Learning to Rank with Graph Consistency

Bo Geng [†‡*], Linjun Yang [‡], Xian-Sheng Hua [‡]
[†] Key Laboratory of Machine Perception (Ministry of Education), Peking University, Beijing, 100871, P.R.China
[‡] Microsoft Research Asia, Beijing, 100190, P.R.China
bogeng@pku.edu.cn, {linjuny, xshua}@microsoft.com

ABSTRACT

The ranking models of existing image search engines are generally based on associated text while the image visual content is actually neglected. Imperfect search results frequently appear due to the mismatch between the textual features and the actual image content. Visual reranking, in which visual information is applied to refine text based search results, has been proven to be effective. However, the improvement brought by visual reranking is limited, and the main reason is that the errors in the text-based results will propagate to the refinement stage. In this paper, we propose a Content-Aware Ranking model based on "learning to rank" framework, in which textual and visual information are simultaneously leveraged in the ranking learning process. We formulate the Content-Aware Ranking learning based on large margin structured output learning, by modeling the visual information into a regularization term. The direct optimization of the learning problem is nearly infeasible since the number of constraints is huge. The efficient cutting plane algorithm is adopted to learn the model by iteratively adding the most violated constraints. Extensive experimental results on a large-scale dataset collected from a commercial Web image search engine demonstrate that the proposed ranking model significantly outperforms the state-of-the-art ranking and reranking methods.

Categories and Subject Descriptors

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

General Terms

Algorithms, Theory, Experimentation, Performance.

Keywords

Learning to Rank, Image Search Reranking, Content-Aware Ranking Model

1. INTRODUCTION

As the rapid evolution of imaging technologies and the emergence of image sharing websites such as Flickr [1], images are substantially easy to be generated and spread all around the Web, as well as consequently accumulated at a prodigious speed. The demand of searching desired images from vast amounts of candidates on the Web drives the academic as well as the industry continuous interests on developing effective image search techniques.

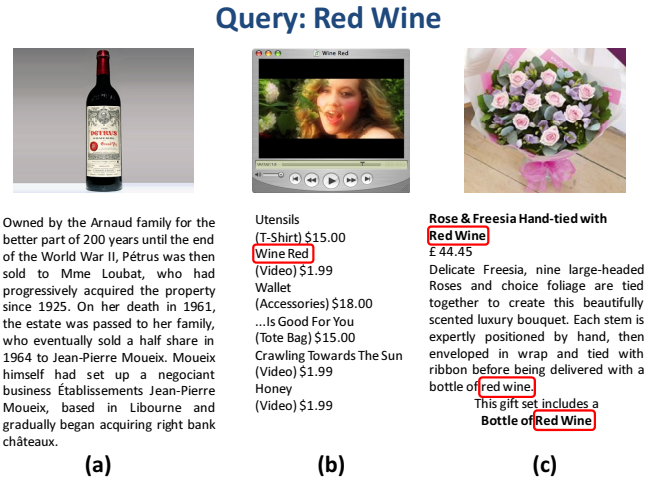

Figure 1: Example images with surronding text features for the query "Red Wine"

The frequently-used existing commercial image search engines, such as Google [2], Yahoo [4] and Live [3], are all based on indexing textual metadata. That is, only the textual information associated with the images in the Web page, e.g., title, anchor text, surrounding text, and etc., are utilized to identify the images. And then the well known methods in information retrieval, such as *tf-idf* [5] and *okapi BM25* [26], are typically employed to rank the images, based on these textual metadata.

Alternatively, a more effective ranking approach that can be adopted is *learning to rank*, which is now intensively studied in both information retrieval and machine learning communities. In this approach, the query dependent features for each image, i.e., the features to describe the correlation between a query and an image, are extracted from various textual sources, e.g., the term frequency of query keywords in the anchor text, title, surrounding text, and etc. Then a ranking function which elaborately combines all the different features is learned from the human labeled training set [8, 9, 10, 13, 20, 32, 34].

However, there are essential difficulties for image search based on only textual information, because the associated

*This work was performed when Bo Geng was visiting Microsoft Research Asia as a research intern.

text often mismatches the actual image content. For example, Fig. 1 (a) shows a "Red Wine" related images while the query term "Red Wine" is missing in the textual description. In Fig. 1 (b), the image is easy to be predicted as relevant to "Red Wine" by using only the surrounding text, as it contains the query term, though with a different meaning. Fig.1 (c) illustrates another typical mismatching case between the image and the associated text, where "Red Wine" is mentioned but has nothing to do with the corresponding image. It can be seen that purely using textual description to learn ranking function often makes mistakes.

We argue that an effective approach to address the above difficulty is to incorporate visual information of the images into the ranking learning scheme. The straightforward method is to extract the query dependent features to describe the correlation between the query words and images' visual content and then concatenate them to the textual features. Concept detection results are such features that can be easily employed in existing ranking models [22] [23][27]. However, current concept detectors are still far from satisfactory in practice, in terms of both accuracy and scalability. It is infeasible to train an acceptable concept detector for each query term due to the complexity of both the semantic concepts and the visual content. Moreover, the number of concept detectors is not scalable due to the high labeling and training cost [17].

Another way to incorporate visual information is *visual reranking* [15] [16][17][29][33], in which visual features are applied to refine text based search results in a separated second step. A basic assumption in visual reranking is that visually similar images can boost each other in the reranked list, which has been proven to be effective to improve the search relevance. However, visual reranking only uses the visual information to refine the text based results, instead of to assist the learning process of the text-based ranking model. If the learned text-based model is biased or overfitted, reranking step will suffer from the error propagated from the initial ranks, thus the performance improvement is limited. In this paper, we propose a novel ranking model which incorporates the low-level visual features directly into the ranking learning objective. We call it *Content-Aware Ranking learning*, as it takes visual content related features into account to enhance the ranking model during the ranking learning process.

The proposed Content-Aware Ranking model can address the drawbacks of visual reranking, i.e., the error propagation and the inability to help text-based ranking learning as aforementioned, as it is a unified model which simultaneously leverages the textual relevance feature and visual features in the ranking learning process. That is, combining visual features into ranking learning will result in a more robust and accurate ranking model as noises in textual features can be suppressed by visual content information. Furthermore, Content-Aware Ranking is easy to scale up as it utilizes the low-level visual features directly, without requiring a large amount of trained concept detectors.

Learning Content-Aware Ranking model is a challenging problem. First, the to-be-learned ranking list is structured, i.e., the ranking of the images are interdependent of each other and cannot be determined independently. The conventional learning methods, such as Support Vector Machine [35], cannot deal with the structured output. Second, since the visual features are query independent, unlike the textual relevance features, the conventional learning to rank methods cannot be adopted directly as well. Therefore, we propose a learning method within the *large margin structured output learning* framework [12], which incorporates the visual information using a regularization framework. The structure within the ranking list is modeled in the loss function. The learning problem can be solved by quadratic programming, but as the solution space of the optimization problem is very large, the cutting plane method is adopted to reduce the number of constraints by iteratively adding the most violated constraints until a desired accuracy is achieved.

The rest of the paper is organized as follows: Section 2 reviews the related work, including learning to rank, concept detection for image search, and image search reranking. The formulation of the proposed Content-Aware Ranking learning is introduced and analyzed in Section 3. In Section 4, we detail the optimization techniques to solve the learning problem. The learning complexity analyses are presented in Section 5, followed by experimental results in Section 6 and conclusions in Section 7.

