Learning to Judge Image Search Results *

Xinmei Tian †
University of Science and Technology of China
Hefei, Anhui, China 230027
xinmeitian@gmail.com

Yijuan Lu
Texas State University
San Marcos, TX 78666
yl12@txstate.edu

Linjun Yang
Microsoft Research Asia
Beijing, China 100190
linjuny@microsoft.com

Qi Tian
University of Texas at San Antonio
San Antonio, TX 78249
qitian@cs.utsa.edu

ABSTRACT

Given the explosive growth of the Web and the popularity of image sharing Web sites, image retrieval plays an increasingly important role in our daily lives. Search engines aim to provide beneficial image search results to users in response to queries. The quality of image search results depends on many factors: chosen search algorithms, ranking functions, indexing features, the base image database, *etc.* Applying different settings for these factors generates search result lists with varying levels of quality. Previous research has shown that no setting can always perform optimally for all queries. Therefore, given a set of search result lists generated by different settings, it is crucial to automatically determine which result list is the best in order to present it to users. This paper proposes a novel method to automatically identify the best search result list from a number of candidates. There are three main innovations in this paper. First, we propose a preference learning model to quantitatively study the best image search result identification problem. Second, we propose a set of valuable preference learning related features by exploring the visual characters of returned images. Third, our method shows promising potential in applications such as reranking ability assessment and optimal search engine selection. Experiments on two image search datasets show that our method achieves about 80% prediction accuracy for reranking ability assessment, and selects optimal search engine for about 70% queries correctly.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models

General Terms

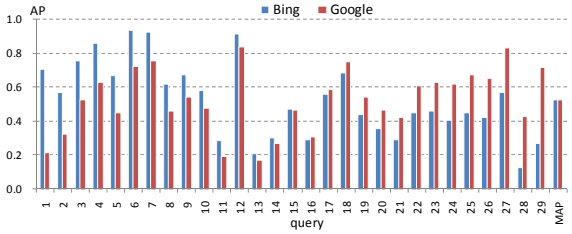
Algorithm, Experimentation, Performance

Keywords

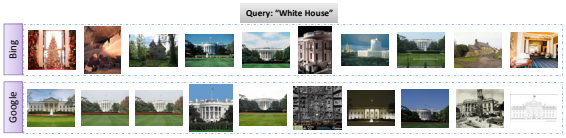
Image retrieval, search results performance comparison, reranking ability assessment

†This work was performed while the first author was a postdoctoral researcher at Texas State University.
*Area chair: Hari Sundaram

MM'11, November 28–December 1, 2011, Scottsdale, Arizona, USA.



(a) AP@40 on each query and MAP over all queries for Bing and Google (queries are sorted according to their AP difference for better view).



(b) The top-10 images returned on query "White House".

Figure 1: Image search result comparison between Bing and Google.

1. INTRODUCTION

Given the explosive growth of the Web and the popularity of image sharing Web sites, image retrieval plays an increasingly important role in our daily lives. Extensive research has been conducted to retrieve images relevant to a given query. Many factors can influence image search results. Existing work aims to get better search results by focussing their efforts on various aspects of the search process, such as designing effective visual features [15, 16], building efficient image indexes [17], developing new ranking algorithms [6, 22] and designing user-friendly interface [24]. The algorithms used in these aspects generate result lists of varying quality when used with different settings. Following are two examples for illustration.

In our first example we compare image search results generated by two popular search engines, Bing and Google. We submitted 29 text queries to them, and collected the images they returned[1]. Fig. 1(a) gives the AP@40 (average precision, ref Section 5.1 for details) for each query of the two search engines and the overall performance MAP (mean AP over all queries). We find that although Bing and Google have comparable MAP values (52.24 and 52.36 respectively), their performance on individual queries is quite different. Google achieves better performance on about half the queries. For example, on the query "White House", the top images returned by Google are more related to the "White House" than those returned by Bing (Fig. 1(b)). If

[1]The data was collected in 2008.

Table 1: MAP@40 (×100) of Bing, Google, and after optimal search engine selection for each query.

	Bing	Google	$\text{Select}_{\text{Opt}}$
MAP@40	52.24	52.36	*60.71*

Table 2: MAP@20 (×100) of the text-based search (Text) and the two reranking methods, PRF and BR.

	Text	PRF	BR
MAP@20	50.28	60.65	63.64
Gain	-	20.62%	26.57%

an algorithm could automatically determine which search engine would generate a better result list for each query, one could achieve better performance by selecting the optimal search engine for each query. Table 1 shows the MAP value after this selection, which is 60.71, about 16% relative improvement over Bing and Google.

In our second example we compare the performance of text-based image search and visual reranking. Most existing image search engines are implemented by indexing and searching textual information associated with images, *e.g.*, surrounding text, URLs. The text-based image search approach is efficient for large scale image databases. However, it suffers when the associated text is incapable of adequately describing the image. To address this difficulty, visual reranking has been developed to refine the text search results by incorporating visual information from images. Recent research has shown that visual reranking can generally improve the performance of text-based image search to some extent [19, 22]. However, it is not guaranteed to benefit every query. It is widely observed that visual reranking can greatly improve retrieval performance for some queries, while for others reranking can even degrade the performance of the initial text-based search.

As an illustration, we apply two popular reranking methods, BR [19] and PRF [21], on a public image search dataset (Web353). This dataset was collected by Krapac *et al.* [13]. It contains 71478 images returned by a Web search engine for 353 general textual queries. Table 2 presents the average performance of the text-based search engine (Text) and the performance of the two reranking methods, in terms of MAP@20 over 353 queries. Table 3 lists the number of queries with improved, degraded, or equivalent performance after reranking. We find that, although overall performance of all 353 queries is improved, there are still around 100 queries (25 - 30 percent) that suffer performance decrease after reranking. By further investigating the reranking performance on each query, we find that the performance of many queries has decreased significantly. For some queries, the decrease in AP value is as great as 0.7. Thus, given a query, it becomes crucial for the search engine to predict its visual reranking performance and decide whether the visual reranking process should be performed or not. Doing so would allow us to avoid presenting reranking results which are even worse than text-based search results to users.

The above two examples raise the same problem: for a query, given a set of result lists, which one is the best (has the highest retrieval performance)? In other words, which result list should be presented to users? This paper aims to solve this problem: *given a set of search result lists returned by multiple search executions of a query, how can we design an algorithm to automatically compare the quality of those result lists in order to identify the result list with the highest performance.* To solve this problem, we build a model to investigate the quality of search results using machine learning. It consists of two stages: training and testing. In the training stage we explore the visual distribution characteristics of good and bad search result lists and derive a set of light-weight features to capture their differences. Then, by forming the search result lists of training queries into preference pairs, we derive a preference learning model (PLM) by training on these pairs with RankSVM [12]. Finally, in the testing stage, the developed PLM is applied to predict the preference score for search result lists of any testing query.

Table 3: The number of queries with improved, degraded, or equivalent performance after reranking.

♯queries	Improved	Unchanged	Degraded
PRF vs. Text	237	6	110
BR vs. Text	256	10	87

To the best of our knowledge this is the first attempt that automatically evaluates the quality of Web image search result lists. The proposed approach has a wide range of applications. For example, it is capable of selecting the best search engine to solve the problem in example 1 and automatically determining whether reranking can benefit the query to solve the problem in example 2. There are also many other promising potential applications. For example, given different search algorithm settings (various visual features, visual reranking methods, *etc.*), our approach can automatically select the optimum settings for each query.

