images reranking methods can be classified into that of classification based [12, 25], clustering based [2, 3, 4, 7, 14], graph based [8, 16], and learning to rerank [22, 26, 27].

Classification based methods use pseudo relevance feedback (PRF) assumption. That is, the top-N images of the initial search result for one query are regarded as pseudo relevant and can be chosen as positive samples. The negative samples are selected from the images in the bottom of the rank list or from other queries's results. Then a classifier is trained using these positive and negative samples [25]. And the images in the initial list are classified and reranked using the trained classifier.

Clustering based methods, such as [2, 3, 4, 7, 14], have similar several integral components. Specifically, in [2], each image is segmented into similar regions or "blobs". A set of features are extracted from each of these blobs and used for clustering. Lastly, initial rank images are reranked based on similarity clusters. And, most of clustering-based methods [7] [2] use global features like color or texture features. However, the foreground parts in the image will have more influence to the relevance score of the image to the query keyword. And these global features can't well express the foreground and local property of image. What's more, human's vision mainly focuses on some local interesting points when viewing the image. So, we use local image descriptors such as scale invariant feature transform (SIFT) [17] to describe the initial images' key points.

The graph based method can also be called random walk-based, which consists of two parts. Part one is graph construction. A graph is constructed with the images as the nodes and the edges between them being weighted by image visual similarity. The other part is random walk for reranking. The process of reranking is formulated as random walk over the graph and the relevance scores are propagated through the edges. So, the relevance of the image to the query keyword can be measured by relevance scores defined on the graphs [8, 16].

Recent years, human supervision is introduced to learn the model for one query and the work is called "learning to rank". Specifically, [26] and [27] try to train a generic (query-independent) model offline through the images that are labeled whether relevant to the query or not. A number of images are used to represent the query and form the meta rerankers. Then these meta rerankers are used to refine the initial rank result online. This model can also be called "learning to rerank."

Exemplar model [19] is proposed to detect objects in images. In the training phase, it uses all images' local feature to build a vocabulary of size L. Then it selects l words from the vocabulary to build model of each class. These l words are selected based on the criterion that they have high occurrence in the images from one specific class. Then the finalized discriminative words can be obtained, and these K discriminative words can be used to model the class. In this paper, we explore the idea of obtaining the local features' re-occurrence to model the query. As mentioned in [23], in object detection, some words from the background parts may have negative influence to the object modeling. Thus, we first constructed the dictionary with enough words based on the initial rank list. The images are randomly selected which include both relevant and irrelevant images. Then a discriminative model is constructed by exploring the consistency or co-occurrency of the words in the top image of the initial rank list.

On the other hand, considering initial rank equally is not suitable for the query results since the images in the top position of the list are normally more relevant compared with the images behind them. [22, 27] consider the initial position influence and propose to use the top images as the prototypes of the query to form the meta-rerankers of the query. So in our local and discriminative modeling method, the images and their contributions to word selection are based on their positions in the initial rank list.

3. Our Proposed Reranking Framework

3.1. Motivation

According to observation, for an initial rank list returned by a specific search engine, most of images are relevant to the query (especially the top N images). What's more, relevant images which are visually similar are apt to cluster together. However, the irrelevant images also distribute widely. For example, Fig. 1 shows the top ten images in the initial rank list for the query "**arc de triomphe**". It includes six relevant images (a, c, e, f, g, i). The rest of irrelevant images show diversity each other. Thus, if we obtain the similar elements from the initial image list, we'll get a dis-

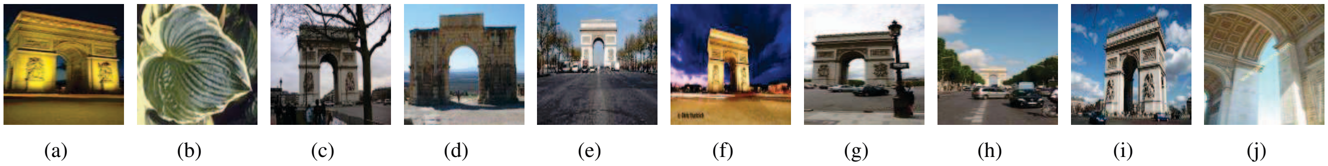

Figure 1. The top 10 returned images to query *arc de triomphe*. The relevant images are (a),(c),(e),(f),(g),(i).

criminative model to describe the query's visual contents. The elements relevant to the query will be increased in the modeling phase while the irrelevant elements will become relatively weaker because they are widely distributed. Finally, we get a discriminative model that can represent the query. We use the model to rerank the initial rank list.

