Exploration of Image Search Results Quality Assessment

Xinmei Tian, *Member, IEEE*, Yijuan Lu, *Member, IEEE*, Nate Stender, Linjun Yang, *Member, IEEE*, and Dacheng Tao, *Fellow, IEEE*

Abstract—Image retrieval plays an increasingly important role in our daily lives. There are many factors which affect the quality of image search results, including chosen search algorithms, ranking functions, and indexing features. Applying different settings for these factors generates search result lists with varying levels of quality. However, 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 aims to solve this problem and makes four main innovations. First, a preference learning model is proposed to quantitatively study and formulate the best image search result list identification problem. Second, a set of valuable preference learning related features is proposed by exploring the visual characters of returned images. Third, a query-dependent preference learning model is further designed for building a more precise and query-specific model. Fourth, the proposed approach has been tested on a variety of applications including reranking ability assessment, optimal search engine selection, and synonymous query suggestion. Extensive experimental results on three image search datasets demonstrate the effectiveness and promising potential of the proposed method.

Index Terms—Image retrieval, search results performance comparison, reranking ability assessment

1 Introduction

Image retrieval plays an increasingly important role in our daily lives. Extensive research have been conducted on retrieving images relevant to a given query. Many factors can influence image search results. Existing work aims to get better search results by focusing their efforts on various aspects of the search process, such as designing effective visual features for image representation [1], [2], building efficient image indexes [3], image annotation [4], [5], [6], [7], and developing new ranking and reranking algorithms [8], [9], [10], [11], [12]. The algorithms used in these aspects of the search process generate result lists of varying quality when used with different settings. Following are two examples for illustration.

In our first example, the image search results generated by two popular search engines, Bing and Google, are compared. Twenty nine text queries are submitted to these two search engines, and the images they returned are collected. Fig. 1 gives their AP@40 (average precision (AP), ref Section 5 for details) difference for each query of the two search engines. It is found that although Bing and Google have comparable MAP values (0.5224 and 0.5236 respectively), their performance on individual queries is quite different. Google achieves better performance on about half the queries. If 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 0.6071, about 16 percent relative improvement over Bing and Google.

In the second example, the performance of text-based image search and visual reranking are compared. 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. Although it has been found that visual reranking can generally improve the performance of text-based image search to some extent [9], [10], it is not guaranteed to benefit every query. Recent research has 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, two popular reranking methods, BR [10] and PRF [12], are applied on a public image search dataset (Web353). This dataset was collected by Krapac et al. [13], which contains 71,478 images returned by a Web search engine for 353 general textual queries. Table 2

- *X. Tian is with the CAS Key Laboratory of Technology, Geo-Spatial Information Processing and Application Systems, University of Science and Technology of China, Hefei 230027, China. E-mail: xinmei@ustc.edu.cn.*
- *Y. Lu is with the Department of Computer Science, Texas State University, San Marcos, TX 78666. E-mail: yl12@txstate.edu.*
- *N. Stender is with the Department of Computer Science, University of Nebraska-Lincoln, Lincoln, NE 68588. E-mail: nstender@cse.unl.edu.*
- *L. Yang is with Microsoft Corporation, Redmond, WA 98052-7329. E-mail: linjuny@microsoft.com.*
- *D. Tao is with the Centre for Quantum Computation & Intelligent Systems and the Faculty of Engineering and Information Technology, University of Technology, Sydney, Ultimo, NSW 2007, Australia. E-mail: dacheng.tao@uts.edu.au.*

Manuscript received 1 July 2015; revised 3 Oct. 2015; accepted 25 Oct. 2015; date of current version 18 Dec. 2015.
Recommended for acceptance by J. Wang, G.-J. Qi, N. Sebe, and C. Aggarwal.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TBDATA.2015.2497710

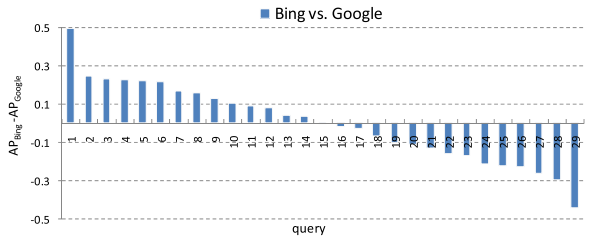


Fig. 1. Image search result comparison between Bing and Google: AP@40 difference $\mathrm{AP}_{\mathrm{Bing}} - \mathrm{AP}_{\mathrm{Google}}$ on each query.

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. It is found 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. In the further investigation of 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: offline training and online testing. In the training stage, the visual distribution characteristics of good and bad search result lists are explored and a set of light-weight features is derived 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 with RankSVM [14]. Finally, in the testing stage, the developed PLM is applied to predict the preference score for search result lists of any testing query.

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. In query expansion/suggestion, the proposed method can automatically identify the best one from multiple candidates of suggested search terms. In general, given different search algorithm settings, our approach can automatically select the optimum settings for each query.

TABLE 1
MAP@40 ($\times$100) of Bing, Google, and After Optimal Search Engine Selection for Each Query

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

TABLE 2
MAP@20 ($\times$100) of the Text-Based Search (Text) and the Two Reranking Methods, PRF and BR

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

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

- The image search result preference learning problem is quantitatively studied and formulated.
- A novel framework and a set of valuable features to automatically compare the quality of image search result lists are proposed.
- A general preference learning model and a query-dependent preference learning model are proposed.
- The proposed approach has been tested on a variety of applications including optimal search engine selection, merging of search result lists, selecting the best visual feature and reranking approach for each individual query, and synonymous query suggestion. The superior performance has demonstrated its promising application potential.
- Our work will explicitly guide the research in visual reranking ability estimation and provide a path for query difficulty modeling.

The preliminary version of this work was presented at ACM Multimedia [15]. In this journal version, we have enhancement in four aspects. 1) We give a comprehensive review of the most related works, query difficulty estimation, in Section 2, and compare our method with them in the experiments. 2) A query-dependent preference learning model is further designed for building a more precise and query-specific model. Corresponding experiments are added to verify the superiority of this query-dependent preference learning model. 3) We collect a new dataset which consists of 38,800 images. We collected 97 synonymous query groups from WordNet and collected the top 200 returned images for each query. 4) We study the effectiveness of our preference learning method for synonymous query suggestion scenario by conducting experiments on our newly collected dataset.

2 Related Work

Considerable research has been proposed to improve image search from various aspects, such as image annotation [4],

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

image ranking [3], [8] and visual reranking [16], [17], [18]. All of 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 list with the highest performance is chosen and presented to users.

There is no previous work on automatic image search result comparison and selection. The most related is the query difficulty prediction. Query difficulty prediction aims to predict whether a query will have a high retrieval performance in a document collection. It has been explored for many years [19], [20], [21], [22], [23], [24], [25] in document retrieval and its importance has been recognized in the information retrieval community. The work can be categorized into two groups, *pre-retrieval prediction* and *post-retrieval prediction*, according to whether the prediction is conducted before or after retrieval.