The main contributions introduced in this paper are summarized as follows:

- We quantitatively study and formulate the image search result preference learning problem. We propose a novel framework and a set of valuable features to automatically compare the quality of image search result lists.
- Our proposed approach shows promising application potential for optimal search engine selection, merging of search result lists, and selecting the best visual feature and reranking approach for each individual query.
- Our work will explicitly guide the research in visual reranking ability estimation and provide a path for query difficulty modeling.

2. RELATED WORK

Image search plays an important role in our daily lives. Considerable research has been proposed to improve image search from various aspects, such as image annotation [2] and visual reranking [19, 21, 22]. All these research efforts have the same objective of returning good image search results to users. Different search result lists are generated by different image search methods and their performance on each query varies greatly. These works show their strength on certain aspects. There is no single method which can always work the best for all queries. Therefore, in addition to developing (overall) effective search approaches, it is also very important to select the most suitable search method for each query. Through this selection, better image search results can be derived. This paper conducts this best method selection for each query by investigating the quality of the image search result lists generated by different search methods. The search result list with the highest performance is picked out and presented to users.

The most related work to this paper is the query difficulty prediction. Query difficulty prediction in document retrieval has been explored for many years [3, 7, 8, 9, 11, 14, 23]. It aims to predict whether a query will have a high retrieval performance in a document collection. It includes two categories, pre-retrieval prediction and post-retrieval prediction. In pre-retrieval, query difficulty prediction attempts to evaluate search performance before the retrieval step [8, 9, 14]. It mainly relies on statistics of query terms over document collections. He and Ounis [8] proposed several pre-retrieval predictors by considering the intrinsic statistical features of queries, including query length, standard deviation of the inverse document frequency (idf) of terms in the query, query scope, and a simplified clarity score (SCS). Kwok *et al.* [14] employed support vector regression to train a query difficulty prediction model with simple features such as log document frequency and query term frequency. Imran and Sharan [9] proposed two pre-retrieval query difficulty predictors based on the co-occurrence information among query terms. They assumed that higher co-occurrence of query terms means more information is conveyed, which leads to an easier query or a lower query difficulty level.

In post-retrieval prediction, the retrieval step is conducted first and query difficulty prediction evaluates the performance of the returned results. Our work is most related to it. In [3], a clarity score was proposed that measures the ambiguity of a query through the Kullback-Leibler divergence between the language models created from top-retrieved documents and all documents in the collection. Hauff *et al.* [7] proposed an improved clarity score to solve the parameter sensitivity problem of the previous one. Elad Yom-Tov *et al.* [23] estimated the quality of search result lists by measuring the agreement between the top results returned by the full query and its sub-queries. Jensen *et al.* [11] predicted query difficulty by using features extracted from surrogate documents represented in the search result list to train a regression model. In image retrieval, little research has been conducted on query difficulty estimation. Xing *et al.* [20] used textual features to predict whether a query is difficult to be represented by images or not. This work does not investigate the image search performance, but only classifies the queries into two categories "easy" or "hard".

Query difficulty prediction estimates the performance of a search result list for a given query. Unlike query difficulty prediction, our work targets comparing several search result lists generated for a particular query. Instead of predicting their exact performance, we only need to know which search result list is better than the others. Furthermore, in query difficulty prediction, the search result lists are independent of each other since they are generated for different queries. In our problem, the compared search result lists are generated for the same query and are thus correlated. We can utilize the correlation between them. Additionally, both the query and documents in the query difficulty prediction problem are in the textual domain. In our problem queries are textual and images are visual, creating a more complex problem. Our image search result performance comparison problem faces many challenges. As a first attempt, this paper only focuses on exploiting visual information, which is the essential description of images. In the case where textual information of images (URL, surrounding texts *et al.*) is also available, we will further exploit the joint usage of textual and visual information for this problem in the future.

3. PROBLEM FORMULATION

For query q and an image collection $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, multiple search result lists can be derived using different search algorithms. Each search result list is a permutation/ranking of the N images sorted in descending order by their ranking scores, which are generated by the search algorithm . We use ranking list variable l to denote a search result list. Assuming there are n_q ranking lists generated for query q, they constitute a set of search result lists $\mathcal{L}^{(q)} = \{l_1, \cdots, l_{n_q}\}$. Our objective is to automatically determine which l in $\mathcal{L}^{(q)}$ has the highest performance,

$$l^* = \text{argmax}_{l \in \mathcal{L}^{(q)}} \ \ y(l), \tag{1}$$

where $y(l)$ denotes the performance of l. $y(l)$ can be measured by commonly used information retrieval measures, such as precision, recall, Average Precision (AP) [1] and Normalized Discounted Cumulated Gain (NDCG) [10].

For two ranking lists, the one with more relevant images ranked at the top gives a better performance than the one with fewer relevant images ranked at the top. If we have the ground truth label of each image (its relevance to query q), then $y(l)$ can be derived by using AP or NDCG and the best search result selection in problem (1) is straight forward. However, in real applications, the ground truth relevance labels for images are unavailable. In this situation, how can we know which ranking list performs better? In this paper, we propose to solve this problem via machine learning. Specifically, we want to learn a preference model $f(l) = \mathbf{w}^{\mathrm{T}}\psi(l)$ from a training set, where $\mathbf{w}$ is the weighting coefficient vector and $\psi(l)$ is a vector which reflects the characteristics of l. This model should satisfy the following constraints on the training set

$$\forall (l_i, l_j), \text{if } y(l_i) > y(l_j), \text{ then } f(l_i) > f(l_j). \tag{2}$$

For two ranking lists l_i and l_j in training set, if the ground truth performance $y(l_i)$ is better than $y(l_j)$ (l_i is preferred to l_j), $f(l_i)$ should be larger than $f(l_j)$. In other words, the ordinal relationship of pair $(f(l_i), f(l_j))$ must be consistent with that of $(y(l_i), y(l_j))$, to reflect the ground truth preference of two ranking lists.

In this paper we formulate the learning problem of $f(\cdot)$ by using the powerful RankSVM [12] algorithm. It minimizes the prediction errors on a set of training queries $\mathcal{Q} = \{q^{(1)}, \cdots, q^{(m)}\}$,

$$\begin{aligned} \min \ & \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum \xi_{ijk} \\ s.t. \ & \forall k, k = 1, \cdots, m. \ \forall (l_i, l_j) \in \mathcal{S}^{(q^{(k)})}, \\ & \mathbf{w}^{\mathrm{T}}\psi(l_i) \geq \mathbf{w}^{\mathrm{T}}\psi(l_j) + 1 - \xi_{ijk}, \ \xi_{ijk} \geq 0 \end{aligned} \tag{3}$$

where ξ is the slack variable and $C > 0$ controls the trade-off between model complexity and training errors. $\mathcal{S}^{(q)}$ is the set of preference ranking list pairs for query q generated from the ranking list set $\mathcal{L}^{(q)} = \{l_1, \cdots, l_{n_q}\}$

$$\mathcal{S}^{(q)} = \{(l_i, l_j) | y(l_i) > y(l_j); i, j = 1, \cdots, n_q\}. \tag{4}$$

The preference learning model $f(\cdot)$ can be derived by solving problem (3). Then, this model can be applied to any testing query q' for which ground truth relevance labels are unavailable. Suppose there are $n_{q'}$ ranking lists generated for this query, $\mathcal{L}^{(q')} = \{l_1, \cdots, l_{n_{q'}}\}$. $f(\cdot)$ can predict a value for each list. For any two ranking lists l_i and l_j, if $f(l_i) > f(l_j)$, we know that l_i performs better than l_j, and *vice versa.* The ranking list with the highest prediction value is the one which has the best performance.