2. RELATED WORK

Our work is motivated by learning to rank, which is recently intensively studied to solve the document ranking problem. A large amount of algorithms have been proposed. Some of them transform the ranking problem into pairwise classification, which takes a pair of documents as a sample, with the label taken as the preference between them, and then binary classification methods can be adapted, e.g., RankSVM [20], RankBoost [13], RankNet [9] and etc. The others take the structure of ranking list into consideration and directly optimize the evaluation measures such as Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG). The methods include ListNet [10], SVM^{M}ap [34], PermuRank [32], LambdaRank [8] and etc. In this paper, the proposed method is different from the conventional learning to rank methods in that it takes not only the textual relevance features but also query independent low-level visual features into the ranking model.

There are various works on incorporating the visual content into image search, which can be divided into concept detection based ranking and visual reranking. For concept detection based ranking methods, firstly the concept models are learned for a selected concept ontology, built from the commonly used query keywords [12][22][23][27]. Then when users submit a query it will be mapped to the related concepts and the corresponding concept detection results will be combined with the textual features to rank the images [24][31]. However, the developments of both image low-level features and machine learning techniques are yet incapable of building robust concept detectors to predict high-level concepts accurately. Secondly, the concept detector based methods cannot scale up, as the number of concepts is huge. Furthermore, the concept ontology itself is hard to build for effective use in search. Finally, query mapping is generally unsatisfactory.

The image search reranking methods can be categorized into three classes: (1) classification-based methods, where the pseudo-positive and pseudo-negative samples are selected from the initial text search results and then a binary classifier is trained based on the pseudo-labeled samples to predict the reranking list [33]; (2) clustering-based methods, where

each sample is labeled with a soft pseudo label based on the initial text search ranking list, then optimal clustering is selected by maximizing the mutual information between the clusters and labels, and the final reranked list is obtained based on the cluster information [15]; (3) graph-based methods, where a graph is constructed with the images as nodes and their visual similarity as edges, and the reranking is performed by propagating the ranking scores among each other [16][19][29]. Recent studies show that the graph-based algorithms with pair-wise ranking distance is superior to the others [29].

3. CONTENT-AWARE RANKING MODEL

Before introducing the proposed Content-Aware Ranking, we firstly give the denotations here. Suppose we are given the query set $\mathcal{Q}$ for the ranking model learning, where for each query $q^i = \{\mathbf{x}^i, \mathbf{v}^i, \mathbf{y}^i\} \in \mathcal{Q}$, $\mathbf{x}^i = [\mathbf{x}^i_1, \ldots, \mathbf{x}^i_{N^i}]^T \in \mathcal{X}$ are the query dependent textual features of the corresponding query image pairs, $\mathbf{v}^i = [\mathbf{v}^i_1, \ldots, \mathbf{v}^i_{N^i}]^T \in \mathcal{V}$ denote the image visual features, $\mathbf{y}^i = [y^i_1, \ldots, y^i_{N^i}]^T \in \mathcal{Y}$ represent the ranking of the images corresponding to the query labeled by human oracle, and N^i stands for the number of involved images. Denote the target ranking function as $f : \mathcal{X} \times \mathcal{V} \to \mathcal{Y}$, which maps the joint textual and visual feature spaces $\mathcal{X} \times \mathcal{V}$ to the ranking space $\mathcal{Y}$. Thus, the goal of ranking learning is to find the optimal ranking function$f$, so that the expected ranking loss in the training set $\mathcal{Q}$,

$$\mathfrak{R}^{\Delta}_{\mathcal{Q}}(f) = \frac{1}{|\mathcal{Q}|}\sum_{i=1}^{|\mathcal{Q}|}\Delta(\mathbf{y}^i, f(\mathbf{x}^i, \mathbf{v}^i))$$

is minimized, where the function $\Delta(\mathbf{y}^i, \widehat{\mathbf{y}})$ measures the loss of the ranking output $\widehat{\mathbf{y}} = f(\mathbf{x}^i, \mathbf{v}^i)$ with the ground-truth $\mathbf{y}^i$.

3.1 Ranking as Structured Output

As discussed before, ranking list is structured since the ranks of images are interdependent. Hence, the large margin structured output learning framework is used here to model the ranking learning problem. In this subsection we will focus mainly on the ranking using textual features, while the visual information will be incorporated in the next subsection. Given the query with textual feature $\mathbf{x}^i$, the ranking function can be defined as:

$$\widehat{\mathbf{y}} = f(\mathbf{x}^i) = \text{argmax}_{y\in\mathcal{Y}} F(\mathbf{x}^i, \mathbf{y}; \mathbf{w}) \tag{1}$$

where $\mathbf{w}$ is the parameter. $F(\mathbf{x}^i, \mathbf{y}; \mathbf{w})$ can be defined as a linear function of $\mathbf{w}$ in the following,

$$F(\mathbf{x}^i, \mathbf{y}; \mathbf{w}) = \mathbf{w}^T\Psi(\mathbf{x}^i, \mathbf{y}), \tag{2}$$

where $\Psi(\mathbf{x}^i, \mathbf{y})$ is a joint feature map by mapping the textual feature $\mathbf{x}^i$ and the ranking prediction $\mathbf{y}$ into real values. In this paper, we design a feature mapping function as $\Psi(\mathbf{x}^i, \mathbf{y}) = \sum_{j=1}^{N^i}\mathbf{x}^i_j y_j$, and then the Problem 2 is transformed into:

$$\widehat{\mathbf{y}} = \text{argmax}_{\mathbf{y}\in\mathcal{Y}}(\mathbf{z}^i)^T\mathbf{y} \tag{3}$$

where $\mathbf{z}^i = [\mathbf{w}^T\mathbf{x}^i_1, \ldots, \mathbf{w}^T\mathbf{x}^i_{N^i}]^T$ can be regarded as the score list of the images for the query q^i. The problem (3) can be interpreted from the geometrical viewpoint. Actually (3) maximizes the cosine of the angle between the score list $\mathbf{z}^i$ and $\mathbf{y}$, which tends to make the direction of $\mathbf{y}$ comply with that of $\frac{\mathbf{z}^i}{\|\mathbf{z}^i\|}$. It's straightforward that for a fixed $\mathbf{w}$, the solution to (3) leads $\widehat{\mathbf{y}}$ to be the ranking list sorted according to the derived relevance scores, i.e., $\mathbf{z}^i = \mathbf{w}^T\mathbf{x}^i$.

3.2 Content-Aware Ranking Model

Since the textual features are normally noisy, the text only ranking model is insufficient for image search, as discussed before. We propose to jointly utilize the textual and visual content features to boost the ranking model learning. It is assumed that the relevant images for a query should have the visual consistency property, i.e., similar images share similar ranking output. Based on the assumption, we propose to learn a ranking list that not only employs the textual relevance features, but also possesses a high visual consistency. Inspired by Laplacian Eigenmaps [6], the proposed Content-Aware Ranking model takes the following form:

$$\begin{aligned}\widehat{\mathbf{y}} &= \text{argmax}_{\mathbf{y}\in\mathcal{Y}} F(\mathbf{w}, \mathbf{x}^i, \mathbf{v}^i, \mathbf{y}) \\ &= \text{argmax}_{\mathbf{y}\in\mathcal{Y}}\mathbf{w}^T\Psi(\mathbf{x}^i, \mathbf{y}) - \gamma\sum_{m,n=1}^{N^i}\mathbf{G}^i_{mn}(y_m - y_n)^2\end{aligned} \tag{4}$$

where $\gamma > 0$ is a trade-off parameter to balance the textual relevance $\mathbf{w}^T\Psi(\mathbf{x}^i, \mathbf{y})$ and the visual consistency term $\sum_{m,n=1}^{N^i}\mathbf{G}^i_{mn}(y_m - y_n)^2$. $\mathbf{G}^i$ denotes the adjacency graph which measures the similarities between each pair of images, with the elements defined as:

$$\mathbf{G}^i_{mn} = \begin{cases}\text{Sim}(\mathbf{v}^i_m, \mathbf{v}^i_n), & \text{if } \mathbf{v}^i_n \text{ is in the } \textit{KNN} \text{ of } \mathbf{v}^i_m \\ 0, & \text{otherwise.}\end{cases}$$