Pre-retrieval query difficulty prediction attempts to evaluate search performance before the retrieval step [20], [21], [25], mainly relying on statistics of query terms over document collections. For example, Kwok et al. [20] extracted simple features such as log document frequency and query term frequency to train a query difficulty prediction model via support vector regression. Imran and Sharan [25] proposed two pre-retrieval query difficulty predictors based on the co-occurrence information among query terms, with the assumption that higher co-occurrence of query terms means more information is conveyed which leads to an easier query.

In post-retrieval query difficulty prediction, the retrieval step is conducted first and query difficulty prediction evaluates the performance of the returned results. Various post-retrieval query difficulty predictors are proposed [19], [22], [23], [24], [26], [27], [28], [29], [30]. Those methods can be further grouped into three categories: clarity-based, stability-based, and coherence-based. Clarity-based methods [19], [24], [28], [30], [31] assume the distribution difference between the retrieved documents and the whole document collection can indicate the query difficulty level. Cronen-Townsend et al. [19] proposed the Clarity Score (CS) which measures the ambiguity of a query through the Kullback-Leibler (KL) divergence [32] between the language models created from top-retrieved documents and the whole document collection. Zhou and Croft [31] estimated the query difficulty by measuring the information change from an average returned document to the actual retrieval results. Stability-based methods [23], [26], [29] predict query difficulty by investigating the stability of several retrieval results obtained from different ways. Yom-Tov et al. [23] measured the agreement between the top returned results of the full query and its sub-queries. Aslam and Pavlu [26] first used different scoring functions to obtain numbers of retrieval lists and then mapped each ranked list to a probability distribution. The query difficulty predictor is derived from the Jensen-Shannon divergence [33] among these distributions. Zhou and Croft [29] defined the ranking robustness as the similarity between ranked lists generated from original collection and corrupted collection. Coherence-based methods indicate the search quality by using the tightness of the top returned documents. A coherence score indicator is proposed by He et al. [34]. It measures the portion of coherent document pairs in the top returned document set. To select the best query expansion for spoken content retrieval, Rudianc et al. [27] exploited the coherence of the top ranked documents returned by the unexpanded query and several query expansion alternatives. There are also some learning based methods. Jensen et al. [22] 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. [35] used textual features to predict whether a query is difficult to represent as images or not. This work does not investigate the image search performance, but only classifies the queries into two categories "easy" or "hard". Li et al. [36] measured the query difficulty in terms of the consistency degree between query image and its top returned images. Here the query is an image and the consistency degree can be calculated by averaging their visual distance. Rudianc et al. [37] extended their previous coherence indicator [27] by generating new concept vectors for video representation.

Unlike query difficulty prediction, which estimates the performance of a search result list for a given query, our work targets at 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 challenging 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 Image Search Result Preference Learning

3.1 Preference Learning Model

For query q and an image collection $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, 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 $\mathcal{L}^{(q)} = \{l_1, \ldots, l_{n_q}\}$. 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 [38] and Normalized Discounted Cumulated Gain (NDCG) [39].

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 straightforward. 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 [14] algorithm. It minimizes the prediction errors on a set of training queries $\mathcal{Q} = \{q^{(1)}, \ldots, q^{(m)}\}$,

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

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 $\mathcal{L}^{(q)} = \{l_1, \ldots, l_{n_q}\}$

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

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, $\mathcal{L}^{(q')} = \{l_1, \ldots, l_{n_{q'}}\}$. $f(\cdot)$ 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.

3.2 Query-Dependent Preference Learning Model

It is obvious that the constitution of the training data will greatly influence the performance of the trained Preference Learning Model. A training set comprising of informative samples will lead to a high-performing model [40]. The study in active learning also shows a small fraction of discriminative training data can often yield satisfying performance [41]. This is especially important in our situation since images returned for difference types of queries have different visual distributions. As a consequence, it is inappropriate to handle all queries using a universal preference learning model. Inspired by previous work [42], we propose to exploit different preference learning models for different queries, i.e., query-dependent PLM.

Specifically, we derive the query-dependent PLM by constructing a query-dependent training query set for each query. The straightforward way is to classify queries into several categories (landmark, people, object, etc.), and then train a PLM model for each category with queries belonging to this category as training set. In the testing stage, the test query is mapped to the pre-defined query category and the corresponding PLM is applied. This approach is efficient (several PLMs need to be trained offline) but requires an accurate query classification process which is difficult to achieve. Therefore, a simple solution is selected. For query q, we find its K closet queries from $\mathcal{Q}_{train}$ as training set to train query-dependent PLM for it.

The key problem is to find the K-nearest queries for q accurately. One text document retrieval work [42] proposes to first adopt BM25 to find the top T ranked documents for query q and then take the means of the features (query-dependent feature, e.g., tf, idf) of the T documents as a feature of the query. With each query represented as a feature vector, the K-nearest queries of q are found by their distance in Euclidean space. The challenge in our problem is that the query and the images are in different domain: textual and visual respectively. It is difficult to derive the query-dependent feature to represent a query. To tackle this domain gap problem, we propose to use the popular bag of visual word (BOVW) image representation [43], which is designed analogically to text retrieval. In BOVW, an image can be treated as a document and represented by a set of visual words. We represent each query by its language model $P(w|q)$, and define the distance between queries $q^{(i)}$ and $q^{(j)}$ as the KL divergence [32] between their query language models,

$$\begin{aligned} \text{Dist}(q^{(i)}, q^{(j)}) &= D_{KL}(P(w|q^{(i)})|P(w|q^{(j)})) \\ &= \sum_{w \in V} P(w|q^{(i)}) \log \frac{P(w|q^{(i)})}{P(w|q^{(j)})}. \end{aligned} \tag{5}$$

$P(w|q)$ is the language model of query q, defined as

$$P(w|q) = \sum_{\mathbf{x} \in \mathcal{R}} P(w|\mathbf{x})P(\mathbf{x}|q), \tag{6}$$

where $w \in V$ is a visual word, and $\mathcal{R}$ is a set of images returned for query q. $P(w|\mathbf{x})$ is defined as the term frequency of the word w in image $\mathbf{x}$. For $P(\mathbf{x}|q)$, since $P(\mathbf{x}|q) \propto P(q|\mathbf{x})P(\mathbf{x})$ and each image $\mathbf{x}$ has an equal prior $P(\mathbf{x})$, we only need to estimate the likelihood $P(q|\mathbf{x})$. The $P(q|\mathbf{x})$ denotes the possibility of image $\mathbf{x}$ to be relevant to q,

therefore we can estimate it by leveraging the image search result l. We define $P(q|\mathbf{x})$ as 1 if image $\mathbf{x}$ appears in the top-T returned images for query q, else 0 if image $\mathbf{x}$ does not appear in the top-T returned images for query q. In other words, the query language model is estimated over the top-T ranked images which are assumed to be pseudo-relevant to the query q according to the widely used pseudo relevance feedback assumption [12], [44].

3.3 Training Sample Augmentation

Up to now, we have shown how to select query-dependent training queries. The next step is to construct preference pairs from the training queries. However, 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) for each query. To solve this problem, we can construct additional ranking lists.