3.2. Framework Description

In this section, we describe the proposed reranking model in detail. The framework is illustrated in Fig. 2. It includes four steps:

- Choose images randomly to construct dictionary.
- Obtain visual dictionary by clustering.
- Train a query-relative model using top N images.
- Rerank the initial rank list with the trained model.

At the first step, we randomly choose M images from initial image list. Then, we obtain a visual dictionary by clustering. Images are chosen randomly from the initial list for clustering so that the dictionary can represent diverse visual contents. Next, we assume that most of top N initial images are relevant to the query. We use the top N images to train a query-relative model on the constructed visual dictionary. At the last step, we use the query-relative model to rerank the initial image list. Finally, the reranking result will be returned to the users.

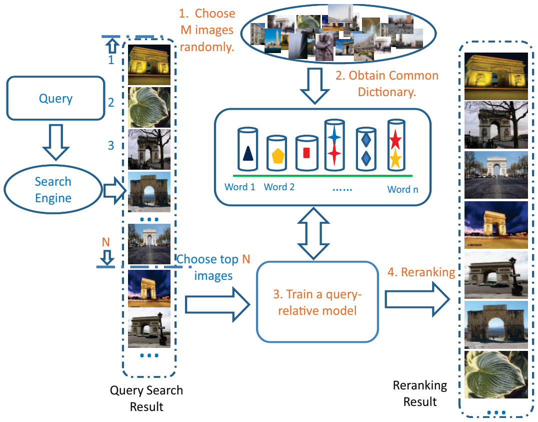

Figure 2. Proposed re-ranking process using adaptive query-relative modeling method.

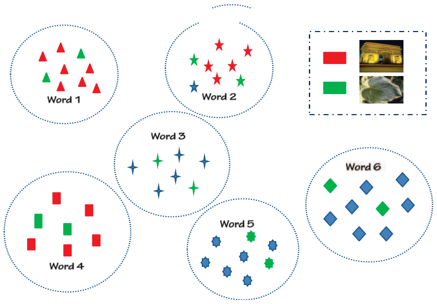

Figure 3. A dictionary example, which includes six words. Red, Green represent two different images. Qurey="**arc de triomphe**" and different shapes represent different clusters (visual contents).

3.3. Obtain Visual Dictionary

Before obtaining a visual dictionary, we extract local features, i.e. SIFT [17] on key points of the images. Then each image in the initial rank list can be represented by the distribution on the visual dictionary. As an example, Fig. 3 shows a visual dictionary including six words (clusters). In the figure, red represents the relevant image and its visual contents are represented by different shapes. Green represents the irrelevant image. From this distribution, it can be observed that relevant images will be clustered into few words (word 1,2,4) while the irrelevant images will be clustered into a variety of words (word 1,2,3,4,5,6). If we only use these two images to train a query-relative model, the final cluster's scores are given in TABLE 1.

Image	Word1	Word2	Word3	Word4	Word5	Word6
Red	$\frac{7}{17}$	$\frac{5}{17}$	$\frac{0}{17}$	$\frac{5}{17}$	$\frac{0}{17}$	$\frac{0}{17}$
Green	$\frac{2}{12}$	$\frac{2}{12}$	$\frac{2}{12}$	$\frac{2}{12}$	$\frac{2}{12}$	$\frac{2}{12}$
Score	$\frac{7}{12} + \frac{2}{17}$	$\frac{5}{12} + \frac{2}{17}$	$\frac{2}{12}$	$\frac{5}{12} + \frac{2}{17}$	$\frac{2}{12}$	$\frac{2}{12}$

Table 1. Calculate the final cluster's scores in Fig. 3 if only use the two images for modeling.

3.4. Train A Query-Relative Model

In this phase, we train a query-relative model which will be used to represent the user's submitted query. As mentioned before, most of the top images are more relevant to the query in the initial rank list. Here, we assume that top N images of the initial rank list are more relevant to the query (of course may contain few noise irrelevant images). And we use top N images to train a query-relative reranking model. In this work we denote the top N images as a set T_N=$\{I_i|I_i$ represents the *i-th* image in the top N images, i=1,2,...,$N\}$. The value of N is related to the query or is based on empirical study.