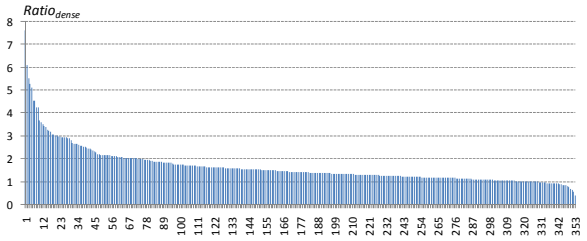


Figure 2: Sorted $Ratio_{dense}$ values in 353 queries.

4. FEATURE CONSTRUCTION

A crucial factor in $f(l) = \mathbf{w}^{\mathrm{T}}\psi(l)$ is the vector $\psi(l)$. It is not trivial to design a feature vector to capture visual characteristics of an arbitrary ranking list l. By analyzing the visual distribution of images in the collection, we propose a set of lightweight features.

4.1 Two Basic Assumptions

Given two ranking lists returned for query q over image collection $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, the key is to investigate the visual difference between relevant and irrelevant images. The relative feature vector $\psi(l)$ discussed in this paper is designed based on the following two basic assumptions:

- *Density Assumption*: Relevant images have higher density than irrelevant images;
- *Visual Similarity Assumption*: Relevant-relevant image pairs share higher visual similarity than relevant-irrelevant and irrelevant-irrelevant image pairs.

4.1.1 Density Assumption

Our density assumption is that relevant images have higher density than irrelevant images. To verify whether this assumption is true or not, we calculate the density of each of the N images in query q and then analyze their statistic characteristics. The density $p_{\mathbf{x}_i}$ for image $\mathbf{x}_i$ is calculated via Kernel Density Estimation (KDE)[18],

$$p_{\mathbf{x}_i} = \frac{1}{|\mathcal{N}(\mathbf{x}_i)|} \sum\nolimits_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)} k(\mathbf{x}_i - \mathbf{x}_j), \tag{5}$$

where $\mathcal{N}(\mathbf{x}_i)$ is the set of neighbors of image $\mathbf{x}_i$ among the N images and $k(\mathbf{x})$ is a kernel function that satisfies both $k(x) > 0$ and $\int k(\mathbf{x})d(\mathbf{x}) = 1$. The Gaussian kernel is adopted in this paper and σ is empirically set as the average of pair-wise distances of all images.

Without ambiguity, we use $\mathbf{x}_i$ to denote both the image and its visual feature vector in this paper. Various visual features can be used in (5). In this paper we adopt the popular visual bag-of-word image representation. More details will be introduced in Section 5.1.

To show the density difference between relevant and irrelevant images, we calculate the average density of all relevant images $AvgDense_+$ and the average density of all irrelevant images $AvgDense_-$ in each query. They are calculated as,

$$AvgDense_+ = \frac{1}{|\mathcal{X}^+|} \sum\nolimits_{\mathbf{x}_i \in \mathcal{X}^+} p_{\mathbf{x}_i}, \tag{6}$$

$$AvgDense_- = \frac{1}{|\mathcal{X}^-|} \sum\nolimits_{\mathbf{x}_i \in \mathcal{X}^-} p_{\mathbf{x}_i}, \tag{7}$$

where $\mathcal{X}^+$ is the set of all relevant images and $\mathcal{X}^-$ is the set of all irrelevant images. $Ratio_{dense}$ is defined as, $Ratio_{dense} = AvgDense_+ / AvgDense_-$.

We compute $Ratio_{dense}$ for all 353 queries in Web353 and plot them in Fig. 2 by sorting them in descending order.

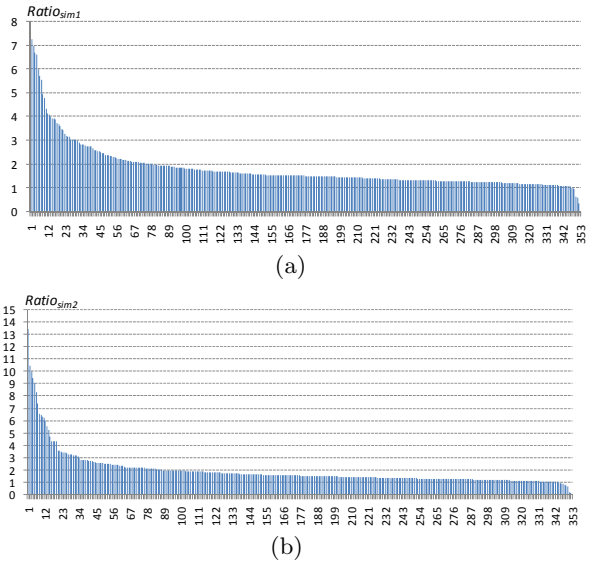


Figure 3: Sorted $Ratio_{sim1}$ (a) and $Ratio_{sim2}$ (b) on 353 queries.

From Fig. 2, we see that, among 353 queries, there are 329 queries whose average density of relevant images is larger than the average density of irrelevant images ($Ratio_{dense} > 1$). To verify whether $AvgDense_+$ is significantly larger than $AvgDense_-$, we further perform a statistical significance test. We used the T-test with a 5% level of significance. The T-test result shows that in 286 queries the average density of relevant images is significantly larger than the average density of irrelevant images. This phenomenon demonstrates that the density assumption holds for most queries.

4.1.2 Visual Similarity Assumption

Our visual similarity assumption is that relevant-relevant image pairs share higher visual similarity than relevant-irrelevant and irrelevant-irrelevant image pairs. For query q, we calculate the visual similarity $sim(\mathbf{x}_i, \mathbf{x}_j)$ for any image pair $(\mathbf{x}_i, \mathbf{x}_j)$. There are various ways to calculate $sim(\mathbf{x}_i, \mathbf{x}_j)$. We use the popular bag-of-visual words representation with intersection kernel [5].

To verify the visual similarity assumption we calculate the average similarity of relevant-relevant, relevant-irrelevant, and irrelevant-irrelevant image pairs for each query in the Web353 dataset. They are denoted as $AvgSim_{++}$, $AvgSim_{+-}$, and $AvgSim_{--}$ respectively. We plot the sorted $Ratio_{sim1} = \frac{AvgSim_{++}}{AvgSim_{+-}}$ and $Ratio_{sim2} = \frac{AvgSim_{++}}{AvgSim_{--}}$ in Fig. 3. It shows that, among 353 queries, there are more than 340 queries whose average similarity of relevant-relevant pairs is larger than the average similarity of relevant-irrelevant and irrelevant-irrelevant pairs. The statistical significance test (T-test with a 5% level of significance) reveals that $AvgSim_{++}$ is significantly larger than $AvgSim_{+-}$ in 347 queries, and $AvgSim_{++}$ is significantly larger than $AvgSim_{--}$ in 339 queries. This proves the validity of our visual similarity assumption.

