VIDEO SEARCH RERANKING VIA ONLINE ORDINAL RERANKING

Yi-Hsuan Yang and Winston H. Hsu

National Taiwan University

ABSTRACT

To exploit co-occurrence patterns among features and target semantics while keeping the simplicity of the keyword-based visual search, a novel reranking methods is proposed. The approach, *ordinal reranking*, reranks an initial search list by utilizing the co-occurrence patterns via the ranking functions such as ListNet. Ranking functions are by nature more effective than classification-based reranking methods in mining ordinal relationships. In addition, ordinal reranking is ease of the ad-hoc thresholding for noisy binary labels and requires no extra off-line learning or training data. When evaluated in TRECVID search benchmark, ordinal reranking, while being extremely efficient, outperforms existing methods and offers 35.6% relative improvement over the text-based search baseline in nearly real time.

Index Terms**—**ranking, rerank, video search, concept

1. INTRODUCTION

Image and video retrieval has been an active research area thanks to the continuing growth of videos, photo collections, media sharing in the social network, etc. The phenomenal success in WWW search has also helped attract increasing interest in investigating new solutions in visual search.

Current image or video search approaches are mostly restricted to text-based solutions which process keyword queries against text tokens associated with the media, such as speech transcripts, captions, file names, etc. However, such textual information may not necessarily come with the image or video sets. The use of other modalities such as image content, audio, face detection, and high-level concept detection has been shown to improve upon text-based video search systems [7], [16]–[18], but such multi-modal approaches require multiple query example images, which could be difficult for users to prepare. Additionally, based on the observations in the current retrieval systems [8], most users expect searching images and videos simply through a few keywords. Therefore, incorporation of multimodal search methods should be as transparent and non-intrusive as possible, in order to keep the simple search mechanism preferred by typical users today [4].

In light of the above observations, reranking frameworks were recently proposed [4]–[6] to leverage low-level visual features or high-level concept detectors in refining the search quality of an initial result produced by a text-based search system. A typical approach [6], referred to as *classification-based* approach in this paper, takes the top-ranked and lower-ranked results of a baseline system as pseudo-positive and pseudo-negative examples to train a support vector machines (SVM) – a discriminative and binary classification model – and regards the normalized classification score over an object as its reranked score. The objects could be photos, web pages, or videos. While being largely unsupervised, such reranking methods have demonstrated significant performance gain over the initial text-based search result [6]. In addition, under the reranking framework, the simplicity of the keyword-based search paradigm is maintained.

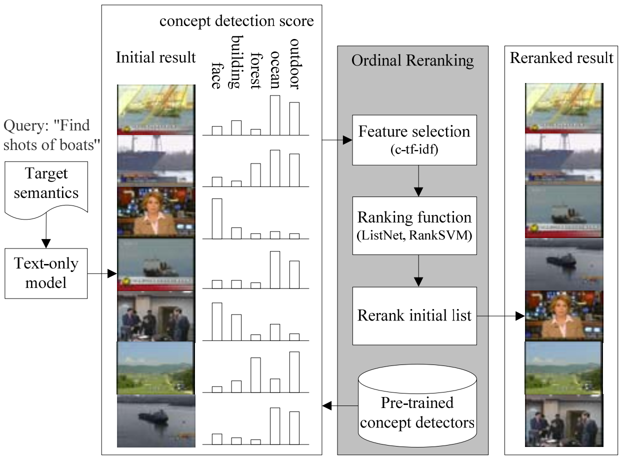

Fig. 1. Architecture of the proposed ordinal reranking framework. The initial result of a text-only method is taken as an approximation of the target semantics; a ranking function is then employed to mine the co-occurrence patterns between the target semantics and extracted features (i.e., concept detection scores [3]), and to rerank the initial result.

Though the classification-based reranking method has the advantage that existing classification methodologies can be directly applied, there arise problems in this direction. First, the objective of learning is formalized as minimizing errors in classifying top ranked and lower ranked results, neglecting the underlying ordinal information of the initial ranked list. Second, it suffers from an ad-hoc mechanism to determine the threshold for noisy binary labels. The size of the pseudo-positive and pseudo-negative sets also needs to be empirically chosen.