Here, $\text{Sim}(\mathbf{v}^i_m, \mathbf{v}^i_n)$ is the similarity between $\mathbf{v}^i_m$ and $\mathbf{v}^i_n$, and $\mathbf{G}^i_{mn}$ is a sparse graph by employing KNN (*k-nearest-neighbor*) strategy. The minimization of the visual consistency term $\sum_{m,n=1}^{N^i}\mathbf{G}^i_{mn}(y_m - y_n)^2$ can be taken as a graph based manifold regularization [7], which drives that visually similar images are assigned with similar rank predictions. Based on the regularization theory, the solution of an ill-posed problem can be approximated from variational principle, which contains both the data and the prior smoothness information [14]. The visual similarity information can be taken as the auxiliary prior knowledge, which is able to reduce the over-fitting problem and boost the generalization ability of the ranking model. However, unlike conventional regularization frameworks, our model is regularized in a different feature space.

For a given $\mathbf{w}$, the optimization problem (4) can also be employed for reranking from the Bayesian reranking perspective. The first term can be regarded as the ranking distance while the second term is the visual consistency. Moreover, Content-Aware Ranking is superior to reranking. Generally reranking can be thought of as a two-stage method. That is, a textual ranking model is learned and then based on it reranking is performed. Reranking only takes the visual information into consideration during the ranking prediction, while cannot help learn a better textual ranking model. In Content-Aware Ranking, since the visual content is unified into the ranking model as well as textual

Algorithm 1 Cutting plane algorithm for content-aware ranking learning

Input: $\mathcal{Q}, C, \gamma$
$\mathcal{W}^i \leftarrow \emptyset$ for all $i = 1, \cdots, |\mathcal{Q}|$
repeat
  for $i = 1, \cdots, |\mathcal{Q}|$ **do**
    $H(\mathbf{y}; \mathbf{w}) \equiv \Delta(\mathbf{y}^i, \mathbf{y}) + F(\mathbf{w}, \mathbf{x}^i, \mathbf{v}^i, \mathbf{y})$
    Compute $\widehat{\mathbf{y}} = \text{argmax}_{\mathbf{y} \in \mathcal{Y}} H(\mathbf{y}; \mathbf{w})$
    Compute $\xi^i = \max\{0, \max_{\mathbf{y} \in \mathcal{W}^i} H(\mathbf{y}; \mathbf{w})\}$
    if $H(\widehat{\mathbf{y}}; \mathbf{w}) > \xi^i + \epsilon$ **then**
      $\mathcal{W}^i \leftarrow \mathcal{W}^i \cup \{\widehat{\mathbf{y}}\}$
      $\mathbf{w} \leftarrow$ optimize learning problem over $\mathcal{W} = \bigcup_i \mathcal{W}^i$
    end if
  end for
until no $\mathcal{W}^i$ has changed during iteration.

features, the visual information will help learn a better ranking model, in addition to boosting the ranking prediction.

3.3 The Learning Problem

To estimate the model parameters $\mathbf{w}$ in (4), we adopt the *large margin structured output learning* [30] framework, which can deal with the learning of complex and structured outputs like trees, sequences and sets, and ranking lists. Given the labeled training set $\mathcal{Q}$, we want to learn a weight vector $\mathbf{w}$ so that the ranking model can perfectly predict the ranks of the images for the queries in $\mathcal{Q}$, i.e.,

$$\begin{aligned} &\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 \\ &s.t. \forall q^i \in \mathcal{Q}, \forall \mathbf{y} \neq \mathbf{y}^i, \\ &\quad F(\mathbf{w}, \mathbf{x}^i, \mathbf{v}^i, \mathbf{y}^i) - F(\mathbf{w}, \mathbf{x}^i, \mathbf{v}^i, \mathbf{y}) \geq 1 \end{aligned} \tag{5}$$

In order to accommodate the noises in the training data, the slack variables are introduced to make the above hard constraints soft. Thus, the proposed learning problem is defined as follows:

$$\begin{aligned} &\min_{\mathbf{w}, \xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum \xi^i \\ &s.t. \forall q^i \in \mathcal{Q}, \xi^i \geq 0, \forall \mathbf{y} \neq \mathbf{y}^i, \\ &\quad F(\mathbf{w}, \mathbf{x}^i, \mathbf{v}^i, \mathbf{y}^i) - F(\mathbf{w}, \mathbf{x}^i, \mathbf{v}^i, \mathbf{y}) \geq \Delta(\mathbf{y}^i, \mathbf{y}) - \xi^i \end{aligned} \tag{6}$$

where $C > 0$ is the trade-off parameter to balance the model complexity $\|\mathbf{w}\|^2$ and the upper bound of the prediction loss $\sum \xi^i$; $\Delta(\mathbf{y}^i, \mathbf{y})$ is the ranking loss function to measure the loss between the prediction $\mathbf{y}$ and the ground truth $\mathbf{y}^i$, as we discussed before. During the learning phase, if the prediction is incorrect, *i.e.* $F(\mathbf{w}, \mathbf{x}^i, \mathbf{v}^i, \mathbf{y}) > F(\mathbf{w}, \mathbf{x}^i, \mathbf{v}^i, \mathbf{y}^i)$, then to satisfy the constraints, the corresponding slack variable ξ^i must be at least $\Delta(\mathbf{y}^i, \mathbf{y})$. For noisy samples we will favor the prediction ranking, though not, but similar in loss to, the ground-truth one. This is reasonable since even though the prediction is wrong we still desire the one which is as similar to the ground-truth as possible.

4. OPTIMIZATION

The direct optimization of (6) is nontrivial, because the number of the constraints is exponential. As in [30], the cutting plane method is adopted to solve it, by iteratively finding a small set of constraints and solving the small-scale problem until the stop condition is satisfied. The algorithm starts with an empty constraint set, and iteratively finds the most violated prediction $\widehat{\mathbf{y}}$ for each query q^i. If the corresponding constraint is violated by more than a predefined threshold ϵ for $\widehat{\mathbf{y}}$, it will be added into the working set $\mathcal{W}^i$ for query q^i, and then the problem is solved with the added constraints for all the queries, i.e. $\mathcal{W} \leftarrow \cup_i \mathcal{W}^i$. The detailed procedure of the algorithm is shown in Algorithm 1. Theorem 1 shows that the algorithm is guaranteed to converge within a polynomial number of steps for a pre-given tolerance ϵ.