Our final query-relative model is composed of K clusters (words). Each cluster (word) is represented by its cluster center, cluster score, and intra cluster distance. But, it doesn't mean that all clusters have the same ability to describe the query. And the ability depends on the cluster's score S_i. The higher the cluster score (S_i) is, the more relevant the cluster to the query. Here, we denote the final model as M_q=$\{W_k$=$(C_k$,S_k,$D_k)\}$, C_k, S_k, D_k represent the cluster center, cluster score, intra cluster distance of the *k-th* cluster, respectively, W_k the *k-th* word (cluster), k=1,2,...,K, where q denotes a specific query.

We get a visual dictionary after we finish the second phase (Section 3.3).We denote the visual dictionary as a set $Dict$=$\{W_i$=$(C_i$,S_i,$D_i)|W_i$ represents the *i-th* word (cluster), i=1,2,...,$\phi\}$. As we know, each word represents a local visual feature (content). And, a query can be described by some of the words. Then we need to determine which words have the ability to describe the query.

We use the top N images (set T_N) to train the final model (M_q) through three steps as follows.

1. For the *i-th* image $I_i \in T_N$, calculate its contribution for each word in dictionary ($Dict$):

$$Con_i(k) = \frac{G_{ik}}{N_i^c} \tag{1}$$

where N_i^c denotes the total number of *i-th* image's clustered key points, G_{ik} the number of key points which are clustered into *k-th* word (cluster). **Clustered key points** means the points which can be classified into the dictionary. On the contrary, the points which can't be classified into any word (the distance between the point and cluster center is larger than the intra cluster distance) are called **Unclustered key points**, their number is denoted as N_i^u. So the total number of *i-th* image's key points is $N_i^c + N_i^u$.

2. Calculate the cluster score:

$$S_k = \sum_{i=1}^{N} \omega_i Con_i(k) \tag{2}$$

where ω_i is the weight of the *i-th* image.

3. Obtain the final model:

$$M_q = \{W_k = (C_k, S_k, D_k)|S_k > Th_q, W_k \in Dict\} \tag{3}$$

where Th_q is the threshold of the *q-th* query.

During the whole training phase, two parameters need to be decided. That is, the weight of the *i-th* image (ω_i), and the value of N. We give our solutions as follows.

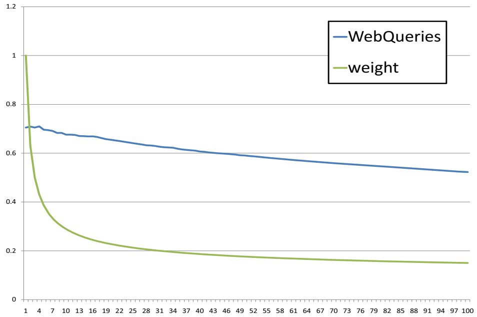

Figure 4. The R-Precision performance on WebQueries (the vertical axis represents precision, horizontal axis R). The precision is 69.61% when R sets to 5.

Weight Solution: How to estimate the weight of the *i-th* image (ω_i)? In this work, we conduct two experiments (**Equal Weight Experiment** and **Unequal Weight Experiment**) using the two methods and make a comparison. On one hand, the weight can be viewed equally in the initial rank list. On the other hand, the weight can be determined based on the position of the *i-th* image in the initial rank list, which can be formulated as below:

$$\omega_i = \frac{1}{log(i+1)} \tag{4}$$

Here, using logarithmic function is motivated by the discount term in Normalized Discounted Cumulative Gain (NDCG) [10], which assigns a larger importance to top images in the initial list since their relevance to the query is assumed to be larger.

As indicated in [26], most image search engines, such as Google, Yahoo, Bing, have good performance in the top several images. Besides, we investigate the dataset of WebQueries [1] and obtain the R-Precision [2] result. R-Precision is the precision at *R-th* position in the initial rank list for a query. Fig. 4 illustrates the R-Precision performance on WebQueries. This figure indicates that, when R is set to 5, the precision gets the value 69.61% for the dataset of WebQueries.