Due to the well-known semantic gap problem, some queries (especially the queries with large intra-class appearance variance) are hard to be well represented by descriptive visual features. That is the reason why our two assumptions fail for some queries, as shown in Figs. 2 and 3. By further investigating the two assumptions on each query, we find that the assumptions are valid for the queries with small appearance variance, such as "pantheon rome", "flag Italy", "mona lisa",

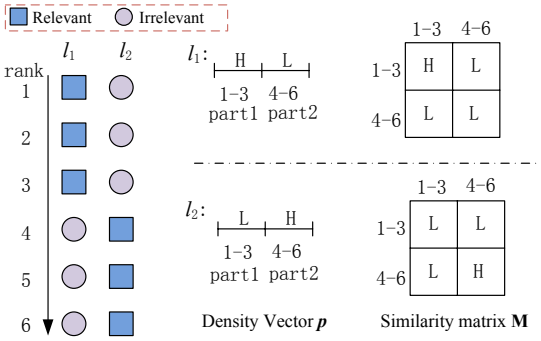


Figure 4: Illustration of density and similarity distribution difference between two ranking lists.

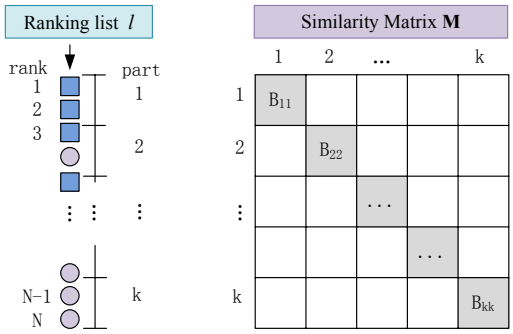


Figure 5: The N images are split into k parts along their ranks equally. Therefore, the $N \times N$ similarity matrix $\mathbf{M}$ is split into $k \times k$ blocks.

"log NBA", *etc.* They are likely to fail for the queries with large appearance variance, such as "flower", "dog", *etc.* Although the assumptions fail for some queries, they are valid for a majority of queries (Figs. 2 and 3). Therefore, it is reasonable to apply them in our method. Our experimental results reported in Section 5 also validate this.

4.2 Preference Learning Feature Extraction

Inspired by the above two assumptions, we propose a set of related features by mining the distribution of density and visual similarity in l. We demonstrate it by using a toy example for illustration, as shown in Fig. 4. Suppose there are 6 images returned for query q, 3 relevant (denoted by square) and 3 irrelevant (denoted by circle). Given the two ranking lists l_1 and l_2, obviously the performance of l_1 is better than l_2, *i.e.* $y(l_1) > y(l_2)$. According to the two assumptions, for the better ranking list l_1, its top ranked images should have high (H) density and share high visual similarity while bottom ranked images should have low (L) density and low visual similarity. This density and similarity distribution difference between the two ranking lists can be utilized for extracting preference learning related features.

4.2.1 Similarity Distribution Feature

For query q, given a ranking result l, a visual similarity matrix $\mathbf{M} \in \mathbb{R}^{N \times N}$ can be obtained by calculating pairwise image similarity. The (i, j) element m_{ij} in $\mathbf{M}$ denotes the visual similarity between the i-th ranked image and j-th ranked image. We split the N images into k groups along their ranks equally. As a consequence, the $N \times N$ similarity matrix $\mathbf{M}$ is split into $k \times k$ grids, as shown in Fig. 5. Then, we analyze the sub similarity matrix in diagonal blocks. Specifically, we calculate the mean and variance of similarities in each block to derive the similarity distribution feature vector F_{SD}:

$$F_{SD}(i) = [mean(\mathbf{M}^{(i,i)}), var(\mathbf{M}^{(i,i)})], i = 1, \cdots, k. \quad (8)$$

where $\mathbf{M}^{(i,i)}$ is the sub similarity matrix in block B_{ii}. Then, a $2k$-dimensional similarity distribution feature vector F_{SD} is derived. The intuition behind this feature is that, for a good ranking result list, more relevant images have higher ranks. In other words, images belonging to the top part may share higher similarity than those in other parts.

4.2.2 Density Distribution Feature

Similar to the visual similarity distribution feature, we also propose a density distribution feature based on the density assumption. For the N images $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, we can derive a density vector $\mathbf{p} = [p_1, \cdots, p_N]^{\mathrm{T}}$, where p_i is the density of the i-th ranked image in l as defined in (5). We also split the N images into k groups and calculate the mean and variance of the density of the images in each part,

$$F_{DD}(i) = [mean(\mathbf{p}^{(i)}), var(\mathbf{p}^{(i)})], i = 1, \cdots, k. \quad (9)$$

where $\mathbf{p}^{(i)}$ is the sub density vector for images in part i. By concatenating $F_{DD}(i)$, $i = 1, \cdots, k$, we can get a $2k$-dimensional density distribution feature vector F_{DD}.

4.2.3 Feature from Top-T Ranked Images

Both F_{SD} and F_{DD} roughly capture the overall density and visual similarity distribution of all N images in l (mean and variance). The following features are designed to exploit them in fine granularity as a complementation. Especially in the case when users only focus on the performance of images ranked in the first several pages. Therefore, we propose a histogram of density and visual similarity to elaborately analyze the top-T ranked images in l.

Specifically, the density value is in range [0, 1] and we equally divide it into C-bins. Then, the densities of top-T ranked images $\{p_1, p_2, \cdots, p_T\}$ can be quantified into a C-bin histogram by mapping them into the corresponding bins. We denote this density histogram feature as F_{HD},

$$F_{HD}(c) = \frac{1}{T}|\{i|i = 1, \cdots, T, p_i \in c\text{-th bin}\}|, \quad (10)$$

where $c = 1, \cdots, C$.

Similarly, we can get a C-bin visual similarity histogram F_{HS} by mapping the $T \times T$ similarity matrix of the top-T ranked images into C-bins,

$$F_{HS}(c) = \frac{1}{T^2}|\{(i,j)|i,j = 1, \cdots, T, m_{ij} \in c\text{-th bin}\}|, \quad (11)$$

where $c = 1, \cdots, C$.

Given F_{SD}, F_{DD}, F_{HD}, and F_{HS}, the final preference learning feature vector $\psi(l)$ can be derived by concatenating these four individual features.

4.3 Training Sample Enlargement

We have shown how to extract the preference learning feature vector $\psi(l)$ for a ranking list l. We then build the preference learning model by training RankSVM on a set of queries $\{q^{(1)}, \cdots, q^{(m)}\}$, as presented in problem (3). In the training set, for each query, there are usually only a few ranking lists in $\mathcal{L}^{(q)}$, which may cause the small (insufficient) sample problem. For example, in our example 1 in Section 1, there are only two ranking lists (one from Bing and the other from Google). To solve this problem, we can construct additional ranking lists to enlarge the training set.

We can manually create ranking list l^{manual} for each query by permutating the N images in query q according to certain rules. Then, l^{manual} is added into the ranking list set $\mathcal{L}^{(q)}$:

$$\mathcal{L}^{(q)} \leftarrow \mathcal{L}^{(q)} \cup \{l^{manual}\} \quad (12)$$

In this paper, we create three manual ranking lists for each query, including:

1) Perfect ranking list: order all relevant images at the top and all irrelevant images at the bottom;

2) Worst ranking list: order all irrelevant images at the top and all relevant images at the bottom;

3) Random ranking list: permutate the images randomly.

Our experiments show that this training set enlargement works well for solving the small training sample problem.