THEOREM 1. *Denote* $\bar{R} = \max_{i,y} \|\Psi(\boldsymbol{x}^i, \boldsymbol{y}^i) - \Psi(\boldsymbol{x}^i, \boldsymbol{y})\|$, $\bar{\Delta} = \max_{i,y} \Delta(\boldsymbol{y}^i, \boldsymbol{y}) + \gamma \sum_{m,n=1}^{N^i} \boldsymbol{G}^i_{mn} \left((\boldsymbol{y}^i_m - \boldsymbol{y}^i_n)^2 - (y_m - y_n)^2\right)$, *and for any* $\epsilon > 0$, *Algorithm 1 terminates after incrementally adding at most*

$$\max\left(\frac{2n\bar{\Delta}}{\epsilon}, \frac{8C\bar{\Delta}\bar{R}^2}{\epsilon^2}\right)$$

constraints to the working set $\mathcal{W}$ *for solving (6).*

To implement Algorithm 1, we need to solve two subproblems, i.e., finding the most violated prediction $\widehat{\mathbf{y}}$, and estimating the optimized model weight vector $\mathbf{w}$ under the current working set $\mathcal{W}$. The following two subsections will be devoted to solve these problems respectively.

4.1 Finding the Most Violated Prediction

In Algorithm 1, we need to find the most violated prediction and add it into the working constraint set, i.e., $\widehat{\mathbf{y}}$ satisfying the following formula:

$$\text{argmax}_{\mathbf{y} \in \mathcal{Y}} \Delta(\mathbf{y}^i, \mathbf{y}) - \gamma \sum_{m,n=1}^{N_i} \mathbf{G}^i_{mn}(y_m - y_n)^2 + \mathbf{w}^T \Psi(\mathbf{x}^i, \mathbf{y}) \tag{7}$$

In this paper, the loss is defined as the cosine similarity between the ground truth ranking $\mathbf{y}^i$ and the prediction one $\mathbf{y} \in \mathcal{Y}$, i.e.,

$$\Delta(\mathbf{y}^i, \mathbf{y}) = 1 - \frac{(\mathbf{y}^i)^T \mathbf{y}}{\|\mathbf{y}^i\|\|\mathbf{y}\|} \tag{8}$$

However, (7) is hard to solve because the variables are discrete ranking values, and the incorporation of the graph regularization term makes the problem even more difficult. Consequently, we relax the ranking list to be a real-value score list, so that the problem (7) is easier to be solved. However, there will be a trivial solution when all the elements of $\mathbf{y}$ are equal to positive infinity. Thereafter, we constrain the score list to be normalized, *i.e.*, $\|\mathbf{y}\| = \|\mathbf{y}^i\| = 1$. By adding the constraint and substituting the loss function (8) into (7), the following optimization problem is derived to find the most violated prediction,

$$\widehat{\mathbf{y}} = \text{argmax}_{\mathbf{y}\in\mathcal{Y}} -(\mathbf{y}^i)^T\mathbf{y} - \gamma \sum_{m,n=1}^{N^i} \mathbf{G}^i_{mn}(y_m - y_n)^2 + (\mathbf{z}^i)^T\mathbf{y}$$

$$= \text{argmax}_{\mathbf{y}\in\mathcal{Y}} \mathbf{c}^T\mathbf{y} - \gamma(2\sum_{m=1}^{N^i} \mathbf{D}^i_{mm}y_m^2 - 2\sum_{m,n=1}^{N^i} \mathbf{G}^i_{mn}y_m y_n)$$

$$= \text{argmin}_{\mathbf{y}\in\mathcal{Y}} 2\gamma\mathbf{y}^T\mathbf{L}^i\mathbf{y} - \mathbf{c}^T\mathbf{y} \quad s.t. \|\mathbf{y}\| = 1 \tag{9}$$

where $\mathbf{c} = \mathbf{z}^i - \mathbf{y}^i$, $\mathbf{L}^i = \mathbf{D}^i - \mathbf{G}^i$ is the Laplacian matrix of the graph, which is a symmetric positive semi-definite matrix, and $\mathbf{D}^i = \text{Diag}(\mathbf{d}^i)$ denotes the degree matrix as $\mathbf{d}^i = [\mathbf{d}^i_1, \ldots, \mathbf{d}^i_{N_i}]$ and $\mathbf{d}^i_m = \sum_{n=1}^{N^i} \mathbf{G}^i_{mn}$.

To solve the optimization problem (9), we introduce the Lagrange multiplier λ to eliminate the constraint, and because $\|\mathbf{y}\|^2 = \|\mathbf{y}\| = 1$, the problem can be transformed to,

$$\max_{\lambda\in\mathbb{R}} \min_{\mathbf{y}\in\mathcal{Y}} 2\gamma\mathbf{y}^T(\mathbf{L}^i + \lambda\mathbf{I})\mathbf{y} - \mathbf{c}^T\mathbf{y} - \lambda \tag{10}$$

where $\mathbf{I}$ is a $N^i \times N^i$ identity matrix.

In order to achieve a feasible solution to problem (10), we only need to consider the case that the matrix $\mathbf{L}^i + \lambda\mathbf{I}$ is positive semi-definite. Or it is easily derived that the optimal value of the inner minimization problem will be negative infinite. Suppose the eigen-decomposition of $\mathbf{L}^i$ is:

$$\mathbf{L}^i = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T \tag{11}$$

where $\mathbf{\Sigma}$ is a diagonal matrix with $\text{diag}(\mathbf{\Sigma}) = \{\lambda_1, \ldots, \lambda_{N^i} | \lambda_1 \geq \ldots \geq \lambda_{N^i}\}$ are the eigenvalues and $\mathbf{U}$ are the eigenvectors of $\mathbf{L}^i$. Because $\mathbf{L}^i$ is a graph Laplacian matrix, it has the smallest eigenvalue $\lambda_{N^i} = 0$ [6]. Thus, to make sure $\mathbf{L}^i + \lambda\mathbf{I} = \mathbf{U}(\mathbf{\Sigma} + \lambda\mathbf{I})\mathbf{U}^\mathbf{T}$ positive semidefinite, we need to add the constraint $\lambda \geq 0$.

When $\lambda > 0$, $\mathbf{L}^i + \lambda\mathbf{I}$ will be invertible. Taking the derivatives of the objective function in (10) w.r.t. $\mathbf{y}$, and set it to zero, we can get the solution of $\mathbf{y}$ as:

$$\widehat{\mathbf{y}} = \frac{1}{2}(\mathbf{L}^i + \lambda\mathbf{I})^{-1}\mathbf{c} \tag{12}$$

Substituting (12) back into (10), we can derive the dual form:

$$\max_{\lambda\in\mathbb{R}^+} -\frac{1}{4}\mathbf{c}^T(\mathbf{L}^i + \lambda\mathbf{I})^{-1}\mathbf{c} - \lambda \tag{13}$$

By utilizing the eigen decomposition of $\mathbf{L}^i$, (13) can be transformed to:

$$\max_{\lambda\in\mathbb{R}^+} -\frac{1}{4}\mathbf{c}^T\mathbf{U}(\mathbf{\Sigma} + \lambda\mathbf{I})^{-\mathbf{1}}\mathbf{U}^\mathbf{T}\mathbf{c} - \lambda \tag{14}$$

Then the following problem is derived,

$$\min_{\lambda\in\mathbb{R}^+} \sum_{j=1}^{N^i} \eta_j^2(\frac{1}{\lambda + \lambda_j}) + 4\lambda \tag{15}$$

where $\eta = \mathbf{U}^\mathbf{T}\mathbf{c}$. We denote $g(\lambda) = \sum_{j=1}^{N^i} \eta_j^2(\frac{1}{\lambda+\lambda_j}) + 4\lambda$ for abbreviation. There is no analytical solution for problem (15). Due that the second derivative $\frac{\partial^2 g(\lambda)}{\partial\lambda^2} = 2\sum_{j=1}^{N^i} \eta_j^2(\lambda + \lambda_j)^{-3} \geq 0$, we can conclude that $g(\lambda)$ is convex w.r.t. λ for $\lambda > 0$. In other words, the local minimum and global minimum of the problem (15) are the same for the given interval $\lambda \in (0, +\infty)$. Thus, we resort to the binary search technique [21] to find the optimal λ. The details of the algorithm are shown in Algorithm 2. Specifically, we recursively search for the optimal λ by bounding its feasible region, until the bounding of current feasible region is no more than ε. We can prove that the initial search upper bound $\widetilde{\Theta}_h = \frac{g(1)}{4}$, is sufficient to find the global optimal solution.