Choose Optimal N: Another problem is how to determine the optimal value of N so that we can achieve the best performance for a specific query. Fig. 5 illustrates the training process. The words (clusters) in the dictionary are ordered by the number of the clustered points after an incoming image's key points clustered. When the top κ words set becomes stable, this words set can represent the query. For example (in Fig. 5), when the *i-th* image arrives, its key points will affect the words' distribution during modeling. After the *i-th* image's key points clustered, if most of elements of the top κ words set are still in the set of top

[1] http://lear.inrialpes.fr/pubs/2010/KAVJ10/

[2] http://en.wikipedia.org/wiki/Average_precision#R-Precision

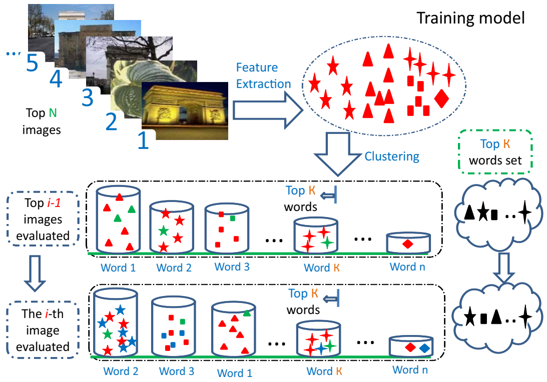

Figure 5. The training model: the process of choosing N.

Symbol	meaning
T_N	The set of top N images for a specific query.
M_q	The final reranking model.
$Dict$	The common dictionary.
$Con_i(k)$	The *i-th* image's contribution to *k-th* word (cluster).
N_i^c	The number of clustered key points for *i-th* image.
N_i^u	The number of unclustered key points for *i-th* image.
S_k	The score of *k-th* word (cluster).
ω_i	The weight of the *i-th* image for training model.
Th_q	The threshold of the *q-th* query.
δ	The choosing factor for the top N images.

Table 2. The meaning of symbols described in the phase of training a query-relative model.

κ words set, we call this state as stable. At that time, the optimal value of N is set to i.

To better understand the process of choosing the optimal value of N, suppose, when the *i-th* image arrives, the *k-th* word (cluster) has $P_{i-1}(k)$ key points and the dictionary's top κ words are denoted as $\kappa_{i-1} = \{W_1^{i-1}, W_2^{i-1}, ..., W_\kappa^{i-1}\}$. After the *i-th* image is evaluated, the dictionary's top κ words become $\kappa_i = \{W_1^i, W_2^i, ..., W_\kappa^i\}$. We define the choosing factor as:

$$\delta = \frac{\varphi(\kappa_i \cap \kappa_{i-1})}{\kappa} \tag{5}$$

where $\varphi(\cdot \cap \cdot)$ denotes the number of elements in the intersection. If this factor $\delta > \theta$, we call this state as stable, otherwise unstable. θ is set based on our empirical study. The main symbols described above are shown in TABLE 2.

3.5. Rerank the List with the Trained Model

In this phase, we use the trained model to rerank the initial image list returned by search engine. Local visual feature, i.e. SIFT, will be extracted before reranking. Thus, each image has some key points. And some points can be classified into the word (cluster) of the trained model, and the others can't. The unclustered points will be abandoned. Based on the score of word (cluster) and the clustered key point information, we obtain each image's score. Finally, we rerank the initial list according to the image's score.

The *i-th* image's score can be described as

$$ImageScore_i = \omega_i \frac{1}{N_i^c} \sum_{j=1}^{N_i^c} S_{k_j} \tag{6}$$

where $ImageScore_i$ denotes the score of *i-th* image, N_i^c the number of clustered key points, k_j the kind of word (cluster) which *j-th* point is clustered into, S_{k_j} the score of k_j*-th* cluster, ω_i the weight of *i-th* position in the initial rank list.

4. Experiments

In this section, we evaluate the adaptive query-relative prototype modeling method for image reranking. In the experiment, we use public dataset so that we can make a comparison of our proposed method with previous related work such as [12].

4.1. Experiment Dataset

***WebQueries*:** We conduct experiments on dataset of WebQueries[12]. It contains 71478 images retrieved from a web search engine for 353 different queries. For each query, the dataset includes the original textual query, the top-ranked images found by the web search engine, and an annotation file for each image by human labeling. For each query, 300 ~ 500 images on the top of text queried ranking list are obtained.