In this paper, we propose a novel reranking method, *ordinal reranking,* in which ranking algorithms such as RankSVM [10] and ListNet [19] are employed to learn the co-occurrence patterns between target semantics and extracted features from the initial result, and then further rerank it. Since the objective of ranking functions is to minimize errors in ranking objects, ordinal reranking is more effective and efficient in mining ordering information, while being ease of the ad-hoc thresholding problem.

The architecture of the proposed ordinal reranking framework is illustrated in Fig. 1. After reviewing existing reranking work in Section 2, the proposed ordinal reranking is detailed in Section 3. As presented in Section 4, ordinal reranking outperforms existing methods when evaluated on TRECVID 2005 (TV05) video search benchmark. Finally Section 5 concludes the paper.

2. RELATED WORK

As shown in Table I, the works on reranking can be categorized to off-line methods and on-line methods. Off-line methods use annotations of training data to discover contextual information to rerank the test data, while on-line methods approximate the initial result of a baseline system as pseudo ground truth to learn to rerank the initial result. Below we review some existing methods briefly.

The notion of *context fusion*, or the use of peripherally related features to refine detection of semantic topics, has been explored in prior work for use in concept detection [14]–[17]. In early *off-line* methods, the learning is fully supervised and requires explicit knowledge of the target semantics and ground truth labels in order to discover the contextual relationships with other features. While this constraint is fine for concept detection, where many labels are available, it is unclear how these approaches could cover the unsupervised conditions in search.

The *classification-based reranking* is rooted in pseudo-relevance feedback for text search [7], [18] where the initial result of a baseline system is utilized to discover the co-occurrence patterns between target semantics and extracted features. Features that can potentially discriminate between top-ranked and lower-ranked images are leveraged to determine a new ranking without resorting to any off-line learning or extra training data. In [4] and [5], the reranking framework is applied on low-level features such as text token frequencies, grid color moments, and image textures. In [6], the use of discriminative classifiers on a large 374 concept lexicon [2] is explored. The pseudo-positive and pseudo-negative examples are used to train a SVM model, and the classification margin for an object is regarded as its (new) reranked score. As [6] shows, classification-based reranking achieves comparable performance to supervised methods such as [14] for the concept detection task.

3. ORDINAL RERANKING FRAMEWORK

Table I
Comparison of reranking algorithms

	off-line (supervised)	on-line (unsupervised)	
		classification-based reranking	ordinal reranking
learning strategy	using annotations of training data to discover contextual info. to rerank test data	using initial result of a baseline system as pseudo ground truth to learn to rerank the initial result	
		top-ranked (pos.) low-ranked (neg.)	maintaining ranking order
advant-age	-	unsupervised, comparable performance to off-line methods	more effective and efficient in mining ordinal information
drawback	not applicable to image/video search	ad-hoc thresholding, losing ordinal info.	-
reference	[14]–[17]	[5]–[7], [18]	This work

To get rid of the drawback of the classification-based method, we propose to incorporate ranking algorithms to the reranking framework. Below we first give a brief review of existing ranking algorithms and then describe the proposed ordinal reranking framework.

3.1. Ranking algorithms

Any system that presents results to a user, ordered by a utility function that the user cares about, is performing a ranking. A common example is the ranking of search results from the search engine (i.e., Google). A ranking function assigns a score to each object, and ranks the object by that. The ranking order represents the relevance of objects with respect to the query. Given a set of queries and the corresponding manually-annotated ranked lists of objects, the task of training a ranking model which can precisely predict the ranking lists in the data is commonly referred to as “learning to rank” and has received great interests from academia and industry [10]–[12].