Algorithm 2 Finding the most violated prediction

Input: $\mathbf{G}^i, \mathbf{w}, \mathbf{x}^i, \mathbf{y}^i, \widetilde{\Theta}_h = \frac{g(1)}{4}, \varepsilon > 0$
Compute $\mathbf{L}^i$, η, and $\mathbf{\Sigma}$ according to (9)(11)(15)
$\lambda \leftarrow \text{BinarySearch}(\eta, \mathbf{\Sigma}, 0, \Theta_l, \Theta_h)$
$\widehat{\mathbf{y}} = \frac{1}{2}(\mathbf{L}^i + \lambda\mathbf{I})^{-1}\mathbf{c}$

Method $\lambda \leftarrow \text{BinarySearch}(\eta, \mathbf{\Sigma}, \Theta_l, \Theta_h)$
  $\lambda \leftarrow (\Theta_l + \Theta_h)/2$
  if $|\Theta_h - \Theta_l| < \varepsilon$ **then**
    Return λ
  end if
  $\tau \leftarrow (\Theta_h - \Theta_l)/6$
  $\delta_h \leftarrow g(\lambda + \tau)$
  $\delta_l \leftarrow g(\lambda - \tau)$
  if $\delta_h < \delta_l$ **then**
    Return $\text{BinarySearch}(\eta, \mathbf{\Sigma}, \lambda - \tau, \Theta_h)$
  else
    Return $\text{BinarySearch}(\eta, \mathbf{\Sigma}, \Theta_l, \lambda + \tau)$
  end if
end Method

PROPOSITION 1. *Setting the initial* $\widetilde{\Theta}_h = \frac{g(1)}{4}$ *is sufficient for the Algorithm 2 to find the global minimum.*

PROOF. For any $\lambda' > \widetilde{\Theta}_h = \frac{g(1)}{4}$,

$$g(\lambda') = \sum_{i=0}^{N^i} \eta_j^2(\frac{1}{\lambda' + \lambda_j}) + 4\lambda' > 4\lambda' > g(1) \geq \min_{\lambda\in\mathbb{R}^+} g(\lambda)$$

Consequently, $\widetilde{\Theta}_h$ is the sufficient upper bound for searching the global minimum of $g(\lambda)$ in Algorithm 2. □

4.2 Learning the Model Weight Vector

After selecting the most violated constraints as working set $\mathcal{W}^i$ we need to solve the optimization problem (6) in Algorithm 1 in each iteration. The optimization problem with the selected working set as follows,

$$\begin{aligned} &\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum \xi^i \\ &s.t. \forall q^i \in \mathcal{Q}, \forall \widehat{\mathbf{y}}^i_j \in \mathcal{W}^i, \xi^i \geq 0, \mathbf{w}^T\mathbf{u}^i_j \geq b^i_j - \xi^i \end{aligned} \tag{16}$$

where for $\widehat{\mathbf{y}}^i_j \in \mathcal{W}^i$, $\mathbf{u}^i_j = \Psi(\mathbf{x}^i, \mathbf{y}^i) - \Psi(\mathbf{x}^i, \widehat{\mathbf{y}})$, and $b^i_j = \Delta(\mathbf{y}^i, \widehat{\mathbf{y}}^i_j) + \gamma\sum_{m,n=1}^{N} \mathbf{G}^i_{mn}((y^i_m - y^i_n)^2 - (\widehat{\mathbf{y}}^i_{jm} - \widehat{\mathbf{y}}^i_{jn})^2)$. We denote $\mathcal{U} = \{\mathbf{u}^i_j | q^i \in \mathcal{Q}, \widehat{\mathbf{y}}^i_j \in \mathcal{W}^i\}$, and define the kernel matrix $\mathbf{K}$ as $K_{ij} = \langle \mathbf{u}_i, \mathbf{u}_j \rangle$ for $\forall \mathbf{u}_i \forall \mathbf{u}_j \in \mathcal{U}$. By introducing the Lagrange multiplier α^i_j for each constraint, we can obtain the dual problem of (16) as follows,

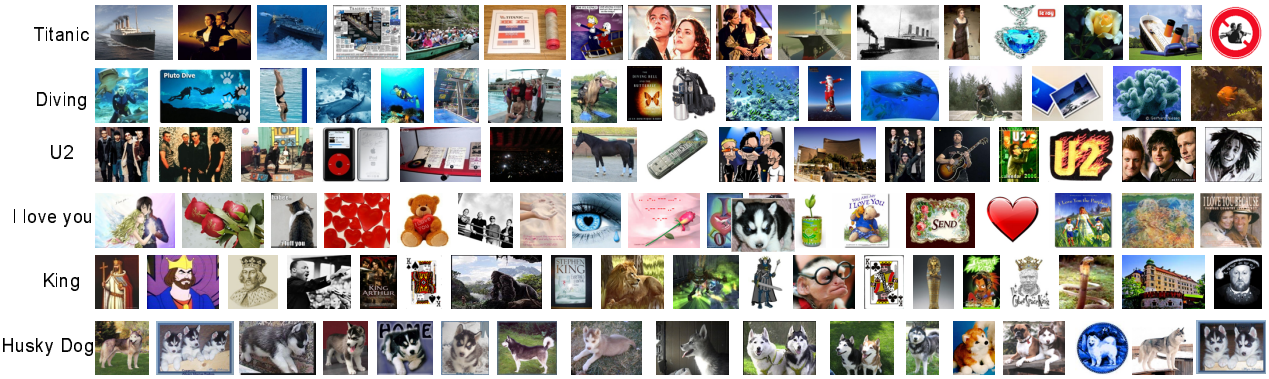

Figure 2: Example queries and associated images with different relevance degreees in the experiment dataset.

$$\max_{\alpha} \mathbf{b}^T\alpha - \frac{\mathbf{1}}{\mathbf{2}}\alpha^{\mathbf{T}}\mathbf{K}\alpha$$
$$\text{s.t. } \alpha \geq \mathbf{0} \text{ and } \sum_{j=1}^{|\mathcal{W}^i|} \alpha_j^i \leq C, \text{ for } \forall q^i \in \mathcal{Q} \quad (17)$$

This is a standard quadratic programming problem, which can be solved by using efficient algorithms such as Interior Point method. Based on the derivation the optimal solution $\mathbf{w}$ of the primal problem (6) is

$$\widetilde{\mathbf{w}} = \sum_{i=1}^{|\mathcal{Q}|}\sum_{j=1}^{|\mathcal{W}^i|} \alpha_j^i \mathbf{z}_j^i \quad (18)$$

4.3 Ranking Prediction

Given a new testing query q^t with textual and visual features $\{\mathbf{x}^t, \mathbf{v}^t\}$, the corresponding ranking can be predicted based on the learned model with parameter $\widetilde{\mathbf{w}}$. The prediction is performed by solving the problem (3) by replacing $\mathbf{w}$ with $\widetilde{\mathbf{w}}$, that is:

$$\widetilde{\mathbf{y}} = \text{argmax}_{\mathbf{y}\in\mathcal{Y}} \widetilde{\mathbf{w}}^T\Psi(\mathbf{x}^t, \mathbf{y}) - \gamma \sum_{m,n=1}^{N^t} \mathbf{G}_{mn}^t (y_m - y_n)^2 \quad (19)$$

We can adopt Algorithm 2 to solve it, but with a different definition of $\mathbf{c}$. Here $\mathbf{c} = \mathbf{z}^t = [\widetilde{\mathbf{w}}^T\mathbf{x}_1^t, \ldots, \widetilde{\mathbf{w}}^T\mathbf{x}_{N^t}^t]^T$ since no ranking loss is required. Finally, the ranking list is given by sorting images according to the predicted score list $\widetilde{\mathbf{y}}$ in the descending order.