Unlike other problems, we have the advantage that 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 $\mathcal{L}^{(q)}$:

$$\mathcal{L}^{(q)} \leftarrow \mathcal{L}^{(q)} \cup \{l^{manual}\}. \tag{7}$$

In this paper we create three manual ranking lists, which are diverse at the same time, 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.

4 Preference Learning 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 $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, 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

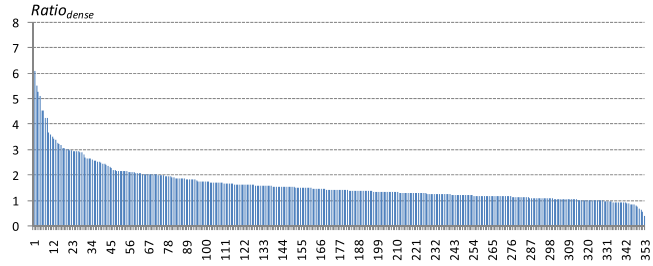


Fig. 2. Sorted $Ratio_{dense}$ values in 353 queries. There are 329 queries whose average density of relevant images ($AvgDense_+$) is larger than the average density of irrelevant images ($AvgDense_-$), i.e., $Ratio_{dense} > 1$. Additionally, the $AvgDense_+$ is significantly larger than $AvgDense_-$ in 286 queries (T-test, significance level 5 percent).

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) [45],

$$p_{\mathbf{x}_i} = \frac{1}{|\mathcal{N}(\mathbf{x}_i)|}\sum\nolimits_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)} k(\mathbf{x}_i - \mathbf{x}_j), \tag{8}$$

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 (8). In this paper we adopt the popular visual bag-of-word image representation. More details will be introduced in Section 5.

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,

$$AvgDense_+ = \frac{1}{|\mathcal{X}^+|}\sum\nolimits_{\mathbf{x}_i \in \mathcal{X}^+} p_{\mathbf{x}_i}, \tag{9}$$

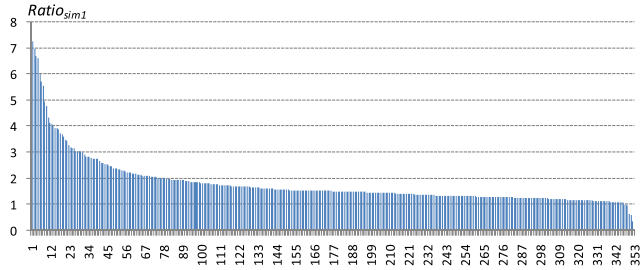
$$AvgDense_- = \frac{1}{|\mathcal{X}^-|}\sum\nolimits_{\mathbf{x}_i \in \mathcal{X}^-} p_{\mathbf{x}_i}, \tag{10}$$

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, $Ratio_{dense} = AvgDense_+/AvgDense_-$.

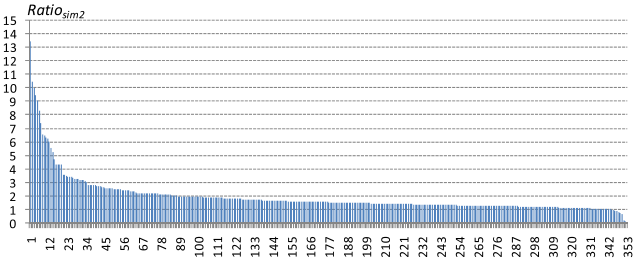
We compute $Ratio_{dense}$ for all 353 queries in Web353 and plot them in Fig. 2 by sorting them in descending order. From Fig. 2, we see that, among 353 queries, there are 329 queries whose average density of relevant images is larger than the average density of irrelevant images ($Ratio_{dense} > 1$). To verify whether $AvgDense_+$ is significantly larger than $AvgDense_-$, we further perform a statistical significance test. We used the T-test with a 5 percent 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-



(a) Sorted $Ratio_{sim1}$ in 353 queries. $AvgSim_{++}$ is larger than $AvgSim_{+-}$ in 348 queries ($Ratio_{sim1} > 1$), and is significantly larger than $AvgSim_{+-}$ in 347 queries (T-test, significance level 5%).



(b) Sorted $Ratio_{sim2}$ in 353 queries. $AvgSim_{++}$ is larger than $AvgSim_{--}$ in 344 queries ($Ratio_{sim2} > 1$), and is significantly larger than $AvgSim_{--}$ in 339 queries (T-test, significance level 5%).

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

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 [46].

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 percent 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 represent with 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", "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.

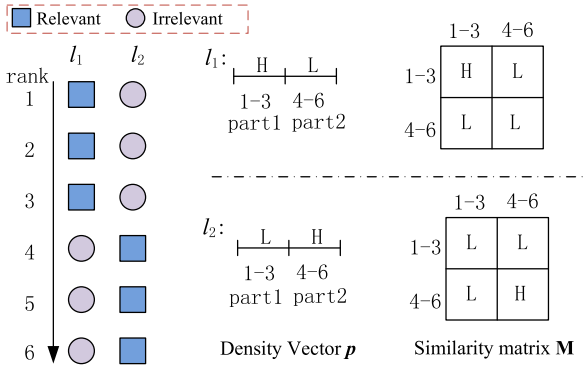


Fig. 4. Illustration of density and similarity distribution difference between two ranking lists. For the better ranking list l_1, its top ranked images have high (H) density and share high visual similarity while bottom ranked images have low (L) density and visual similarity.

Our experimental results reported in Sections 5, 6, 7 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 them by using a toy example for illustration, as shown in Fig. 4. Suppose there are six images returned for query q, three relevant (denoted by square) and three 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 pair-wise image similarity. The (i, j) element m_{ij} in $\mathbf{M}$ denotes the visual similarity between the ith ranked image and jth 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,

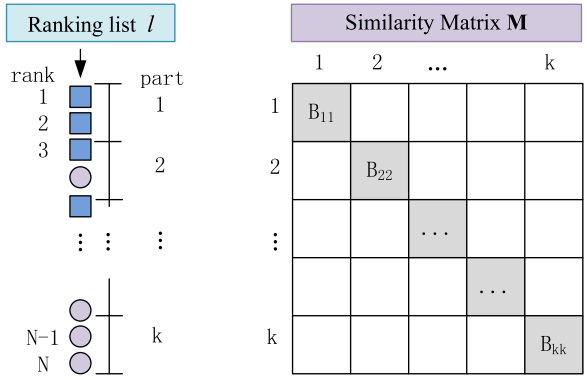


Fig. 5. The N images are split into k parts along their ranks equally. Therefore, the $N \times N$ similarity matrix $\mathbf{M}$ is split into $k \times k$ blocks. We calculate the mean and variance of the k diagonal blocks to derive the similarity distribution feature vector F_{SD}.