Many of the existing ranking methods take object pairs as instances, formulate the learning task as classification of object pairs into two categories (correctly and incorrectly ranked), and train classification models for ranking. The use of SVM as the classification model leads to the method called RankSVM [10]. Though the pairwise approach offers advantages, it ignores the fact that ranking is a prediction task on the list of objects. Moreover, the pairwise approach can be time-consuming as it takes every possible pair in the data and runs at a complexity of $O(N^2)$, where N is the number of objects to be ranked.

The listwise approach, *ListNet*, recently proposed in [19] conquers these shortcomings by using score lists directly as learning instances and minimizing the listwise loss between the initial list and the reranked list. In this way, the optimization is conducted directly on the list, and the computational cost can be reduced to $O(N)$, making it possible for other promising “online” reranking applications. More specifically, ListNet transforms both the initial scores and the reranked scores into probability distributions by sum-to-one normalization, and uses cross-entropy to measure the distance between these two probability distributions as the listwise loss function. To minimize the listwise loss, linear neural network model and gradient descent technique are employed. Due to space limitation, please refer to [10] and [19] for more details.

3.2. Ordinal reranking

While learning to rank requires a great amount of supervision, reranking takes an unsupervised fashion and requires no ground truth of the ranking result. To utilize the ranking algorithms for reranking, some modifications for the reranking framework are needed. Below we first formulate the standard reranking framework, and then describe how we apply the cross-validation technique as [6] to adapt ranking algorithms to the reranking framework.

Given a query, a baseline system is adopted to produce an initial ranked list, which is composed of a list of objects $D = (d_1, d_2, \ldots, d_N)$, where d_j denotes the j-th object, and a list of associated relevance scores $Y = (y_1, y_2, \ldots, y_N)$, where $y_j \in [0, 1]$ denotes the relevance score of d_j with respect to the query. Furthermore, for each object d_j a feature vector $\mathbf{X}_j = (X_{j1}, X_{j2}, \ldots, X_{jM})$ is extracted, where M is the dimension of the feature space. The purpose of reranking is to mine the co-occurrence patterns among $\mathbf{X}$ and Y, and then exploit the patterns to assign a reranked score z_j to object d_j. As reviewed in Section 2, a number of algorithms have been proposed to mine the co-occurrence patterns [4]–[6].

The *pseudo ground truth* data needed to train ranking algorithms can be obtained by randomly sampling a number of objects in the initial list. The ranking algorithms are then used to predict the reranked scores of the remaining objects. In practice, we employ cross validation as [6] to randomly divide the dataset into a certain number of folds. One fold is used as the test data and the remaining folds for training in iteration. This process is repeated until each fold is held out for testing once.

After normalization, the initial score y_j and reranked score z_j are fused to produce a merged score s_j by taking the weighted averaged as follows: $s_j = \alpha y_j + (1\text{-}\alpha) z_j$. $\alpha \in [0, 1]$ denotes a fusion weight on the initial score and reranked score; α=1 means totally reranked. Such a linear fusion model, though simple, has been shown adequate to fuse visual and text modalities in video retrieval and concept detection [16], [17]. The fused scores are then sorted to produce a reranked search result.

Fig.1 gives an illustrative example. From the initial text-based search result, ordinal reranking mines the co-occurrence patterns and identifies “ocean” and “outdoor” as relevant concepts to the target semantics. Reranking is then made by reordering the shots with high search scores linearly fused by these relevant concepts.

3.3. Feature selection for reranking

We further investigate feature selection methods to remove irrelevant or redundant features and thus enhance accuracy. Most feature selection algorithms are dedicatedly proposed for classification and thus not applicable here. In addition, the feature selection algorithm needs to be unsupervised due to the nature of reranking. Therefore, we adopt the concept tf-idf (c-tf-idf) proposed in [9] to select informative features. Viewing video shots as documents and features as terms, [9] constructs a shot-feature co-occurrence table and then defines the c-tf-idf in a similar way as tf-idf, the best known term-informativeness measurement which offers a good combination between popularity (idf) and specificity (tf) [13]. Note that c-tf-idf is quite generic and can be applied on both concept scores and visual features. We use c-tf-idf to select informative features for each query independently.