Notice that since the predictive ranking of a new query involves all the images in the database, which may contains a huge amount of data resulting into a large graph structure G_{mn}^t, therefore the optimization of (19) may be inefficient. In practical applications, we can firstly utilize the textual based features to rank the images in the database, and then apply (19) to the top ranked images (typically no more than 1000). Experimentally, such a strategy is proven to be both effective and efficient.

5. COMPLEXITY ANALYSIS

We present the learning complexity analysis in this section. To simplify the discussion, here we suppose that all the queries possess the equivalent number of images to be ranked, i.e., for $\forall q^i \in Q, N^i = N$.

Firstly let's analyze the computational cost for Algorithm 2. The time cost of binary search is $O(N\log\frac{\widetilde{\Theta}_h}{\varepsilon})$ where $\log\frac{\widetilde{\Theta}_h}{\varepsilon}$ is the number of recursive depth for finding the solution. The algorithm of finding the most violated prediction y needs matrix multiplications, the computational cost of which is $O(N^3)$, or even $O(N^{2.8})$ using the Strassen Algorithm [28]. Thus, the total time cost of Algorithm 2 for all the queries in Q is about $O(|Q|N^{2.8})$.

For a given constraint set $\mathcal{W}$, learning the model weight vector $\mathbf{w}$ is a standard QP problem as described in Section 4.2. Thereafter, the computational cost will be $O(|\mathcal{W}|^3)$ using standard QP solver, or approximately $O(|\mathcal{W}|^{2.3})$ using SMO [25].

Algorithm 1 is guaranteed to terminate in a polynomial number of iterations [30]. In each iteration Algorithm 2 and QP Solver will be executed sequentially. Suppose the number of iterations in Algorithm 1 is T, its computational cost will be $O(T(|Q|N^{2.8} + |\mathcal{W}|^{2.3}))$, which is the total training time. Once the model is learned, making prediction is at the same cost as Algorithm 2, i.e., $O(N^{2.8})$ for each query.

6. EXPERIMENTS

In order to demonstrate the effectiveness of the proposed Content-Aware Ranking (CAR) model, we perform several experiments over a large-scale image search dataset crawled from the Web, and report the results together with some analysis and conclusions.

6.1 Dataset Description

We collect a large-scale image search dataset[1] from a commercial Web image search engine, which comprises 594 queries sampled from the query log. For each query at least 100 images are collected by sampling from the images whose associated text contains the query words. In total, there are 137348 query-image pairs, on average 250 images per query. Each image is labeled with three relevance degrees by the human oracle, according to its relevance to the corresponding query, namely "Not Relevant", "Relevant", and "Highly Relevant". The examples of queries and images are shown in Fig. 2.

For each query-image pair, various query dependent fea-

[1]We plan to release the dataset for academic use soon.

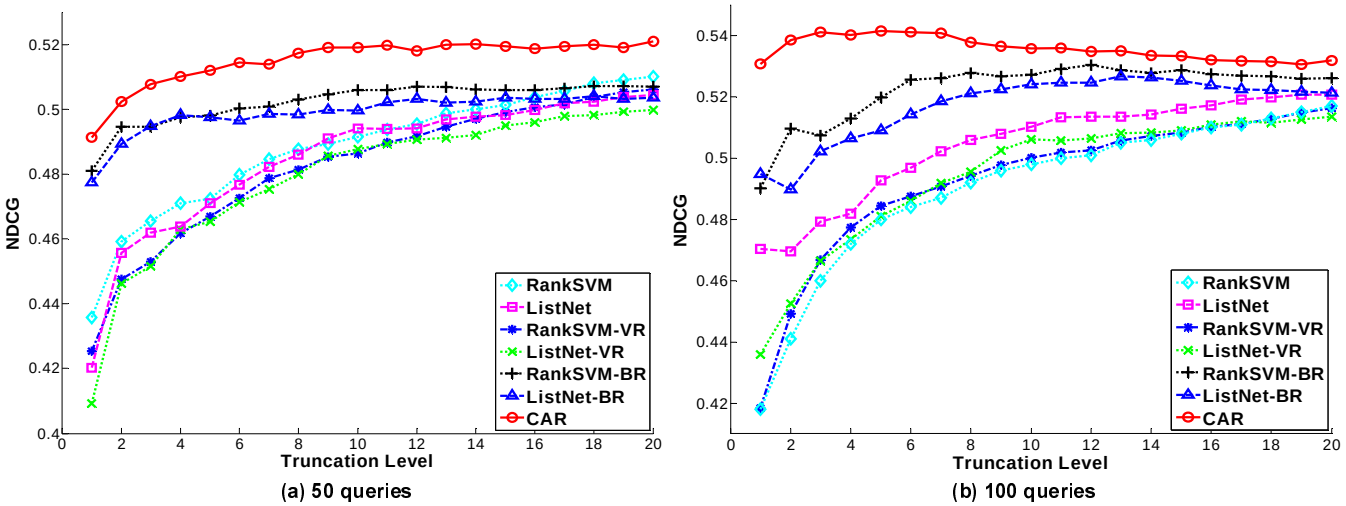

Figure 3: Performance comparisions of different algorithms: RankSVM, ListNet, RankSVM-VR, ListNet-VR, RankSVM-BR, ListNet-BR, and CAR, on 50 queries and 100 queries dataset respectively. (a) NDCG for 50 queries; (b) NDCG for 100 queries

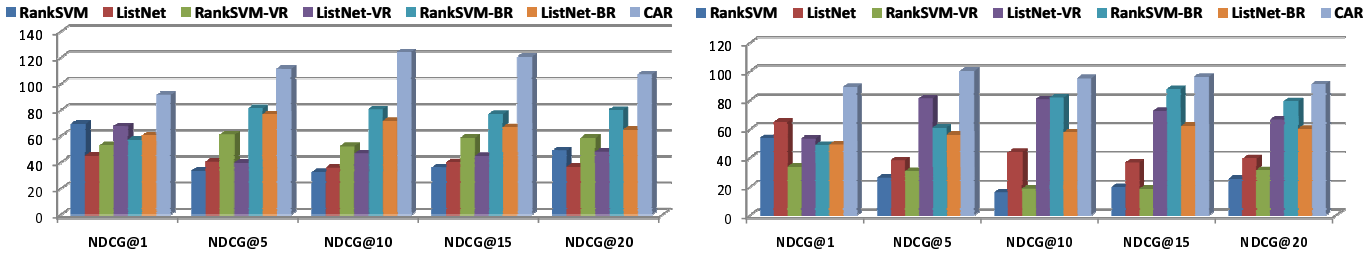

Figure 4: The number of best performed queries among different algorithms, on NDCG@$\{1, 5, 10, 15, 20\}$.

tures are extracted from different textual sources, e.g., the term frequency of query keywords in image anchor text, title, surrounding text, etc. Totally, there are 1011-dimensional textual features. Furthermore, we adopt RankBoost algorithm [13] to perform feature selection so that 164-dimensional features are finally selected to be used in the experiments.