To illustrate a more detailed evaluation and make comparison, we group concepts (as in [12]) into several sets as following:

- Low Precision (LP): 25 queries where the search engine performs worst, e.g. 'will smith', 'rugby pitch', 'bass guitar', 'mont blanc', 'jack black'.

- High Precision (HP): 25 queries where the search engine performs best, e.g. 'batman', 'aerial photography', 'shrek', 'pantheonrome', 'brazil flag'.

- Search Engine Poor (SEP): 25 queries where the search engine improves least over random ordering of the query set, e.g. 'clipart', 'cloud', 'flag', 'car'.

- Search Engine Good (SEG): 25 queries where the search engine improves most over random ordering, e.g. 'rugby pitch', 'tennis court', 'golf course'.

- Overall: all queries or concepts (353 in total).

4.2. Reranking Results on WebQueries

The weight ω_i will affect the final reranking results. To make a comparison, we conduct two experiments. One is for the same weight, i.e. all ω_i are set to equal values. Another is for the different weights which are set using Equ. (4).

1. Equal Weight Experiments: TABLE 3 shows the reranking result when N, i.e. the number of top images used for training, is set to different values of 5 to 50. And we make a comparison to the approach proposed in [12].

mAP×100	Overall	LP	HP	SEP	SEG
Search engine	56.9	26.8	83.0	52.5	31.5
Base text features	53.7	24.0	82.0	58.9	49.3
+ partial match [12]	54.8	23.5	82.4	60.3	51.2
Best (b=100) [12]	57.0	24.3	84.1	62.4	54.8
Best (a=400) [12]	64.9	24.1	91.0	71.9	58.4
N=5	58.26	**27.53**	89.77	84.60	**61.46**
N=10	59.11	25.68	**89.90**	86.38	**61.46**
N=15	**59.49**	27.15	89.43	86.16	60.21
N=20	59.09	26.61	89.18	85.90	56.86
N=30	58.57	25.53	89.47	86.59	56.29
N=40	58.02	24.52	89.71	86.76	54.83
N=50	57.55	23.35	89.86	**87.00**	54.31

Table 3. Reranking results on WebQueries when using equal weight strategy. Where a represents number of visual features, b number of context features as in [12].

From Table 3, it can be observed as follows.

- The proposed method (equal weight) performs better than [12] in most sets of concepts when using pure visual features (not consider texture information). For example, the best performance of LP, SEP, SEG (27.53%, 87.00%, 61.46%) are better than the best performance of that in [12] (24.1%, 71.90%, 58.40%).

- For the set of HP (High precision), it can be observed that if the search engine shows good result, the variable N has less affect on the final results. No matter what's the value of N, the images that we use to train the model are likely to be relevant to the query. So the model can describe the query very well.

- For the set of LP (Low precision), if the search engine doesn't shows good result, the value of N has significant affect on the final results. The more images we use to train the model, the worse the final result we may get. It lies that the larger the value of N, the more irrelevant images are used to train the final model. So, we need to determinate optimal value of N.

- For the set of SEP (Search Engine Poor), it can be observed that the best performance achieved when N is set to 50. Since the 25 queries obtained by improving least over random ordering of the query set, so we may need more images to model these queries.

2. Unequal Weight Experiments: TABLE 4 shows the reranking result when the weight is based on the position of the image in the initial rank list (Equ. (4)).

mAP×100	Overall	LP	HP	SEP	SEG
Search engine	56.9	26.8	83.0	52.5	31.5
Base text features	53.7	24.0	82.0	58.9	49.3
+ partial match [12]	54.8	23.5	82.4	60.3	51.2
Best (b=100) [12]	57.0	24.3	84.1	62.4	54.8
Best (a=400) [12]	64.9	24.1	91.0	71.9	58.4
N=5	58.37	29.24	89.76	84.34	61.85
N=10	59.38	28.51	89.90	86.04	**62.87**
N=15	**59.78**	**29.49**	89.73	86.09	61.51
N=20	59.67	28.74	89.64	86.02	59.46
N=30	59.53	28.52	90.03	86.57	59.10
N=40	59.10	27.34	90.21	86.81	57.85
N=50	58.75	26.19	**90.30**	**86.96**	57.15

Table 4. Reranking results on WebQueries when using unequal weight strategy. Where a represents number of visual features, b number of context features as in [12].