5. EXPERIMENTS

5.1 Reranking Ability Assessment

In this section, we investigate the effectiveness of the proposed preference learning model (PLM) by applying it to reranking ability assessment. In reranking, each query q has two ranking lists: l_{Text} generated by text-based search engine and l_{rerank} generated by the reranking process. The *reranking ability* t_q^* is defined as the performance improvement of reranking over text-based search, $t_q^* = y(l_{rerank}) - y(l_{Text})$. The reranking ability measures to what degree reranking can improve text-based search results. For a query, if its reranking ability is positive (suitable to be reranked), the reranking result list will be presented to users; otherwise the text-based search result list will be presented. In other words, the search engine can achieve guaranteed performance enhancement by only reranking queries which are suitable for reranking while leaving the remaining unsuitable ones unchanged. With this motivation, we apply PLM to assess reranking ability. Specifically, with the model $f(\cdot)$, PLM can predict a value for l_{Text} and l_{rerank} respectively. The prediction difference $f(l_{rerank}) - f(l_{Text})$ is used to approximate the ground truth reranking ability t_q^*.

Dataset: In order to demonstrate the capacity of PLM for reranking ability assessment, we conduct experiments on a large public web image search dataset "Web353", collected by Krapac *et al.* [13]. This dataset consists of 71478 images returned by the French search engine Exalead [2] for 353 search queries, which were sampled from the most frequent terms searched by Exalead users. These 353 queries are very diverse and cover a broad range of topics, including landmark, design (painting, map, logo, flag), people (movie, sports, singer star), object (vehicle, instrument, building, sports tool), and others (animal, plant, product, place, event, abstract word). Queries are somewhat evenly distributed across these topics. For each query, there are about 200 images returned by Exalead. The ground-truth relevance label for each image is given a binary value: "relevant" or "irrelevant". In this dataset, there are 43.86% images labeled as relevant. For each query, we conduct BR (Bayesian Reranking) [19] to generate its reranking result l_{rerank}.

Ranking List Performance $y(l)$: For query q, given a ranking result list l, its ground truth performance $y(l)$ is measured via non-interpolated average precision (AP) [1], which is widely used in information retrieval. AP is the mean of the precision values obtained when each relevant image occurs. The AP of top-T ranked images is defined as

$$\text{AP@}T = \frac{1}{Z_T}\sum_{i=1}^{T}[precison(i) \times rel(i)] \quad (13)$$

where $precison(i)$ is the precision of top-i ranked images and $rel(i)$ is a binary function denoting the relevance of the ranked image with "1" for relevant and "0" for irrelevant. Z_T is a normalization constant which is chosen to guarantee AP@$T = 1$ for a perfect ranking result list.

Model training: We use the leave-one-out method for PLM training. At each step we train the model on 352 queries and test this model on the leftover query. We repeat the process 353 times to ensure that each query has been used as test query at least once.

Image visual representation: The density and visual similarity features described in Section 4 are calculated based on visual representation of images. In this paper we use a bag-of-visual word histogram to visually represent an image. Scale-invariant feature transform (SIFT) [16] local descriptors are extracted from each image on a dense grid. Then, a codebook is generated by clustering all local descriptors into 1000 groups [5]. By quantizing local descriptors into visual words, each image is represented as a 1000-dimensional histogram. Spatial pyramid matching [15] is used to encode spatial information. Calculating the similarity between two histograms is done using an intersection kernel.

Evaluation: For each query q, there are two ranking lists: l_{Text} and l_{rerank}. The query's ground truth reranking ability is $t_q^* = y(l_{rerank}) - y(l_{Text})$, and the reranking ability estimated by the learned PLM is $t_q = f(l_{rerank}) - f(l_{Text})$. We evaluate PLM from the following two aspects.

1. *Prediction Accuracy* (AC):

$$\text{AC} = \frac{\sharp\text{Correctly predicted queries}}{\sharp\text{Total queries}} \quad (14)$$

Correctly predicted queries are those which satisfy $t_q^* t_q > 0$. AC examines whether PLM can correctly predict the binary relationship (improved or not) between the result lists of reranking and text-based search. In addition to this overall accuracy, we also examine the prediction accuracy P+, P− of positive and negative queries. A positive (negative) query is one in which reranking performs better (worse) than the text-based search, *i.e.,* $t_q^* > 0$ ($t_q^* < 0$). P+ and P− are defined as:

$$\text{P+} = \frac{\sharp\text{Correctly predicted positive queries}(t_q^* > 0 \text{and} t_q > 0)}{\sharp\text{Total positive queries}}$$

$$\text{P−} = \frac{\sharp\text{Correctly predicted negative queries}(t_q^* < 0 \text{and} t_q < 0)}{\sharp\text{Total negative queries}}$$

We examine P+ and P− because we want to investigate the model's capacity for negative query detection as well as the percentage of sacrificed positive queries.

2. *Correlation Coefficient*: Accuracy only measures the binary prediction of reranking ability, *i.e.*, improved or not after reranking. To further verify the effectiveness of PLM in terms of reranking ability degree prediction, we check the consistency between the ground truth reranking ability vector $\mathbf{t}^* = [t_{q(1)}^*, \cdots, t_{q(353)}^*]^{\mathrm{T}}$ and the one predicted by PLM $\mathbf{t} = [t_{q(1)}, \cdots, t_{q(353)}]^{\mathrm{T}}$. As widely used in query difficulty prediction [11, 23], we calculate the Kendall's τ rank correlation coefficient between $\mathbf{t}^*$ and $\mathbf{t}$. Kendall's τ is defined as $\tau = \frac{n_c - n_d}{n_c + n_d}$, where n_c and n_d are the numbers of concordant and discordant pairs respectively. A pair of two queries $(q^{(i)}, q^{(j)})$ is concordant if the orders of $(t_{q^{(i)}}^*, t_{q^{(j)}}^*)$ and $(t_{q^{(i)}}, t_{q^{(j)}})$ agree. Kendall's τ value falls within the range [-1, 1], where -1 means perfect negative correlation, 1

[2] http://www.exalead.com/search/image

Table 4: Correlation coefficients and accuracy in *reranking ability assessment*.

		Kendall's τ	AC(%)	P+(%)	P-(%)
T=20	QD	0.0543	57.51	67.97	33.33
	PLM	0.3654	75.64	92.19	35.63
T=40	QD	0.1485	65.44	77.44	30.49
	PLM	0.4414	78.75	91.35	42.68
T=60	QD	0.0826	65.44	77.09	26.03
	PLM	0.4782	82.44	92.73	49.32
T=80	QD	0.1244	68.84	81.09	27.40
	PLM	0.4820	80.45	89.82	50.68
T=100	QD	0.1400	70.82	82.44	28.99
	PLM	0.4659	80.74	89.25	52.17

Table 5: MAP (×100) comparison in *reranking ability assessment*. MAP is the mean of AP over all queries.

	Text	BR	$\text{Select}_{\text{QD}}$	$\text{Select}_{\text{PLM}}$	$\text{Select}_{\text{Opt}}$
MAP@20	50.28	63.64	59.65	**64.32**	*66.80*
MAP@40	45.24	57.37	55.29	**57.97**	*59.40*
MAP@60	43.06	54.35	51.89	**55.02**	*55.96*
MAP@80	42.60	52.90	51.24	**53.45**	*54.32*
MAP@100	43.08	52.99	51.65	**53.38**	*54.24*

means perfect positive correlation, and 0 means $\mathbf{t}^*$ and $\mathbf{t}$ are independent to each other.