To model the visual consistency, we extract several visual features that are widely used in computer vision, i.e., Attention Guided Color Signature, Color Spatialet, Wavelet, SIFT, Multi-Layer Rotation Invariant EOH (MRI-EOH), Histogram of Gradient (HoG), and Facial Feature. The distances computed on each feature are linearly combined as the ultimate distance between the images [11].

6.2 Experiment Settings and Evaluations

We compare the performance of CAR with textual ranking methods using the well-known learning to rank algorithms, i.e., RankSVM [20] and ListNet [10]. In addition, we also compare CAR with reranking methods including Visual Rank (VR) [19] and Bayesian Reranking (BR) [29]. For each reranking methods two kinds of text baselines are used, i.e., RankSVM and ListNet. Finally there are four combinations of reranking methods, denoted by RankSVM-VR, ListNet-VR, RankSVM-BR, and ListNet-BR respectively. Since it is difficult to label the training set for learning the concepts from the 594 queries and there are no appropriate concept ontology for Web image search, the learning to rank with concept detection method is not compared in this paper.

The data are split into four parts, training, validation A, validation B, and test. The validation A is used to validate the ranking learning, i.e., CAR, RankSVM, and ListNet; while Validation B is used to validate the model parameters in reranking, including VR and BR. Since the computational cost of RankSVM is very high, we only select 50 queries for the training. Then 50 queries are randomly selected from the data for Validation A and Validation B respectively. The remaining is used for test. In addition, we design another dataset which contains 100 training queries to compare the methods except RankSVM in a large training set.

The performance is evaluated based on Normalized Discounted Cumulative Gain (NDCG@k) [18]. For a given query q^i, NDCG is defined as:

$$\text{NDCG@}k = \frac{1}{Z_i} \sum_{j=1}^{k} \frac{2^{r(j)} - 1}{log(1+j)} \tag{20}$$

where $r(j)$ is the relevance level of the jth document, which is 0 for "Not Relevant", 2 for "Relevant", and 3 for "Highly Relevant" in our experiments. Z_i is the normalization coefficient to make the NDCG of a prefect ranking equals to 1, and k is the truncation level.

6.3 Experimental Results

The performances of different algorithms are shown in Fig.

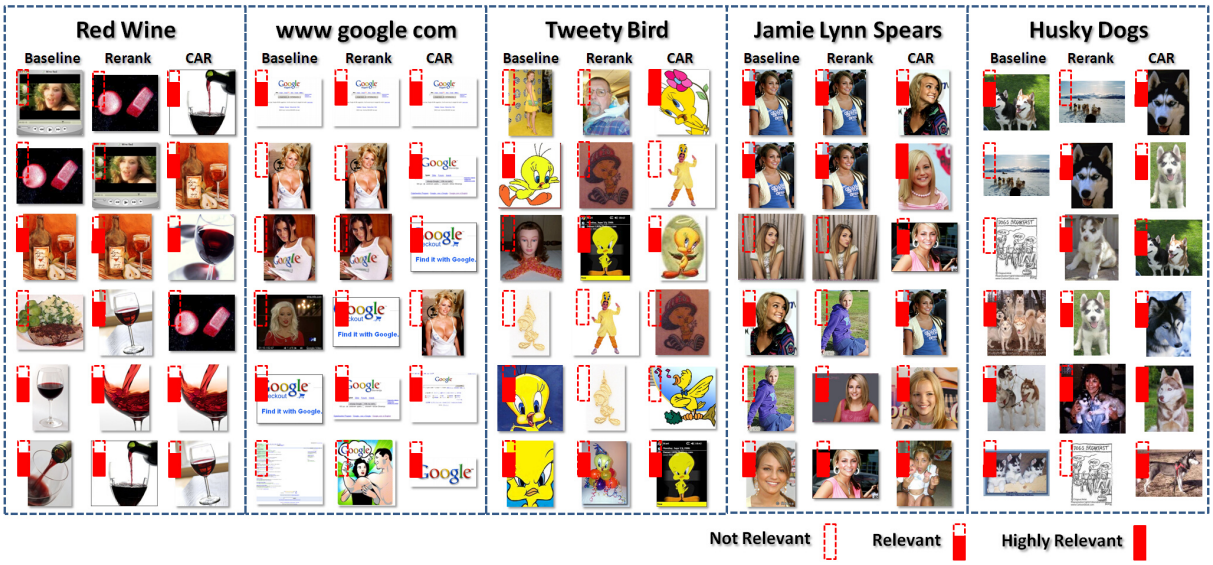

Figure 5: Ranking and reranking results for several example queries. Baseline: ListNet. Rerank: ListNet-BR. The queries and image results are excerpted from the test data.

Table 1: The quantitive performance comparisions of different algorithms, NDCG@1, 5, 10, 20.

	50 queries					100 queries				
	Baseline	Reranking		CAR		Baseline	Reranking		CAR	
		Perf	Gain	Perf	Gain		Perf	Gain	Perf	Gain
NDCG@1	0.4202	0.4775	13.64%	0.4913	16.92%	0.4703	0.4949	5.23%	0.5308	12.86%
NDCG@5	0.4710	0.4976	5.65%	0.5119	8.68%	0.4927	0.5091	3.33%	0.5415	9.90%
NDCG@10	0.4941	0.4996	1.11%	0.5191	5.06%	0.5102	0.5241	2.72%	0.5357	5.00%
NDCG@20	0.5045	0.5036	-0.18%	0.5210	3.27%	0.5208	0.5214	0.12%	0.5320	2.15%

3. For 50 queries training set, the results of RankSVM and ListNet are quite comparable, since both of them rely on only the textual features for learning the ranking model. Similarly, RankSVM-BR and ListNet-BR are also comparable, and both achieve a performance improvement over the corresponding text baseline at the truncation level less than 15. This attributes to the contribution of visual consistency to refine the text-based search results. However, RankSVM-VR and ListNet-VR don't show its superiorities in our experiments. It is because that the ranking distance used in VR which just cares for the scores cannot capture the important information in the text baseline [11]. CAR, which jointly leverages both the textual and visual information, performs consistently and significantly better than all the other methods. We can conclude that the visual consistency really contributes positively to the learned ranking model by dealing with the noisy samples in the textual domain.

For 100 queries training set, there will be 1,110,351 pairwise samples for RankSVM training so that the computational cost is too high to be afforded. Hence only the results for ListNet, ListNet-VR and ListNet-BR are compared with CAR, as shown in Fig. 3. CAR still performs significantly better than the other methods including textual ranking and visual reranking. We conclude that the proposed CAR model and the corresponding training and prediction algorithms are really powerful to rank the images by incorporating the visual content into the ranking model.