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}:

$$F_{SD}(i) = [mean(\mathbf{M}^{(i,i)}), var(\mathbf{M}^{(i,i)})], i = 1, \ldots, k, \quad (11)$$

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 should 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 $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, we can derive a density vector $\mathbf{p} = [p_1, \ldots, p_N]^{\mathrm{T}}$, where p_i is the density of the ith ranked image in l as defined in (8). We also split the N images into k groups and calculate the mean and variance of the density of the images in each part,

$$F_{DD}(i) = [mean(\mathbf{p}^{(i)}), var(\mathbf{p}^{(i)})], i = 1, \ldots, k, \quad (12)$$

where $\mathbf{p}^{(i)}$ is the sub density vector for images in part i. By concatenating $F_{DD}(i)$, $i = 1, \ldots, k$, 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 complement. Especially in the case where 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 $\{p_1, p_2, \ldots, p_T\}$ can be quantified into a C-bin histogram by mapping them into the corresponding bins. We denote this density histogram feature as F_{HD},

$$F_{HD}(c) = \frac{1}{T}|\{i|i = 1, \ldots, T, p_i \in c\text{th bin}\}|, \quad (13)$$

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

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,

$$F_{HS}(c) = \frac{1}{T^2}|\{(i,j)|i, j = 1, \ldots, T, m_{ij} \in c\text{th bin}\}|, \quad (14)$$

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

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.

5 Experiment 1: Reranking Ability Assessment

We investigate the effectiveness of the proposed preference learning model by applying it to three applications. The first is the reranking ability assessment described in Section 5, the second is the optimal search engine selection in Section 6, and the third is the synonymous query ranking is Section 7.

In this section, we investigate the effectiveness of the proposed preference learning model 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 abilityt*$^*_q$ 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.

5.1 Experimental Setting

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 71,478 images returned by the French search engine Exalead[1] 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 percent images labeled as relevant. For each query, we conduct Bayesian Reranking (BR) [10] 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 [38], 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

$$\mathrm{AP@}T = \frac{1}{Z_T}\sum_{i=1}^{T}[precison(i) \times rel(i)], \quad (15)$$

1. http://www.exalead.com/search/image

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 $\mathrm{AP@}T = 1$ for a perfect ranking result list.

Model training. We use the leave-one-out method for PLM training. We train the query-dependent PLM for each query by selecting query-dependent training query set from the leftover 352 queries, as introduced in Section 3.2. We repeat the process 353 times to ensure that each query has been used as test query at least once. The K, cardinality of query-dependent training query set, is set as 250 empirically. We will discuss its effect later in Section 5.3.

Visual image 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) [2] local descriptors are extracted from each image on a dense grid. Then, a codebook is generated by clustering all local descriptors into 1,000 groups [46]. By quantizing local descriptors into visual words, each image is represented as a 1,000-dimensional histogram. Spatial pyramid matching [1] 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):

$$\mathrm{AC} = \frac{\#\text{Correctly predicted queries}}{\#\text{Total queries}}. \tag{16}$$

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:

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

$$\mathrm{P-} = \frac{\#\text{Correctly predicted negative queries}(t_q^* < 0 \text{ and } t_q < 0)}{\#\text{Total negative queries}}.$$

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 $\mathbf{t}^* = [t^*_{q^{(1)}}, \ldots, t^*_{q^{(353)}}]^{\mathrm{T}}$ and the one predicted by PLM $\mathbf{t} = [t_{q^{(1)}}, \ldots, t_{q^{(353)}}]^{\mathrm{T}}$. As widely used in query difficulty prediction [22], [23], we calculate the correlation coefficient between $\mathbf{t}^*$ and $\mathbf{t}$. Two of the mostly used correlation measurements are applied, including the non-parametric rank correlation Kendall's τ [47] adopted in [20], [23], [24], [25], [26], [29], [30] and Spearman's ρ [48] adopted in [19], [21], [22], [25], [34]. The correlation coefficient falls within the range $[-1, 1]$, where -1 means perfect negative correlation and 1 means perfect positive correlation, and 0 means independence between $\mathbf{t}^*$ and $\mathbf{t}$.

TABLE 4
Correlation Coefficients and Accuracy in *Reranking Ability Assessment*

		Kendall's τ	Spearman's ρ	AC(%)	P+(%)	P−(%)
T = 20	QD	0.0543	0.0787	57.51	67.97	33.33
	CI	0.3617	0.5165	73.65	**95.70**	17.24
	PLM	**0.3643**	**0.5276**	**75.92**	91.41	**39.08**
T = 40	QD	0.1485	0.2166	65.44	77.44	30.49
	CI	0.4217	0.5964	75.64	**94.74**	18.29
	PLM	**0.4511**	**0.6236**	**79.32**	91.73	**43.90**
T = 60	QD	0.0826	0.1208	65.44	77.09	26.03
	CI	0.4520	0.6261	77.62	91.36	27.40
	PLM	**0.4798**	**0.6476**	**81.59**	**92.00**	**47.95**
T = 80	QD	0.1244	0.1804	68.84	81.09	27.40
	CI	0.4418	0.6146	73.65	87.27	27.40
	PLM	**0.4842**	**0.6569**	**81.59**	**91.27**	**50.68**
T = 100	QD	0.1400	0.207	70.82	82.44	28.99
	CI	0.4396	0.6057	73.94	83.51	40.58
	PLM	**0.4814**	**0.6545**	**83.00**	**91.76**	**53.62**

5.2 Reranking Ability Assessment

We evaluate PLM at five truncation levels, i.e., $y(l) = \mathrm{AP@}T, T = \{20, 40, 60, 80, 100\}$. We implement the document query difficulty method proposed in [22] as a baseline, since this method conducts query difficulty prediction through supervised model training. Based on [22], 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 [22] as QD. We also compare our method with the recently proposed coherence indicator [37]. We note this method as CI. For the parameters in the comparison methods, we follow the best settings reported in [22] and [37].

Table 4 shows the correlation coefficients and accuracy of our approach and the baselines QD and CI. It reveals that our method outperforms QD and CI in both correlation coefficients and accuracy. Our method achieves strong correlation and achieves about 80 percent prediction 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 percent of the negative queries, preventing performance decrease, while sacrificing performance gain on only 8.73 percent of the positive queries. Although CI achieves a little better P+ when T = 20 and T = 40 than PLM, its P− (< 20 percent) is much lower than PLM. It means that CI can hardly identify queries which are not suitable for reranking.

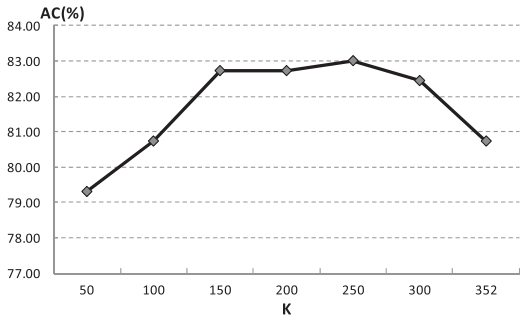


Fig. 6. AC (T = 100) with different K.

5.3 Effects of K in Query-Dependent PLM

We test the performance of query-dependent PLM with different values of parameter K, the number of similar queries selected for training the query-dependent model, to validate its effectiveness as well as the effects of K. Notice that when K = 352, it becomes the universal PLM in which all training queries are used. We present the results (in terms of AC at T = 100 for illustration) with respect to different values of K, as given in Fig. 6.