Reranking Ability Assessment: We evaluate PLM at 5 truncation levels, *i.e.*, $y(l) = \text{AP@}T, T = \{20, 40, 60, 80, 100\}$. We implement the document query difficulty method proposed in [11] as a baseline, since this method conducts query difficulty prediction through supervised model training. Based on [11], we extract textual features for each image from its associated textual information (URL, surrounding text, *etc.*) and train a regression model. The reranking ability is denoted as the query difficulty difference between the two ranking lists (l_{Text} and l_{rerank}). We note the method in [11] as QD. Table 4 shows the Kendall's τ correlation coefficients and accuracy of our approach and the baseline QD. It reveals that our method outperforms QD in both correlation coefficients and accuracy. By further investigating P+ and P-, we conclude that PLM removes about half of the negative queries while keeping most positive queries. For example, in T=80, PLM detects 50.68% of the negative queries, preventing performance decrease, while sacrificing performance gain on only 10.18% of the positive queries.

With the predicted reranking ability, we can choose to execute reranking only on those queries whose reranking ability is positive. This operation can prevent large performance decreases on some queries, possibly improving the user experience. The reranking process selection for each query can also lead to a better overall performance (mean AP over all queries, MAP). We select a better one from l_{Text} and l_{Rerank} for each query via QD ($\text{Select}_{\text{QD}}$) and PLM ($\text{Select}_{\text{PLM}}$.) Table 5 lists the MAP values for text-based search (Text), reranking (BR), $\text{Select}_{\text{QD}}$, and $\text{Select}_{\text{PLM}}$. Column $\text{Select}_{\text{Opt}}$ shows the maximal MAP by selecting the best search result list between Text and BR for each query according to their ground truth performance, which gives the upper bound of the MAP value we can achieve. Table 5 shows that $\text{Select}_{\text{PLM}}$ performs better than both Text and BR, while $\text{Select}_{\text{QD}}$ only achieves a moderate performance between Text and BR. The reason why the QD method in [11] does not work well here is that textual features in image retrieval are not the essential descriptions for the images, therefore more noise (*e.g.*, mismatching between surrounding text and image content) may be introduced.

Table 6: MAP (×100) comparison in reranking ability assessment for the queries which do not satisfy the assumptions.

	Text	BR	$\text{Select}_{\text{PLM}}$	$\text{Select}_{\text{Opt}}$
MAP@20	41.00	**41.29**	40.78	*47.51*
MAP@40	36.57	36.91	**37.20**	*41.09*
MAP@60	34.58	35.38	**35.46**	*38.37*
MAP@80	35.32	36.15	**36.16**	*38.92*
MAP@100	36.38	**37.90**	37.50	*40.19*

Table 7: MAP (×100) comparison in reranking ability assessment for each of the 5 query categories (T=60).

	Text	BR	$\text{Select}_{\text{PLM}}$	$\text{Select}_{\text{Opt}}$
landmark	50.92	62.44	**63.75**	*64.88*
design	41.94	57.83	**58.59**	*59.00*
people	47.08	**61.61**	61.59	*62.08*
object	33.66	35.33	**36.64**	*39.00*
others	40.38	50.70	**51.31**	*52.10*

As we discussed in Section 4.1, the assumptions are not always valid for all queries. For those queries which do not satisfy the assumptions, the extracted preference learning features may be noisy. Consequently the preference prediction may be unreliable. To investigate the implication of the failures of the assumptions on the prediction results, we examined the performance of the proposed PLM on the 67 queries which do not satisfy either the density assumption or the visual similarity assumption. The experimental results show that our approach still achieves (50±4.9)% prediction accuracy (AC over $T = 20, 40, 60, 80, 100$) in reranking ability assessment, which is close to random prediction. And the $\text{Select}_{\text{PLM}}$ is comparable to Text and BR, as shown in Table 6. It reveals that our method does not worsen the search engine's performance even on the queries which do not satisfy the assumptions.

To further investigate the effectiveness of PLM for different categories of queries, the 353 queries are grouped into 5 categories: landmark (53), design (60), people (98), object (54) and others (88). We conducted experiments on each of the 5 categories. The experimental results show that our method works well on all query categories. The prediction accuracy (ACs) are 86.79%, 95.00%, 87.76%, 61.11%, 78.41% respectively. Even in the category "object", of which queries usually have images with a large visual appearance variance, moderate AC (61.11%) is obtained and the MAP value of $\text{Select}_{\text{PLM}}$ is better than both Text and BR, as shown in Table 7. Better performance is achieved in "landmark" and "design" since the images of those categories are more visually consistent. For "people", high AC is obtained, but the $\text{Select}_{\text{PLM}}$ is very close to BR. The reason is that BR already improves most of queries in this category a lot, hence the improvement space of PLM over BR is limited.

Reranking Feature Selection: PLM can also be applied to select optimal reranking features. The reranking

Table 8: MAP (×100) comparison in *reranking feature selection* from {Text, BR_{SIFT} and BR_{CF}}

	Text	BR_{SIFT}	BR_{CF}	$\text{Select}_{\text{Random}}$	$\text{Select}_{\text{PLM}}$	$\text{Select}_{\text{Opt}}$
MAP@20	50.28	63.64	60.21	58.04	**64.74**	*69.13*
MAP@40	45.24	57.37	53.99	52.20	**58.44**	*61.35*
MAP@60	43.06	54.35	50.65	49.35	**55.17**	*57.34*
MAP@80	42.60	52.90	49.20	48.23	**53.85**	*55.58*
MAP@100	43.08	52.99	49.08	48.38	**53.67**	*55.36*

Table 9: MAP (×100) comparison in *reranking algorithm selection* from {Text, BR_{SIFT} and PRF_{SIFT}}.

	Text	BR_{SIFT}	PRF_{SIFT}	$\text{Select}_{\text{Random}}$	$\text{Select}_{\text{PLM}}$	$\text{Select}_{\text{Opt}}$
MAP@20	50.28	63.64	60.65	58.19	**63.76**	*69.44*
MAP@40	45.24	57.37	54.75	52.45	**57.79**	*60.99*
MAP@60	43.06	54.35	51.97	49.79	**55.07**	*57.26*
MAP@80	42.60	52.90	50.68	48.73	**53.39**	*55.56*
MAP@100	43.08	52.99	50.57	48.88	**53.62**	*55.43*

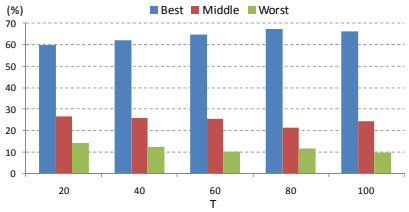


Figure 6: ***Reranking feature selection.*** **Percentages of queries for which PLM selects the Best/Middle/Worst ranking list.**

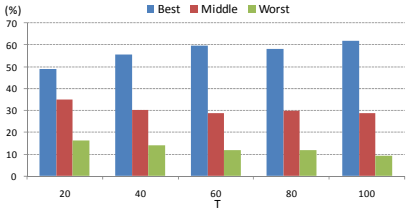


Figure 7: ***Reranking algorithm selection.*** **Percentages of queries for which PLM selects the Best/Middle/Worst ranking list.**

feature is the visual feature used in the reranking process which has a great influence on reranking performance. We use two different visual features for BR reranking and obtain their corresponding ranking lists. One feature is the aforementioned SIFT based bag-of-visual word histogram, denoted as SIFT. The other one is a combination of several low-level features adopted in [4], denoted as CF.