4. EXPERIMENTAL RESULTS

4.1. Experiment setups

We conduct experiments on TV05 search benchmark [1], where 24 query topics are provided with ground truth labels. The video data are segmented into shots and each shot is represented by a few keyframes by NIST [1]. Since TRECVID evaluates the search results over the top 1000 shots, we use the top 1300 subshots (which typically encompass the top 1000 shots) returned by the text-based search method “text-okapi” [5] for reranking. Empirically we set the fusion weight α to 0.5 for simplicity, and use five-fold cross validation to conduct ordinal reranking. The program is implemented in MATLAB on a regular Intel Pentium server. Performances are evaluated in shot-level average precision (AP) or mean average precision (MAP) (See more explanations in [1]).

For feature representation, we adopt the detection scores of pre-trained concept detectors [3] for the LSCOM (cp374) and LSCOM-Lite (cp39) lexicons [2] to provide high-level semantics. The LSCOM-Lite lexicon is with 39 concepts and is an early version of LSCOM. A detection score is generally normalized into [0, 1], showing the confidence of the existence of a concept. Low-level visual features including 5x5 grid color moments (grid) and 4x6 Gabor textures (gabor) [5] are also extracted to compare against the high-level concept scores.

4.2. Performance evaluation of ordinal reranking

Table II shows the performance comparison of variant reranking methods on TV05 video search task. While the MAP of the baseline “text-okapi” is 0.087, existing methods [5], [6] improve the MAP to 0.105 and 0.112 respectively. Note [5] uses low-level visual features without feature selection, whereas [6] utilizes mutual information to select the 75 most informative concepts from cp374.

We first compare RankSVM and ListNet using cp39 as the feature set. As we have predicted, RankSVM is extremely time-consuming and thus get abandoned in the following experiments. On the contrary, thanks to the linear kernel, ListNet is surprisingly efficient, and takes less than

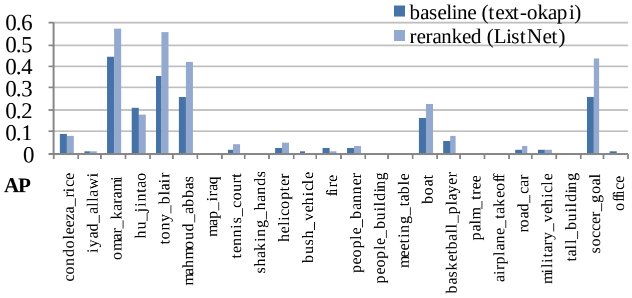

Fig. 2. Average precision of baseline and reranked search results for each query in TRECVID 2005. The reranking algorithm is ListNet, with 75 most informative concepts selected from LSCOM 374 concepts by c-tf-idf. Ordinal reranking improves the result of almost every query and improves the MAP from 0.087 to 0.118 (35.6% relative improvement).

one second to rerank a single query. This real-time efficiency has made ListNet superior to [5], which needs a clustering process and [6], where nonlinear optimization is conducted. Moreover, the fact that ListNet improves the MAP to 0.113 with a small concept lexicon cp39 further demonstrates its effectiveness in reranking.

We evaluate the performance of ListNet with variant feature sets. Apparently, reranking based on low-level visual features does not outperform that based on high-level concepts. This is not surprising since concepts tend to capture both the visual similarities and the semantic correlations. In addition, the visual patterns of target semantics may not be consistent, bringing noises to the reranking procedure. For example, a name-entity may wear clothes of different colors and appear in diverse locations.

It can also be observed, with the semantic-richer cp374 and feature selection method c-tf-idf (75 most informative concepts selected), ListNet improves the MAP up to 0.118, which significantly outperforms existing methods. Moreover, the performance improvements are consistent – almost all queries are improved, as shown in Fig. 2.