It shows that when K varies from 50 to 352, the performance first increases and then decreases. When K is small (e.g., less than 100), the performance is unsatisfactory due to the insufficiency of training queries. More training queries will bring more information and thus a better performance can be achieved. Therefore the performance improves gradually as K increases and arrives at the peak around K = 250. However, when K keeps increasing to 300 and 352, the performance decreases substantially because noisy training queries have a negative effect on the model. This result demonstrates the effectiveness of query-dependent PLM.

Our proposed method is highly efficient. When a testing query is given, the main time cost is in the step of calculating the similarity matrix $\mathbf{M}$ which is about $O(dN^2)$ where d is the dimension of image representation, and in the step of searching the query-dependent training set which is about $O(dT + dm + mlogK)$ where m is the number of queries in the training set. We test the average time cost on Web353 dataset. It takes less than 0.1 second per query. The algorithm is implemented using MATLAB and run on a PC with 3.40 GHz Intel Core CPU and 4 GB memory in single thread.

5.4 Reranking Filtering

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 ($\mathrm{Select_{QD}}$), CI ($\mathrm{Select_{CI}}$) and PLM ($\mathrm{Select_{PLM}}$.) Table 5 lists the MAP values for text-based search (Text), reranking (BR), $\mathrm{Select_{QD}}$, $\mathrm{Select_{CI}}$, and $\mathrm{Select_{PLM}}$. Column $\mathrm{Select_{Opt}}$ 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 $\mathrm{Select_{PLM}}$ performs better than both Text and BR, while $\mathrm{Select_{QD}}$ and $\mathrm{Select_{CI}}$ only achieves a moderate performance between Text and BR. The reason why the QD method in [22] 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.

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 (51.82±2.0) percent prediction accuracy (AC over $T = 20, 40, 60, 80, 100$) in reranking ability assessment, which is close to random prediction. And the $\mathrm{Select_{PLM}}$ 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.

TABLE 6
MAP (×100) Comparison in Reranking Ability Assessment for the Queries Which Do Not Satisfy the Assumptions

	Text	BR	$\mathrm{Select_{Random}}$	$\mathrm{Select_{PLM}}$	$\mathrm{Select_{Opt}}$
MAP@20	41.00	41.29	41.15	**41.78**	*47.51*
MAP@40	36.57	36.91	36.74	**37.24**	*41.09*
MAP@60	34.58	**35.38**	34.98	35.25	*38.37*
MAP@80	35.32	36.15	35.74	**36.27**	*38.92*
MAP@100	36.38	**37.90**	37.14	37.76	*40.19*

TABLE 5
MAP (×100) Comparison in *Reranking Ability Assessment*

	Text	BR	$\mathrm{Select_{Random}}$	$\mathrm{Select_{QD}}$	$\mathrm{Select_{CI}}$	$\mathrm{Select_{PLM}}$	$\mathrm{Select_{Opt}}$
MAP@20	50.28	63.64	56.96	59.65	64.22	**64.56**	*66.80*
MAP@40	45.24	57.37	51.31	55.29	57.85	**58.03**	*59.40*
MAP@60	43.06	54.35	48.71	51.89	54.70	**54.96**	*55.96*
MAP@80	42.60	52.90	47.75	51.24	52.89	**53.50**	*54.32*
MAP@100	43.08	52.99	48.04	51.65	52.81	**53.54**	*54.24*

MAP is the mean of AP over all queries.

TABLE 7
MAP ($\times$100) Comparison in Reranking Ability Assessment for Each of the Five Query Categories (T = 60)

	Text	BR	$\text{Select}_{\text{Random}}$	$\text{Select}_{\text{PLM}}$	$\text{Select}_{\text{Opt}}$
landmark	50.92	62.44	56.68	**63.75**	*64.88*
design	41.94	57.83	49.89	**58.59**	*59.00*
people	47.08	**61.61**	54.35	61.59	*62.08*
object	33.66	35.33	34.50	**36.64**	*39.00*
others	40.38	50.70	45.54	**51.31**	*52.10*

To further investigate the effectiveness of PLM for different categories of queries, the 353 queries are grouped into five categories: landmark (53), design (60), people (98), object (54) and others (88). We conducted experiments on each of the five 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 percent respectively. Even in the category "object", of which queries usually have images with a large visual appearance variance, moderate AC (61.11 percent) 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.

5.5 Best Reranking List Selection

PLM can also be applied to select the optimal reranking results. 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 apply PLM to select the best result list for each individual query. For each query, four reranking lists as well as the Text result list are ranked according to the value predicted by PLM. Table 8 gives the MAP value comparison between their individual performances, as well as the performance after selection. 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 ith best ranking list ($i = 1, \ldots, 5$; $i = 1$ means the best and $i = 5$ means the worst), as shown in Fig. 7. It shows that PLM selects the Best/second best ranking list for about 40 percent/20 percent of the queries. It obviously outperforms random selection, which would select each of the five ranking lists for about 20 percent of the queries.

6 Experiment 2: Search Engine Selection

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

6.1 Experimental Setting

Dataset. A dataset was collected from two popular image search engines, Bing (Live) and Google. We selected 29 queries[2] from the top-1,000 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 1,000 images returned, resulting in 50,566 total images. For each query, the returned images are labeled as either "relevant" or "irrelevant". In this dataset, there are 42.23 percent images labeled as relevant.

Experimental setting is the same as that in Section 5. We used the bag-of-visual words histogram for image representation and the leave-one-out method for model training. Here we apply the universal PLM instead of query-dependent PLM since the number of queries is limited.

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 (16). 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.

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

6.2 Search Engine Selection

We also evaluate five different Ts as in Section 5. Table 9 shows the correlation coefficients and accuracy. It shows that moderate correlation coefficients are achieved and the AC is more than 70 percent 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 ($\text{Text}_{\text{Bing}}$), Google ($\text{Text}_{\text{Google}}$) and the one generated after our PLM Selection ($\text{Select}_{\text{PLM}}$) are given in Table 10. Column $\text{Select}_{\text{Opt}}$ is the maximal MAP value arrived at by selecting the optimal search engine according to their ground truth. Table 10

2. 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 8
MAP (×100) Comparison in *Best Reranking List Selection* from {Text, $\mathrm{BR}_{\mathrm{SIFT}}$, $\mathrm{BR}_{\mathrm{CF}}$, $\mathrm{PRF}_{\mathrm{SIFT}}$ and $\mathrm{PRF}_{\mathrm{CF}}$}

	Text	$\mathrm{BR}_{\mathrm{SIFT}}$	$\mathrm{BR}_{\mathrm{CF}}$	$\mathrm{PRF}_{\mathrm{SIFT}}$	$\mathrm{PRF}_{\mathrm{CF}}$	$\mathrm{Select}_{\mathrm{Random}}$	$\mathrm{Select}_{\mathrm{PLM}}$	$\mathrm{Select}_{\mathrm{Opt}}$
MAP@20	50.28	63.64	60.21	60.65	54.00	57.76	**64.76**	*72.26*
MAP@40	45.24	57.37	53.99	54.75	49.67	52.20	**58.71**	*63.68*
MAP@60	43.06	54.35	50.65	51.97	47.52	49.51	**55.47**	*59.45*
MAP@80	42.60	52.90	49.20	50.68	50.68	49.21	**53.96**	*57.57*
MAP@100	43.08	52.99	49.08	50.57	47.58	48.66	**54.09**	*57.29*

shows that $\mathrm{Select}_{\mathrm{PLM}}$ achieves consistent performance improvements over both $\mathrm{Text}_{\mathrm{Bing}}$ and $\mathrm{Text}_{\mathrm{Google}}$ for all Ts. From Tables 9 and 10, we conclude that the proposed PLM method can be successfully applied to optimal search engine selection.