With two ranking lists generated by BR_{SIFT} and BR_{CF}, as well as the text-based search result list (Text), we apply PLM to select the best result list for each individual query. Table 8 gives the MAP comparison between their individual performances, as well as the performance after selection. We compare PLM with random selection. Without prior knowledge, random selection just selects from the three ranking lists randomly. It shows that our model $\text{Select}_{\text{PLM}}$ not only outperforms random selection $\text{Select}_{\text{Random}}$, but also outperforms all three individual ranking methods. In Fig. 6, we further give the % of the Best/Middle/Worst selection queries, which shows the percent of all queries for which our method chose each one of the three ranking lists. Fig. 6 clearly demonstrates that, for most queries (60%-70%), PLM selects the Best ranking list from {Text, BR_{SIFT}, BR_{CF}}. This is a substantial improvement over the random

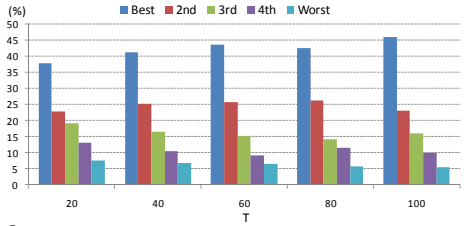


Figure 8: ***Reranking feature and algorithm mixture selection.*** **Percentages of queries for which PLM selects the Best/2nd/3rd/4th/Worst ranking list.**

selection method, which chooses the Best ranking list for 33.33% of queries, and verifies the effectiveness of our model.

Reranking Method Selection: Similar with reranking feature selection, PLM can also be applied to select the optimal reranking method for each query. This selection is conducted on reranking results generated by BR [19] and PRF [21]. The SIFT feature is used in both reranking methods. Table 9 and Fig. 7 give the MAP value comparison and % of the Best/Middel/Worst selection queries, respectively. It also demonstrates the effectiveness of our method in selecting better ranking method for each query.

Reranking Feature and Algorithm Mixture Selection: With two reranking features (SIFT and CF) and two reranking algorithms (BR and PRF), four reranking result lists are generated by their combination, *i.e.*, BR_{SIFT}, PRF_{SIFT}, BR_{CF} and PRF_{CF}. We further test PLM on this mixture selection. For each query, four reranking lists as well as the Text result list are ranked according to the value predicted by PLM. Table 10 gives the MAP value comparison We note an increase in performance after PLM selection, producing better results than all five basic methods and random selection. We also analyzed the number of queries for which PLM selects the i-th best ranking list ($i = 1, \cdots, 5$; $i = 1$ means the best and $i = 5$ means the worst), as shown in Fig. 8. It shows that PLM selects the Best/second best ranking list for about 40%/20% of the queries. It obviously outperforms random selection, which would select each of the five ranking lists for about 20% of the queries.

5.2 Search Engine Selection

In this section we investigate the effectiveness of the proposed method by applying it to optimal image search engine

Table 10: MAP (×100) comparison in *reranking feature and algorithm mixture selection* from {Text, BR_{SIFT}, BR_{CF}, PRF_{SIFT} and PRF_{CF}}.

	Text	BR_{SIFT}	BR_{CF}	PRF_{SIFT}	PRF_{CF}	$Select_{Random}$	$Select_{PLM}$	$Select_{Opt}$
MAP@20	50.28	63.64	60.21	60.65	54.00	57.76	**64.86**	*72.26*
MAP@40	45.24	57.37	53.99	54.75	49.67	52.20	**58.57**	*63.68*
MAP@60	43.06	54.35	50.65	51.97	47.52	49.51	**55.45**	*59.45*
MAP@80	42.60	52.90	49.20	50.68	50.68	49.21	**53.83**	*57.57*
MAP@100	43.08	52.99	49.08	50.57	47.58	48.66	**54.00**	*57.29*

selection and search result merging. Specifically, each query q has two ranking lists generated by two search engines: Bing and Google. Our objective is to determine which search engine returns better performance for any query q.

Dataset: A dataset was collected from two popular image search engines, Bing (Live) and Google. We selected 29 queries [3] from the top-1000 queries of Live Image Search and popular tags on Flickr. The 29 queries satisfy all the following three criteria: 1) *Popularity*: they are either top queries of Live Image Search or popular tags of Flickr; 2) *Broad topic coverage*: the 29 queries cover wide topics, *e.g.*, animals, plants, scene, objects, *etc.*; 3) *Including both simple and compound queries*: the 29 queries contain both simple queries which normally consist of one term (*e.g.*, "Cat", "Flower", *etc.*) and compound queries which are refined terms based on some certain attribute, *e.g.*, color ("White Cat"), time ("White House Night"), emotion ("Funny Dog"), *etc.* We submitted each query to Bing and Google respectively, and collected the top 1000 images returned, resulting in 50566 total images. For each query, the returned images are labeled as either "relevant" or "irrelevant". In this dataset, there are 42.23% images labeled as relevant.

To the best of our knowledge, there is no publicly available dataset which collects the search results of the same queries from several search engines and the queries are sampled from real query logs. To collect such a large query set is difficult since it may need the collaboration between academia and industry and it will involve a large amount of efforts in getting query logs from popular search engines , collecting the huge amount of data and employing human for labeling. As the first try, we collected a moderate-size dataset with 50K images [22] with the collaboration with Bing search. In the future, with the continuing collaboration with industry, we plan to collect a larger dataset from several search engines.

Experimental setting is the same as that in Section 5.1. We used the bag-of-visual words histogram for image representation and the leave-one-out method for model training.

Evaluation: For each query q, there are two ranking lists: l_{Bing} and l_{Google}. Each query's ground truth performance difference is denoted as $\delta_q^* = y(l_{Bing}) - y(l_{Google})$, and the performance difference estimated by PLM is $\delta_q = f(l_{Bing}) - f(l_{Google})$. We also evaluate PLM's preference prediction ability from the following two aspects:

1. *Prediction Accuracy* defined in (14). In this application, the correctly predicted queries are those which satisfy $\delta_q^* \delta_q > 0$, *i.e.*, the preference relationship between the two ranking lists is correctly predicted.

[3]Animal, Beach, Beijing Olympic 2008, Building, Car, Cat, Clouds, Earth, Flower, Fox, Funny Dog, George W. Bush, Grape, Hearts, Hello Kitty, Hiking, Mercedes Logo, Panda, Sky, Statue of Liberty, Sun, Trees, Wedding, White Cat, White House Night, White House, Winter, Yellow Rose, Zebra.

Table 11: Correlation coefficients and Accuracy in search engine selection from {Bing, Google}

	T=20	T=40	T=60	T=80	T=100
Kendall's τ	0.1724	0.1970	0.1232	0.1626	0.2315
AC(%)	62.07	68.97	65.52	79.31	75.86

2. Correlation coefficient: Kendall's τ correlation coefficients between the ground truth performance difference vector $\mathbf{\Delta}^* = [\delta^*_{q^{(1)}}, \cdots, \delta^*_{q^{(29)}}]^\mathbf{T}$ and the one predicted by our PLM $\mathbf{\Delta} = [\delta_{q^{(1)}}, \cdots, \delta_{q^{(29)}}]^\mathbf{T}$.