From TABLE 4, it can be observed as follows.

- Unequal weight method achieves better result than that of using equal weight. That is, it needs to assign a higher importance to top images in the initial rank list since their relevance to the query is assumed to be larger.

- Similar to the results given by Table 3, the best performances are obtained when N is set between 10 and 20.

- For the set of HP, our proposed method can achieve the best performance as much as 90.30%, which is comparable with that in [12] (91.0%).

What's more, we give a statistic information in Fig. 6 to show how many queries are improved after reranking using the two weight strategies. It also illustrates that the best performances are obtained when N is set between 10 and 20.

3. The Optimal N Experiments: In the former two experiments, we conduct our experiments when N gets different values (like 5,15,20, etc.). However, for different queries, we adopt a query-relative optimal value N in the experiments. Here, we set different thresholds (θ) for choosing the optimal N. The results are given in TABLE 5.

What's more, we do the reranking experiments using PRF (Pseudo-relevance Feedback) method [25] on the dataset of WebQueries. We use the color features such as CAC (color auto-correlogram), CCV (color coherence vector), CLD (color layout descriptor), CSD (color structure

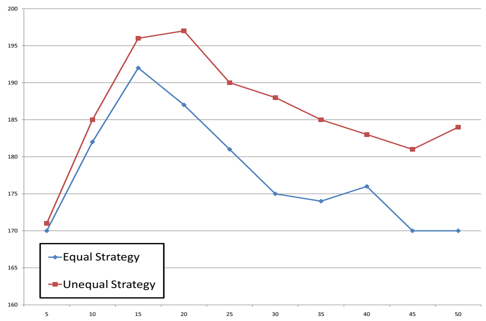

Figure 6. Number of queries improved after reranking when using two different weight strategies over the different values of N (the horizontal axis).

mAP×100	Overall	LP	HP	SEP	SEG
Search engine	56.9	26.8	83.0	52.5	31.5
Base text features	53.7	24.0	82.0	58.9	49.3
+ partial match [12]	54.8	23.5	82.4	60.3	51.2
Best (b=100) [12]	57.0	24.3	84.1	62.4	54.8
Best (a=400) [12]	64.9	24.1	91.0	71.9	58.4
CAC [25]	53.82	22.79	86.56	79.95	55.10
CLD [25]	55.60	24.30	86.43	81.95	58.02
SCD [25]	56.09	24.51	84.63	80.59	55.88
CCV [25]	54.34	22.43	86.25	78.63	52.28
CSD [25]	58.79	28.54	85.64	80.88	60.59
Best Unequal	59.78	29.49	90.30	86.96	62.87
Best Equal	59.49	27.53	89.90	87.00	61.46
Optimal N : θ=0.95	59.77	**29.61**	90.15	85.98	**63.48**
Optimal N : θ=0.98	**59.78**	29.21	90.04	86.04	62.76

Table 5. Reranking results on WebQueries when choosing query-relative N (optimal) using different threshold θ. Where a represents number of visual features, b number of context features in [12], Best Unequal means the best result when using unequal weight strategy, Best Equal when using equal weight strategy.

descriptor), SCD (scalable color descriptor) for reranking based on PRF method.

It can be observed from TABLE 5, the value of N becomes query specific. That is, we will adopt the optimal value of N for different queries. It can achieve the optimal reranking result. What's more, in some set of queries, it can achieve better performance than the best performance using equal and unequal weight strategy (as in Overall, LP, SEG).

4. Object/Scene Experiments: To investigate whether object and scene have different affects on image reranking, we manually label all queries (concepts) as either object or scene. TABLE 6 illustrates some examples of our manually labeled concepts (top ten concepts each) on the dataset of WebQueries. There are 299 concepts in the class of object and 54 concepts in the class of scene.

object concepts	scene concepts
”4x4” ”50 cent”	”crowd”
”airliner” ”al pacino”	“athletics track”
”amelie mauresmo”	”basilica saint peter”
”american flag”	”beach” ”cannes festival”
”amy winehouse”	”capitol” ”cemetary”

Table 6. Examples of our manual labeled concepts.