6.3 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 Δ 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 10, comparing with the equal merging in which $w_{Bing} = w_{Google} = 0.5$. The search engine selection discussed above is actually a hard merge of the two search results with weight either 1 or 0. Table 10 clearly demonstrates that merging by leveraging our preference prediction outperforms equal merging.

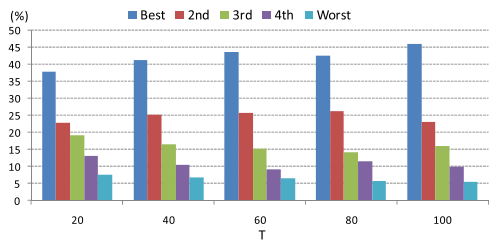


Fig. 7. *Best reranking list selection*. Percentages of queries for which PLM selects the Best/second/third/fourth/Worst ranking list. For random selection, it should be 20 percent for each of the five ranking lists. This figure shows that our model can select the Best ranking list for about 40 percent of the queries, and the second best list for about 20 percent of the queries.

7 Experiment 3: Synonymous Query Suggestion

In image search, it is not trivial for users to formulate a proper query which can express their search intents clearly and precisely. Sometimes, the users can imagine what they desire, but they have difficulty expressing their desire in precise wording [49], [50]. This is the so called *intention gap* problem, i.e., the gap between users' search intents and the queries. To address this problem, query suggestion techniques have been proposed [51], [52] which provide a set of alternative search terms to users. Although suggesting alternative search terms to the user could help the user improve the quality of their results, different queries generate different search result lists. To find which query gets the best search performance, users need to examine the search results of all suggested queries. It would be good if the suggested queries could be ranked according to their search quality. Our method proposed in this paper can serve this role by automatically comparing the quality of search results generated by those alternative suggestions. We start with the synonymous query suggestion problem. For each query, there are two synonymous query terms. We compare their search quality and to pick out the better one.

7.1 Experimental Setting

Dataset. We collected a synonymous query image dataset for synonymous query selection from Web. 97 synonymous query groups are collected from WordNet [53]. In each group, there are two synonymous query terms, for example, {"hen", "chicken"}, {"ship", "boat"}, {"mouse", "rat"}, {"stone", "rock"}, {"parcel", "package"}, {"rabbit", "bunny"}, {"bicycle", "bike"}. We submitted each of the query terms into Google, and collected the top 200 returned images, resulting in a dataset consisting of 38,800 images in total. The relevance of each image to its corresponding query is evaluated manually through the developed labeling tool. The interface of the labeling tool is shown in Fig. 8. Each image is given a a label of "relevant" or "irrelevant". There are 67.41 percent images labeled as relevant. The bag-of-visual words histogram is also adopted for image representation and the leave-one-out method is applied for model training.

TABLE 9
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
Spearman's ρ	0.2483	0.3281	0.2320	0.2360	0.3305
AC(%)	62.07	68.97	65.52	79.31	75.86

TABLE 10
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**

Evaluation. For each of the 97 synonymous query groups, we denote the two terms in group G as q_{s1} and q_{s2} and their corresponding ranking lists as l_{s1} and l_{s2} respectively. The ground truth performance difference between l_{s1} and l_{s2} in G is δ^*_G. The performance difference estimated by PLM is δ_G. The evaluation is similar to that in Section 6:

1. *Prediction accuracy*: percentage of groups in which the preference relationship between the two synonymous query terms is correctly predicted, i.e., $\delta^*_G \delta_G > 0$.

2. *Correlation coefficien*t: Kendall's τ and Spearman's ρ correlation coefficients between the ground truth performance difference vector $\Delta^* = [\delta^*_{G^{(1)}}, \ldots, \delta^*_{G^{(97)}}]^{\mathbf{T}}$ and the one predicted by our PLM $\Delta = [\delta_{G^{(1)}}, \ldots, \delta_{G^{(97)}}]^{\mathbf{T}}$.

7.2 Synonymous Query Term Suggestion

Fig. 9 gives the absolute values of ground truth AP@40 difference between the two synonymous query terms in each of the 97 groups. We can see that the quality of the search result varies a lot within synonymous query terms. It is highly desired to automatically identify which one is better.

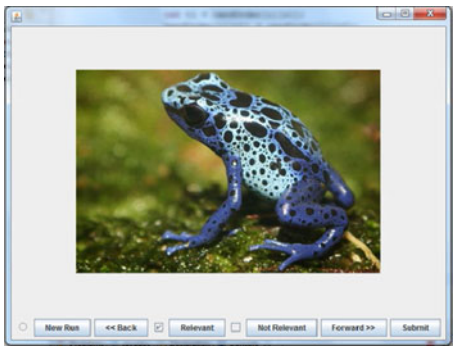


Fig. 8. The interface of the image labeling tool. Each image is evaluated manually and is assigned a label of "relevant" or "not relevant".

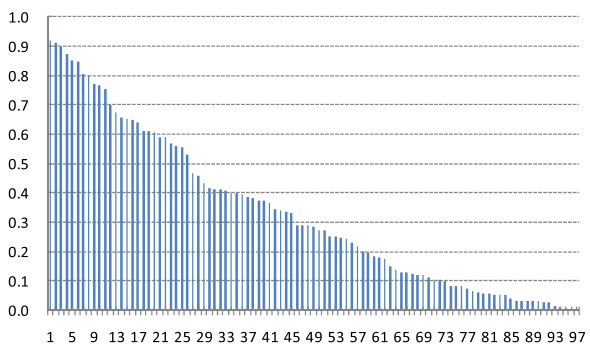


Fig. 9. Absolute value of AP@40 difference between the two synonymous query terms in each of the 97 groups (sorted in desending order for better view).

PLM compares the quality of the search result lists of two synonymous query terms in each group, and the experimental results in terms of correlation coefficients and accuracy are given in Table 11. It shows that PLM is able to identify the better one from the two synonymous query terms in most cases with an accuracy above 70 percent. When T = 40, the accuracy reaches 77.89 percent.