Search Engine Selection: We also evaluate 5 different Ts as in Section 5.1. Table 11 shows the correlation coefficients and accuracy. It shows that moderate correlation coefficients are achieved and the AC is more than 70% when T=80 and 100. It demonstrates that PLM can choose the better search engine between Bing and Google for the majority of queries. Therefore, better performance will be achieved after this suitable search engine selection. The MAP values of Bing ($Text_{Bing}$), Google ($Text_{Google}$) and the one generated after our PLM Selection ($Select_{PLM}$) are given in Table 12. Column $Select_{Opt}$ is the maximal MAP value arrived at by selecting the optimal search engine according to their ground truth. Table 12 shows that $Select_{PLM}$ achieves consistent performance improvements over both $Text_{Bing}$ and $Text_{Google}$ for all Ts. From Tables 11 and 12, we conclude that the proposed PLM method can be successfully applied to optimal search engine selection.

Search Results Merging: In search engine selection, for each query, we choose a better one between l_{Bing} and l_{Google}. In addition to this binary selection, we can also merge the two result lists to get a new one. For query q, when we have no idea of the performance of the two search results, they may contribute equally to the final merged result list. If we have the prior knowledge of which one is better than the other, then higher merging weight can be assigned to this better one. Our PLM can serve this role by using the predicted δ_q to set appropriate merging weights. To complete this goal, the $\mathbf{\Delta}$ is first normalized into [-1, 1]. We denote the normalized performance difference as $\tilde{\delta}_q$. Then, for query q, the merging weights for l_{Bing} and l_{Google} are defined as $w_{Bing} = \frac{1}{2}(1 + \tilde{\delta}_q)$ and $w_{Google} = \frac{1}{2}(1 - \tilde{\delta}_q)$ respectively ($w_{Bing} \geq 0$, $w_{Google} \geq 0$, $w_{Bing} + w_{Google} = 1$). To form the merged result list, we assign a merging score to each image in l_{Bing} and l_{Google}. The merging score for the i-th ranked images in l_{Bing} is $i \times (1 - w_{Bing})$. The merging score for the i-th ranked image in l_{Google} is $i \times (1 - w_{Google})$. The final merged ranking list is derived by sorting all images in l_{Bing} and l_{Google} in ascending order of their merging scores. The performance of this weighted merging result is given in Table 12, comparing with the equal merging in which $w_{Bing} = w_{Google} = 0.5$. The search engine selection

Table 12: MAP (×100) comparison in search engine selection and search results merging from {Bing, Google}

	$\text{Text}_{\text{Bing}}$	$\text{Text}_{\text{Google}}$	$\text{Select}_{\text{Random}}$	$\text{Select}_{\text{PLM}}$	$\text{Select}_{\text{Opt}}$	$\text{Merge}_{\text{Equal}}$	$\text{Merge}_{\text{Weight}}$
MAP@20	57.91	64.26	61.09	**65.80**	*71.51*	65.36	**67.21**
MAP@40	52.24	52.36	52.30	**56.12**	*60.71*	59.57	**59.80**
MAP@60	49.18	44.35	46.77	**50.78**	*54.63*	54.78	**55.85**
MAP@80	46.52	39.41	42.97	**47.77**	*50.64*	51.31	**52.76**
MAP@100	44.52	36.20	40.36	**45.16**	*48.18*	48.61	**49.83**

discussed above is actually a hard merge of the two search results with weight either 1 or 0. Table 12 clearly demonstrates that merging by leveraging our preference prediction outperforms equal merging.

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a method to automatically compare a set of ranking result lists for a given query by mining their visual information. The method is formulated within the RankSVM framework and a set of lightweight features are designed to reflect the visual difference between two ranking lists. The proposed method is successfully applied to reranking ability estimation and automatic search engine selection. Experimental results have demonstrated the effectiveness of our approach and its promising applications on reranking feature and model selection, as well as merging of image search results.

Currently, our preference learning model is built based on visual features of images only, and their textual information is not considered. In the future, we plan to further exploit this ranking list performance comparison problem by investigating both visual and textual features, to achieve better performance. Another direction of our future work is to apply this model to more potential applications, *e.g.*, query suggestion, query language selection and so on.

Acknowledgments This work was supported by Research Enhancement Program (REP) and start-up funding from the Texas State University, and was supported in part to Dr. Qi Tian by NSF IIS 1052851, Faculty Research Awards by Google, FXPAL and NEC Laboratories of America, respectively.

7. REFERENCES

[1] Trecvid video retrieval evaluation. hppt://www-nlpir.nist.gov/projects/trecvid/.

[2] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *PAMI*, 29(3):394–410, 2007.

[3] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. *ACM SIGIR*, pages 299–306, 2002.

[4] J. Cui, F. Wen, and X. Tang. Real time google and live image search re-ranking. *ACM Multimedia*, pages 729–732, 2008.

[5] J. Deng, A. C. Berg, K. Li, and F.-F. Li. What does classifying more than 10,000 image categories tell us? *ECCV*, pages 71–84, 2010.

[6] B. Geng, L. Yang, C. Xu, and X.-S. Hua. Content-aware ranking for visual search. *CVPR*, pages 3400–3407, 2010.

[7] C. Hauff, V. Murdock, and R. Baeza-Yates. Improved query difficulty prediction for the web. *ACM CIKM*, pages 439–448, 2008.

[8] B. He and I. Ounis. Inferring query performance using pre-retrieval predictors. *SPIRE*, pages 43–54, 2004.

[9] H. Imran and A. Sharan. Co-occurrence based predictors for estimating query difficulty. *IEEE ICDM Workshops*, pages 867–874, 2010.

[10] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[11] E. Jensen, S. Beitzel, D. Grossman, O. Frieder, and A. Chowdhury. Predicting query difficulty on the web by learning visual clues. *SIGIR*, pages 615–616, 2005.

[12] T. Joachims. Optimizing search engines using clickthrough data. *SIGKDD*, pages 133–142, 2002.

[13] J. Krapac, M. Allan, J. Verbeek, and F. Juried. Improving web image search results using query-relative classifiers. *CVPR*, pages 1094–1101, 2010.

[14] K.-L. Kwok, L. Grunfeld, H. L. Sun, and P. Deng. Trec 2004 robust track experiments using pircs. *TREC*, 2004.

[15] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR*, pages 2169–2178, 2006.

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[17] K. Min, L. Yang, J. Wright, L. Wu, X.-S. Hua, and Y. Ma. Compact projection: Simple and efficient near neighbor search with practical memory requirements. *CVPR*, pages 3477–3484, 2010.

[18] E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[19] X. Tian, L. Yang, J. Wang, Y. Yang, X. Wu, and X.-S. Hua. Bayesian video search reranking. *ACM Multimedia*, pages 131–140, 2008.

[20] X. Xing, Y. Zhang, and M. Han. Query difficulty prediction for contextual image retrieval. *ECIR*, pages 581–585, 2010.

[21] R. Yan, A. G. Hauptmann, and R. Jin. Multimedia search with pseudo-relevance feedback. *CIVR*, pages 238–247, 2003.

[22] L. Yang and A. Hanjalic. Supervised reranking for web image search. *ACM Multimedia*, pages 183–192, 2010.

[23] E. Yom-Tov, S. Fine, D. Carmel, and A. Darlow. Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. *ACM SIGIR*, pages 512–519, 2005.

[24] Z.-J. Zha, L. Yang, T. Mei, M. Wang, and Z. Wang. Visual query suggestion. *ACM Multimedia*, pages 15–24, 2009.