5. CONCLUSION

In this paper, to improve the text-based visual search, a novel reranking algorithm, ordinal reranking, is proposed to mine the co-occurrence patterns between the target semantics and extracted features. The adoption of ranking algorithms makes ordinal reranking more effective and efficient than classification-based reranking methods in mining ordinal information. Moreover, because ordinal ranking optimizes the ordering of an initial list directly, it is ease of the ad-hoc thresholding for noisy binary labels and requires no extra off-line learning processes or training data. Experimental results show that ordinal reranking is much more efficient and effective than existing reranking methods and improves 36% against the text-based initial search.

Table II
Performance comparison of variant reranking methods on the TRECVID 2005 search task using "text-okapi" as the baseline

Reranking algorithm	Feature set	Feature selection	MAP	improv. (%)	time/ query
baseline	text-only	-	0.087	-	-
IB [5]	grid+gabor	-	0.105	20.7%	18s
SVM [6]	cp374	mutual info	0.112	28.7%	17s
RankSVM	cp39	-	0.103	18.4%	1hr
ListNet	cp39	-	**0.113**	30.0%	**0.2s**
ListNet	grid+gabor	-	0.105	20.7%	0.9s
ListNet	cp374	-	0.116	33.3%	1.4s
ListNet	cp374	c-tf-idf	**0.118**	35.6%	**0.4s**

6. REFERENCES

[1] NIST TREC Video Retrieval Evaluation. http://www-nlpir.nist.gov/projects/trecvid/.

[2] M. Naphade et al, "Large-scale concept ontology for multimedia," *IEEE Multimedia Magazine*, pp. 86–91, 2006.

[3] Akira Yanagawa et al, "Columbia University's baseline detectors for 374 LSCOM semantic visual concepts," Columbia Univ. ADVENT Tech. Report #222-2006-8, 2007.

[4] W. Hsu et al "Video search reranking through random walk over document-level context graph," *ACM Multimedia*, 2007.

[5] W. Hsu et al, "Video search reranking via information bottleneck principle," *ACM Multimedia*, 2006.

[6] L. Kennedy and S.-F. Chang, "A reranking approach for context-based concept fusion in video indexing and retrieval," *ACM CIVR*, pp. 333–340, 2007.

[7] R. Yan, A. Hauptmann, and R. Jin, "Multimedia search with pseudo-relevance feedback," *ACM CIVR*, 2003.

[8] J. Battelle, *The Search: How Google and Its Rivals Rewrote the Rules of Business and Transformed Our Culture*, Portfolio Trade, 2006.

[9] X. Li et al, "Video search in concept subspace: a text like paradigm," *ACM CIVR*, pp. 603–610, 2007.

[10] R. Herbrich, T. Graepel, and K. Obermayer, "Support vector learning for ordinal regression," *ICANN*, pp. 97–102, 1999.

[11] T. Joachims, "Optimizing search engines using clickthrough data," *ACM SIGKDD*, pp. 133–142, 2002.

[12] Y. B. Cao et al, "Adapting ranking SVM to document retrieval," *ACM SIGIR*, pp. 186–193, 2006.

[13] A. Aizawa, "An information-theoretic perspective of tf-idf measures," *Information Processing and Management*, vol. 39, pp. 45–65, 2003.

[14] W. Jiang et al, "Context-based concept fusion with boosted conditional random fields," *IEEE ICASSP*, 2007.

[15] C. G. Snoek et al, The MediaMill TRECVID 2006 Semantic Video Search Engine, *NIST TRECVID workshop*, 2006.

[16] S.-F. Chang et al, "Columbia University TRECVID-2006 video search and high-level feature extraction", *NIST TRECVID workshop*, 2006.

[17] M. Campbell et al, "IBM Research TRECVID-2006 Video Retrieval System," *NIST TRECVID workshop*, 2006.

[18] T.-S. Chua et al, "TRECVID 2004 search and feature extraction task by NUS PRIS," *NIST TRECVID workshop*, 2004.

[19] Z. Cao et al, "Learning to rank: from pairwise approach to listwise approach," *IEEE ICML*, pp. 129–136, 2007.