Since we know the performance preference within the two synonymous query terms in each group, we can suggest the better one to users. A better performance can be achieved with this optimal query term suggestion. Therefore, our PLM is suitable for the task of synonymous search term selection. Table 12 demonstrates the improvement our PLM selection can have, showing the MAP of the baseline lists as well as the MAPs resulting from a random selection of search term from each group, our PLM selection and the optimal selection $\text{Suggest}_{\text{Opt}}$, in which the query term is suggested optimally according to ground truth performance. It can be clearly observed that the performance is noticeably improved after PLM suggestion. From these results, we can conclude that our proposed framework can train a PLM which can be successfully applied to the task of image search result judgment for synonymous queries.

8 Conclusion And Future Work

In this paper, we proposed a method to automatically compare the quality of 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 distribution difference between ranking lists with varying levels

TABLE 11
Correlation Coefficients and Accuracy in Synonymous Query Suggestion

	T = 20	T = 40	T = 60	T = 80	T = 100
Kendall's τ	0.2749	0.3001	0.3063	0.3153	0.3222
Spearman's ρ	0.3902	0.429	0.4215	0.4376	0.4595
AC(%)	74.42	77.89	72.16	72.16	70.10

TABLE 12
MAP (×100) Comparison in *Synonymous Query Suggestion*

	$\text{Suggest}_{\text{Random}}$	$\text{Suggest}_{\text{PLM}}$	$\text{Suggest}_{\text{Opt}}$
MAP@20	74.71	**85.43**	*90.51*
MAP@40	71.16	**82.79**	*87.75*
MAP@60	68.43	**78.87**	*85.17*
MAP@80	66.03	**75.69**	*82.96*
MAP@100	64.10	**73.14**	*81.27*

of quality. The proposed method is successfully applied to reranking ability estimation, automatic search engine selection and synonymous query suggestion. Extensive experimental results have demonstrated the effectiveness of our approach and its promising applications on reranking feature and model selection, merging of image search results, as well as query suggestion.

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.

Acknowledgments

This work is supported in part by the 973 project No.2015CB351803, the NSFC No.61390514, No.61201413, and No. 61572451, Youth Innovation Promotion Association CAS CX2100060016, the Fundamental Research Funds for the Central Universities WK2100060011 and WK2100100021 to Xinmei Tian, in part by the Texas State University Research Enhancement Program (REP), Army Research Office grant W911NF-12-1-0057, and NSF CNS 1305302 to Yijuan Lu, in part by Australian Research Council Projects: DP-140102164, ARC FT-130101457, and ARC LP-140100569 to Dacheng Tao.

References

[1] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2006. pp. 2169–2178.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[3] 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," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 3477–3484.

[4] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos, "Supervised learning of semantic classes for image annotation and retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 394–410, Mar. 2007.

[5] D. Tao, L. Jin, W. Liu, and X. Li, "Hessian regularized support vector machines for mobile image annotation on the cloud," *IEEE Trans. Multimedia*, vol. 15, no. 4, pp. 833–844, Jun. 2013.

[6] J. Tang, H. Li, G.-J. Qi, and T.-S. Chua, "Image annotation by graph-based inference with integrated multiple/single instance representations," *IEEE Trans. Multimedia*, vol. 12, no. 2, pp. 131–141, Feb. 2010.

[7] W. Liu, H. Liu, D. Tao, Y. Wang, and K. Lu, "Multiview Hessian regularized logistic regression for action recognition," *Signal Process.*, vol. 110, pp. 101–107, 2015.

[8] B. Geng, L. Yang, C. Xu, and X.-S. Hua, "Content-aware ranking for visual search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 3400–3407.

[9] Y. Jing and S. Baluja, "VisualRank: Applying pagerank to large-scale image search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1877–1890, Nov. 2008.

[10] X. Tian, L. Yang, J. Wang, X. Wu, and X.-S. Hua, "Bayesian visual reranking," *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 639–652, Aug. 2011.

[11] L. Yang and A. Hanjalic, "Prototype-based image search reranking," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 871–882, Jun. 2012.

[12] R. Yan, A. G. Hauptmann, and R. Jin, "Multimedia search with pseudo-relevance feedback," in *Proc. 2nd Int. Conf. Image Video Retrieval*, 2003, pp. 238–247.

[13] J. Krapac, M. Allan, J. Verbeek, and F. Juried, "Improving web image search results using query-relative classifiers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 1094–1101.

[14] T. Joachims, "Optimizing search engines using clickthrough data," in *Proc. 8th ACM Int. Conf. Knowl. Discovery Data Mining*, 2002, pp. 133–142.

[15] X. Tian, Y. Lu, L. Yang, and Q. Tian, "Learning to judge image search results," in *Proc. 19th ACM Int. Conf. Multimedia*, 2011, pp. 363–872.

[16] J. Yu, D. Tao, M. Wang, and Y. Rui, "Learning to rank using user clicks and visual features for image retrieval," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 767–779, Apr. 2015.

[17] J. Yu, Y. Rui, and B. Chen, "Exploiting click constraints and multi-view features for image re-ranking," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 159–168, Jan. 2014.

[18] J. Yu, Y. Rui, and D. Tao, "Click prediction for web image reranking using multimodal sparse coding," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2019–2032, May 2014.

[19] S. Cronen-Townsend, Y. Zhou, and W. B. Croft, "Predicting query performance," in *Proc. 25th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2002, pp. 299–306.

[20] K.-L. Kwok, L. Grunfeld, H. L. Sun, and P. Deng, "Trec 2004 robust track experiments using pircs," in *Proc. TREC*, 2004, pp. 1–7.

[21] B. He and I. Ounis, "Inferring query performance using pre-retrieval predictors," in *Proc. Symp. String Process. Inf. Retrieval*, 2004, pp. 43–54.

[22] E. C. Jensen, S. M. Beitzel, D. Grossman, O. Frieder, and A. Chowdhury, "Predicting query difficulty on the web by learning visual clues," in *Proc. ACM Conf. Res. Develop. Inf. Retrieval*, 2005, pp. 615–616.

[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," in *Proc. 28th Annu. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2005, pp. 512–519.

[24] C. Hauff, V. Murdock, and R. Baeza-Yates, "Improved query difficulty prediction for the web," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2008, pp. 439–448.

[25] H. Imran and A. Sharan, "Co-occurrence based predictors for estimating query difficulty," in *Proc. IEEE Int. Conf. Data Mining Workshops*, 2010, pp. 867–874.

[26] J. A. Aslam and V. Pavlu, "Query hardness estimation using Jensen-Shannon divergence among multiple scoring functions," in *Proc. 29th Eur. Conf. IR Res.*, 2007, pp. 198–209.

[27] S. Rudinac, M. Larson, and A. Hanjalic, "Exploiting result consistency to select query expansions for spoken content retrieval," in *Proc. 32nd Eur. Conf. IR Res.*, 2010, pp. 645–648.

[28] G. Amati, C. Carpineto, and G. Romano, "Query difficulty, robustness, and selective application of query expansion," in *Proc. 26th Eur. Conf. IR Res.*, 2004, pp. 127–137.