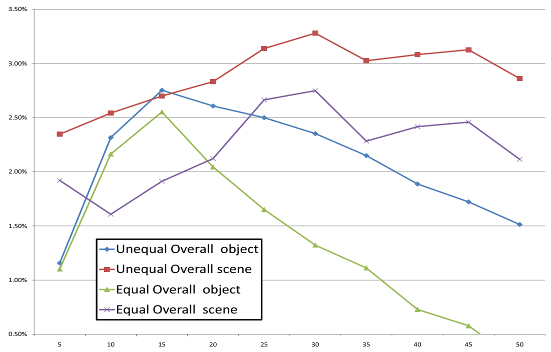

Figure 7. The mAP increment after reranking when N gets different values on the set of Overall.

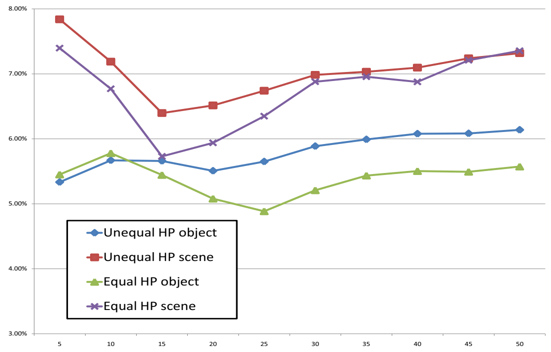

Figure 8. The mAP increment after reranking when N gets different values on the set of HP.

Fig.7,8,9 show all results in object and scene (Unequal means using unequal weight strategy, Equal means using equal weight strategy). Specifically, Fig. 7 shows the mAP increment after reranking when N get different values on the set of Overall queries. It can be observed that reranking on scene obtains better performance than that on object. Fig. 8 (HP) also shows the same result. However, on the set of LP (Fig. 9), all queries which are labeled as scene don't show better result after reranking. But after reranking, the queries belonging to object show better result than the initial rank list. It indicates that low precision (LP) query cannot be well modeled using the top N images. And, the queries belonging to object can be better modeled than those belonging to scene.

5. Conclusions

In this paper, we proposed an adaptive query-relative prototype modeling method for image search reranking. Firstly we randomly choose M images in the initial rank list for clustering to obtain a visual dictionary. Then we use

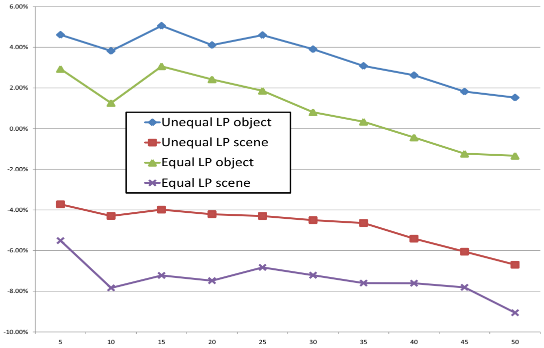

Figure 9. The mAP increment after reranking when N gets different values on the set of LP.

the top N images in the initial rank list to train a query relative model, which is composed of some words (described by the cluster center, cluster score, intra cluster distance). N can be determined by checking the incrementally obtained model on the top images is stable or not. Finally we use the model to rerank initial rank list returned by a specific image search engine.

Another contribution of the paper is that we present an adaptive method for choosing top N images so that we can obtain optimal reranking results. Also, we have relaxed the assumption of the top images having equal contribution to modeling the query and proposed a rank-relative, unequal weighting method. Experimental results on public dataset demonstrate the effectiveness of the method.