Table 1 presents the quantitative details about the performance of CAR as well as the other methods. Here, we use ListNet to represent the textual ranking and ListNet-BR as visual reranking, to compare with CAR. The results of the other textual ranking and visual reranking methods are similar. We can see that CAR performs consistently better than both ListNet and ListNet-BR. For NDCG@1, NDCG@5 the improvements are more than 8%, even up to 17% for NDCG@1, which is significant and promising. Moreover, in terms of NDCG@20, visual reranking improves little (0.12%) in 100 queries training set or even degrades (-0.18%) in 50 queries training set, compared with the text baseline. However, CAR, which learns a unified model to leverage textual and visual information, still benefits from the utilization of visual information, and outperforms visual reranking as well as the text baseline.

In order to see how many queries can benefit from the proposed CAR, we do the following statistics. For each algorithm, we count the number of queries for which the algorithm outperforms all the other methods. It is based on the experimental results on the 50 queries dataset. The statistical results are shown in Fig. 4. It can be observed that CAR outperforms all the other methods significantly. It is

concluded that CAR not only achieves the best overall performance but also performs the best for most of the queries, which shows that CAR is not only accurate but also robust compared with the state-of-the-art methods.

Some example queries and the corresponding ranking results are shown in Fig. 6. From the example images, it is obvious that CAR outputs better ranking list than the other two. Furthermore, we can see that an imperfect text baseline limits the performance improvement in reranking. For example, the queries "Red Wine" and "www google com" cannot achieve a noticable performance boosting by reranking since there are a lot of irrelevant images on the top of the text-based results. For the query "Tweety Bird", reranking even degrades the text baseline. It is because that the top 1 image in the text baseline is irrelevant while the other irrelevant images which are similar to this one are promoted after reranking, such as the second image in the reranking result. However, in CAR, the visual content has been incorporated during the ranking model learning, it can deal with the noisy images by visual information and lead to a better ranking model be learned. As illustrated in Fig. 3, 4, and 6, it performs much better than not only textual ranking but also visual reranking methods.

6.4 Parameter Analysis

The most critical parameter in CAR is γ, which balances the contributions from textual features and visual content. A larger γ tends to prefer the ranking prediction with higher visual consistency, which means that the visual information will take more effects during the learning of CAR model, and vice versa. We perform CAR under the 50 training queries setting, with different γ fixed, and validate all the other parameters over the validation set. The other parameters are validated as in Section 6.2. The results over the test set are shown in Figure 6. It can be observed that, for NDCG, $\gamma = 1$ and $\gamma = 0.5$ achieve the best results among all the variations, while $\gamma = 5$ and $\gamma = 0$ lead to the worst performances. This states that a larger γ (5) which relies heavily on the visual consistency and $\gamma = 0$ which uses only the textual information cannot achieve a good performance. The performance improvements of $\gamma = 0.5$ to $\gamma = 5$ and $\gamma = 0$ are on average 7% for NDCG at different truncation levels. The results demonstrate our conjecture that the joint utilization of both the textual and visual features will lead that a better ranking model can be learned. Furthermore, the balance between the two kinds of information is important for a good ranking performance.

7. CONCLUSIONS

Existing image search engines, which are mostly based on textual ranking model with the surrounding text, title, etc., often suffer from imperfect results incurred by the noisy textual description in image search. Though various methods are developed to handle this problem, such as visual reranking, the performance improvements are limited. In this paper, we propose a Content-Aware Ranking model based on "learning to rank" framework, in which textual and visual information are simultaneously leveraged in the ranking learning process. This unified framework will result in a more robust and accurate ranking model to be learned as noises in textual features can be suppressed by visual content information. The learning algorithms are developed based on large margin structured output learning, by modeling the visual information with a regularization term. Extensive experiments conducted on a large-scale dataset collected from a commercial Web image search engine show that the proposed Content-Aware Ranking outperforms the state-of-the-art methods and so is a promising approach to improve the image search relevance.

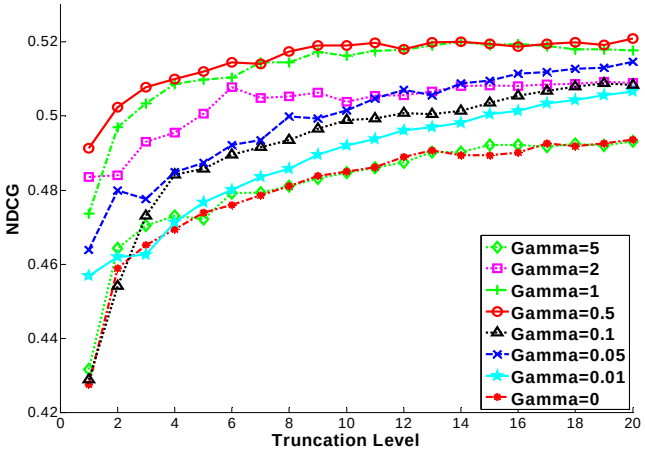

Figure 6: The performances of CAR under different values for parameter γ.

8. REFERENCES

[1] http://www.flickr.com/.

[2] http://www.google.com/.

[3] http://www.live.com/.

[4] http://www.yahoo.com/.

[5] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, May 1999.

[6] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[7] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, 7:2399–2434, 2006.

[8] C. J. C. Burges, R. Ragno, and Q. V. Le. Learning to rank with nonsmooth cost functions. In *NIPS*, pages 193–200, 2006.

[9] C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. N. Hullender. Learning to rank using gradient descent. In *ICML*, pages 89–96, 2005.

[10] Z. Cao and T. yan Liu. Learning to rank: From pairwise approach to listwise approach. In *In Proceedings of the 24th International Conference on Machine Learning*, pages 129–136, 2007.

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

[12] J. Fan, H. Luo, Y. Gao, and R. Jain. Incorporating concept ontology for hierachical video classification, annotation, and visualization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 9(5):939–957, 2007.

[13] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An

efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, 2003.

[14] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995.

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

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

[17] X.-S. Hua and G.-J. Qi. Online multi-label active annotation: towards large-scale content-based video search. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pages 141–150, New York, NY, USA, 2008. ACM.

[18] K. Järvelin and J. Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *SIGIR*, pages 41–48, 2000.

[19] Y. Jing and S. Baluja. Visualrank: Applying pagerank to large-scale image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1877–1890, 2008.

[20] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, pages 133–142, 2002.

[21] D. Knuth. *The Art of Computer Programming Volume 3: Sorting and Searching, Third Edition.*

[22] M. R. Naphade. Statistical techniques in video data management. In *IEEE Workshop on Multimedia Signal Processing*, pages 210–215, 2002.

[23] M. R. Naphade and et al. A light scale concept ontology for multimedia understanding for trecvid 2005. Technical report, In IBM Research Technical Report, 2005.

[24] A. Natsev, A. Haubold, J. Tesic, L. Xie, and R. Yan. Semantic concept-based query expansion and re-ranking for multimedia retrieval. In *ACM Multimedia*, pages 991–1000, 2007.

[25] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. pages 185–208, 1999.

[26] S. E. Robertson and D. A. Hull. The trec-9 filtering track final report. In *TREC*, 2000.

[27] C. Snoek, M. Worring, J. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *ACM Multimedia*, pages 421–430, 2006.

[28] V. Strassen. Gaussian elimination is not optimal. *Numer. Math*, 13:354–356, 1969.

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

[30] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

[31] D. Wang, X. Li, J. Li, and B. Zhang. The importance of query-concept-mapping for automatic video retrieval. In *ACM Multimedia*, pages 285–288, 2007.

[32] J. Xu, T.-Y. Liu, M. Lu, H. Li, and W.-Y. Ma. Directly optimizing evaluation measures in learning to rank. In *SIGIR*, pages 107–114, 2008.

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

[34] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *SIGIR*, pages 271–278, 2007.