[29] Y. Zhou and B. Croft, "Ranking robustness: A novel framework to predict query performance," in *Proc. 15th ACM Int. Conf. Inf. Knowl. Manage.*, 2006, pp. 567–574.

[30] D. Carmel, E. Yom-Tov, A. Darlow, and D. Pelleg, "What makes a query difficult?" in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2006, pp. 390–397.

[31] Y. Zhou and W. B. Croft, "Query performance prediction in web search environments," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2007, pp. 543–550.

[32] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley-Interscience, 1991.

[33] J. Lin, "Divergence measures based on the Shannon entropy," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 145–151, Jan. 1991.

[34] J. He, M. Larson, and M. De Rijke, "Using coherence-based measures to predict query difficulty," in *Proc. 30th Eur. Conf. IR Res.*, 2008, pp. 689–694.

[35] X. Xing, Y. Zhang, and M. Han, "Query difficulty prediction for contextual image retrieval," in *Proc. Eur. Conf. IR Res.*, 2010, pp. 581–585.

[36] Y. Li, B. Geng, D. Tao, Z.-J. Zha, L. Yang, and C. Xu, "Difficulty guided image retrieval using linear multiple feaure embedding," *IEEE Trans. Multimedia*, vol. 14, no. 6, pp. 1618–1630, Dec. 2012.

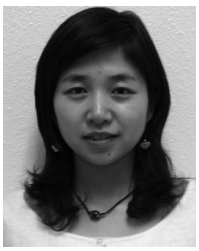
[37] S. Rudinac, M. Larson, and A. Hanjalic, "Leveraging visual concepts and query performance prediction for semantic-theme-based video retrieval," *Int. J. Multimedia Inf. Retrieval*, vol. 1, no. 4, pp. 263–280, 2012.

[38] (2006). Trecvid video retrieval evaluation [Online]. Available: hppt://www-nlpir.nist.gov/projects/trecvid/

[39] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, 2002.

[40] L. Yang, L. Wang, B. Geng, and X.-S. Hua, "Query sampling for ranking learning in web search," in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2009, pp. 754–755.
[41] X. Tian, D. Tao, X.-S. Hua, and X. Wu, "Active reranking for web image search," *IEEE Trans. Image Process.*, vol. 19, no. 3, pp. 805–820, Mar. 2010.
[42] X. Geng, T.-Y. Liu, T. Qin, A. Arnold, H. Li, and H.-Y. Shum, "Query dependent ranking using k-nearest neighbor," in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2008, pp. 115–122.
[43] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. IEEE 9th Int. Conf. Comput. Vis.*, 2003, pp. 1470–1477.
[44] J. G. Carbonell, Y. Yang, R. E. Frederking, R. D. Brown, Y. Geng, and D. Lee, "Translingual information retrieval: A comparative evaluation," in *Proc. Int. Joint Conf. Artif. Intell.*, 1997, pp. 708–714.
[45] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1065–1076, 1962.
[46] J. Deng, A. C. Berg, K. Li, and F.-F. Li, "What does classifying more than 10,000 image categories tell us?" in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 71–84.
[47] S. M. Kendall and J. D. Gibbons, *Rank Correlation Methods*. Edward Arnold: London, 1990.
[48] J. D. Gibbons and S. Chakraborty, *Nonparametric Statistical Inference*. New York, NY, USA: Marcel Dekker, 1992.
[49] R. Gerrig and P. Zimbardo, *Psychology and Life*, 16 Ed. Boston, MA, USA: Allyn & Bacon, 2001.
[50] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Comput. Surv.*, vol. 40, no. 2, pp. 1–60, 2008.
[51] R. Baeza-yates, C. Hurtado, and M. Mendoza, "Query recommendation using query logs in search engines," in *Proc. Int. Conf. Extending Database Technol.*, 2004, pp. 588–596.
[52] R. Jones, B. Rey, and O. Madani, "Generating query substitutions," in *Proc. 15th Int. Conf. World Wide Web*, 2006, pp. 387–396.
[53] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, "Wordnet: An on-line lexical database," *Int. J. Lexicography*, vol. 3, no. 4, pp. 235–244, 1990.

Xinmei Tian (M'13) received the BE and PhD degrees from the University of Science and Technology of China in 2005 and 2010, respectively. She is an associate professor in the CAS Key Laboratory of Technology in Geo-spatial Information Processing and Application System, University of Science and Technology of China. Her current research interests include multimedia information retrieval and machine learning. She received the Excellent Doctoral Dissertation of Chinese Academy of Sciences award in 2012 and the Nomination of National Excellent Doctoral Dissertation award in 2013. She is a member of the IEEE.



Yijuan Lu (M'05) received the PhD degree in computer science from the University of Texas at San Antonio in 2008. She is an associate professor in the Department of Computer Science, Texas State University. Her current research focuses on multimedia information retrieval, computer vision, and machine learning. She has published extensively and has served on program and organizing committee for many international conferences. She received 2013 ICME Best Paper Award, 2012 ICIMCS Best Paper Award, and is one of top winners of 2013 Eurographics SHERC competitions in large-scale sketch-based 3D retrieval track, range scan track, and low-cost depth-sensing camera track. She received 2015 Texas State Presidential Distinction Award, 2014 College Achievement Award, and was not is 2012 dean nominee for Texas State Presidential Award for Excellence in Scholarly/Creative Activities, and was not is a nominee for 2008 Microsoft Research Faculty Summit. Her research has been funded by National Science Foundation, Texas Department of Transportation, Department of Defense, Army Research, and Texas State. She is a member of the IEEE.

Nate Stender is with the Department of Computer Science, University of Nebraska-Lincoln, Lincoln. He was a summer research intern at Adventium Labs during year 2013. His research interests include machine learning, artificial intelligence, and computer vision.

Linjun Yang (M'08) received the PhD degree from Delft University of Technology, The Netherlands, in 2013. He is currently a senior development lead in Microsoft, focusing on developing state-of-the-art image understanding technologies to improve multimedia search experience. He has authored more than 50 refereed papers and received the Best Paper Award from ACM Multimedia 2009 and the Best Student Paper Award from ACM Conference on Information and Knowledge Management 2009. He is a member of the IEEE.

Dacheng Tao (F'14) is a professor of computer science with the Centre for Quantum Computation & Intelligent Systems, and the Faculty of Engineering and Information Technology in the University of Technology, Sydney. He mainly applies statistics and mathematics to data analytics and his research interests spread across computer vision, data science, image processing, machine learning, neural networks, and video surveillance. His research results have expounded in one monograph and 100+ publications at prestigious journals and prominent conferences, such as *IEEE TPAMI*, *TNNLS*, *TIP*, *TCYB*, *JMLR*, *IJCV*, NIPS, ICML, CVPR, ICCV, ECCV, AISTATS, ICDM; and ACM SIGKDD, with several Best Paper Awards, such as the Best Theory/Algorithm Paper Runner Up Award in IEEE ICDM'07, the Best Student Paper Award in IEEE ICDM'13, and the 2014 ICDM 10 Year Highest-Impact Paper Award. He is a fellow of the IEEE.