References

[1] N. Ben-Haim, B. Babenko, and S. Belongie. Improving web-based image search via content-based clustering. In *CVPR Workshop*, pages 106–106, 2006. 1, 2

[2] N. Ben-Haim, B. Babenko, and S. Belongie. Improving web-based image search via content based clustering. In *CVPR*, pages 1–6, 2006. 2

[3] D. Cai, X. He, Z. Li, W.-Y. Ma, and J.-R. Wen. Hierarchical clustering of www image search results using visual, textual and link information. In *ACM Multimedia*, pages 1–8, 2004. 1, 2

[4] M. Chi, P. Zhang, Y. Zhao, R. Feng, and X. Xue. Web image retrieval reranking with multi-view clustering. In *Proceedings of the 18th international conference on World wide web*, pages 1189–1190, 2009. 2

[5] J. Fan, Y. Shen, N. Zhou, and Y. Gao. Harvesting large-scale weakly-tagged image databases from the web. In *CVPR*, pages 802–809, 2010. 1, 2

[6] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *ICCV*, volume 2, pages 1816–1823, 2005. 1

[7] W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Video search reranking via information bottleneck principle. In *ACM Multimedia*, pages 1–10, 2006. 2

[8] W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Video search reranking through random walk over document-level context graph. In *ACM Multimedia*, pages 971–980, 2007. 2

[9] J. Huang, X. Yang, X. Fang, W. Lin, and R. Zhang. Integrating visual saliency and consistency for re-ranking image search results. *IEEE Trans. Multimedia*, 13(4):653–661, 2011. 1, 2

[10] K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *ACM Special Interest Group on Information Retrieval*, pages 41–48, 2000. 1, 4

[11] L. S. Kennedy and S.-F. Chang. A reranking approach for context-based concept fusion in video indexing and retrieval. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, CIVR '07, pages 333–340, 2007. 1

[12] J. Krapac, M. Allan, J. Verbeek, and F. Juried. Improving web image search results using query-relative classifiers. In *CVPR*, pages 1094–1101, 2010. 1, 2, 5, 6, 7

[13] L.-J. Li and L. Fei-Fei. OPTIMAL: automatic Online Picture collecTion via Incremental Model Learning. *IJCV*, 88(2):147–168, 2010. 1

[14] P. Li, L. Zhang, and J. Ma. Dual-ranking for web image retrieval. In *Proc. ACM Int'l Conf. on Image and Video Retrieval*, pages 166–173, 2010. 2

[15] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang. Tag ranking. In *Proceedings of the 18th international conference on World wide web*, pages 351–360. ACM, 2009. 1

[16] Y. Liu, T. Mei, and X.-S. Hua. CrowdReranking: Exploring multiple search engines for visual search reranking. In *ACM Special Interest Group on Information Retrieval*, pages 500–507, 2009. 2

[17] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, volume 2, pages 1150–1157, 1999. 2, 3

[18] B. McFee and G. Lanckriet. Metric learning to rank. In *ICML*, 2010. 1

[19] O.Chum and A. Zisserman. An examplar model for learning object classes. In *CVPR*, pages 1–8, 2007. 2

[20] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *ICCV*, pages 1–8, 2007. 1

[21] A. Smeulders, M. Worring, S.Santini, A. Gupta, and R. Jain. Content based image retrieval at the end of early years. *IEEE Trans. PAMI*, 22(12):1349–1380, 2000. 1

[22] X. Tian, Y. Lu, L. Yang, and Q. Tian. Learning from search engine and human supervision for web image search. In *ACM Multimedia*, pages 1365–1368, 2011. 2

[23] R. Wei, H. Lu, Y. Zheng, L. Cen, C. Jin, X. Xue, and W. Wu. How context helps: A discriminative codeword selection method for object detection. In *IEEE International Conference on Image Processing*, pages 3905–3908, 2010. 2

[24] K. Wnuk and S. Soatto. Filtering internet image search results towards keyword based category recognition. In *CVPR*, pages 1–8, 2008. 1

[25] R. Yan, A. Hauptmann, and R. Jin. Multimedia search with pseudo-relevance feedback. In *Proc. ACM Int'l Conf. on Image and Video Retrieval*, pages 238–247, 2003. 1, 2, 6, 7

[26] L. Yang and A. Hanjalic. Supervised reranking for web image search. In *ACM Multimedia*, pages 183–192, 2010. 1, 2, 4

[27] L. Yang and A. Hanjalic. Prototype-based image search reranking. *IEEE Trans. Multimedia*, 14(3):871–882, 2012. 2

[28] N. Zhou, Y. Shen, and J. Fan. Automatic image annotation by using relevant keywords extracted from auxiliary text documents. In *Proceedings of the international workshop on Very-large-scale multimedia corpus, mining and retrieval*, VLS-MCMR '10, pages 7–12, 2010. 1, 2