Improving Web Image Search using Meta Re-rankers

B.Kavitha[1], N. Sujata [2]

[1]Department of Computer Science and Engineering, Chitanya Bharathi Institute of Technology,Hyderabad, INDIA

[2]Department of Computer Science and Engineering,Sridevi Women's Engineering College,Hyderabad, INDIA

Abstract: The previous methods for image search reranking suffer from the unreliability of the assumptions under which the initial text based image search result is employed in the reranking process. In our proposed system, prototype-based reranking method is suggested address this problem in scalable fashion. This typical assumption that the top-images in the text-based search result are equally relevant is relaxed by linking the relevance of the images to their initial rank positions. The number of images is employed by the initial search result as the prototypes that serve to visually represent the query and that are subsequently used to construct Meta re-rankers. For applying different Meta re-rankers to an image from the initial result, then the re-ranking scores are generated, which are then aggregated by using a linear model to produce the final relevance score and the new rank position for an image in the re-ranked search result. It is improving the performance over the text-based image search engine.

Key concepts: Prototype based Meta re-ranker, Text based search Image Re-ranking.

1. Introduction

The existing web image search engines, including Google, Bing and Yahoo retrieve and rank images mostly based on the textual information associated with the image in the hosting web pages, such as the title and the surrounding text. While text-based image ranking is often effective to search for relevant images, the precision of the search result is largely limited by the mismatch between the true relevance of an image and its relevance inferred from the associated textual descriptions.

To improve the precision of the text-based image search ranking and visual reranking has been proposed to refine the search result from the text-based image search engine by incorporating the information conveyed by the visual modality. Based on the images in the initial result, visual prototypes are generated that are visually representing the query. Each of the prototypes is used to construct a Meta reranker to produce a reranking score for any other image from the initial list. Finally, the scores from all the Meta rerankers are aggregated together using a linear reranking model to produce the final relevance score for an image and to define its position in the reranked results list. The linear reranking model is learned in a supervised fashion to assign appropriate weights to different meta rerankers. Since the learned model weights are related to the initial text-based rank position of the corresponding image and not to the image itself, then the reranking model is query-independent and can be generalized across queries.

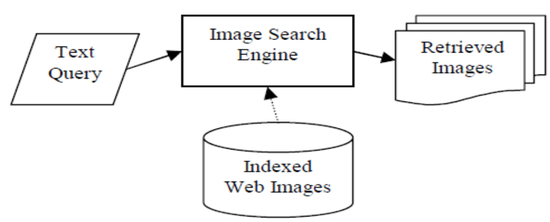


Figure 1. A high-level overview of web image Search engine

2. Proposed work

To improve the performance of searching images visual search reranking is very good option. 4 steps are needed in our module text ranking, Prototype generation, and Meta Re-Ranker and also Re-Ranking Result.

• Text Ranking:

Initial search is text based search. We need image search engine to submit query from user. In the search engine query is in text format. It is the text based Image search in which we get the image ranking on the bases of text query which we give.

• Prototype generation:

In prototype generation phase we create a rules for image reranking on which further images has been reranked. In this we examine the visual similarities. From the top L image set prototypes are generated using visual similarities. These prototypes are used as an input to the meta ranker.

• Meta Ranking:

In meta reranking, multiple set prototype technique is used. This technique is used to compute the ranking score. The computed reranking score give as an input to reranking model to estimate ultimate reranking score. In re-ranking module use of K-means clustering is beneficial. The K-means handles the ranking problem. Thus the basic idea is for decomposing a ranking in to a set off pair-wise preferences and then to reduce the ranking-learning problem into a pair wise classification problem.

• Reranking Result:

In this step we get final reranked images in prototype based ranking. In this paper we proposed a prototype-based reranking framework, which constructs meta rerankers corresponding to visual prototypes

representing the textual query and learns the weights of a linear reranking model to combine the results of individual meta rerankers and produces the reranking score of a given image taken from the initial text-based search result.

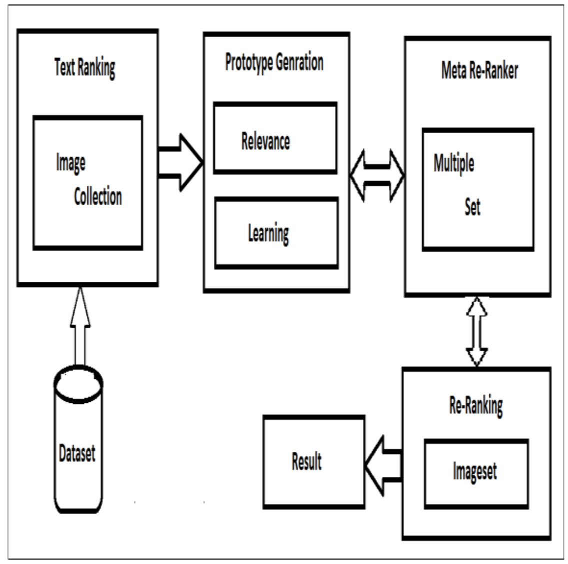


Figure 2: Architecture of Image re-ranking.

3. Image Reranking Framework

The proposed prototype-based reranking method consists of two steps.

Online:

In the online part, when a textual query is submitted to the image search engine by a user, initial search is performed using any contemporary text-based search technique. Then, the visual prototypes are generated and for each prototype a meta reranker is constructed. So, for each of the top N images in the initial search result, an L-dimensional score vector is obtained comprising the scores from all meta rerankers when applied to that image. Finally, the score vector is used as input to a reranking model, which is already turn to offline to estimate the ranking scores in the reranked image search list.

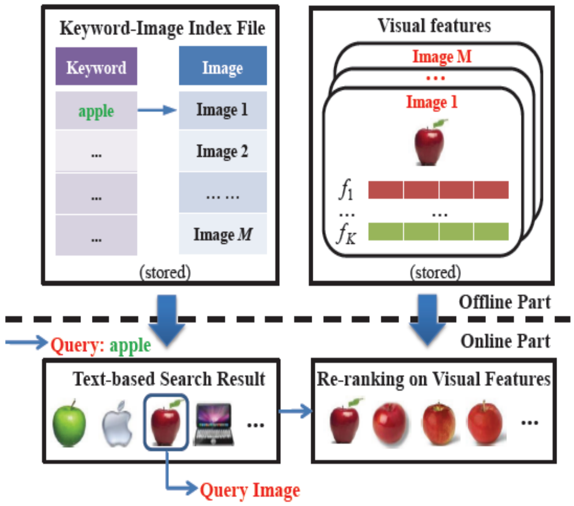


Figure 3 Image Reranking Framework

Offline:

The offline component is devoted to learning the reranking model from user-labeled training data. Since the learned model will be used for reranking the text-based search results, the training set is constructed from these results through the following steps. First, several representative queries sampled from the query log are selected. Then, using these queries the top images are retrieved from the text-based image search engine and downloaded for processing. Finally for each query-image pair, people are invited to label the relevance between them to form the ground-truth. After the training data is collected, score vector can be computed from the meta rerankers, as mentioned in the online part, for each image and the corresponding query. Then the reranking model is learned and stored in the memory to be used in the online part for responding to user's submitted queries.

Learning the reranking model:

The linear reranking model has learned by estimating the weights of the combined scores coming from different meta rerankers. This problem can be addressed using a learning-to-rank method, by regarding this score vector as the ranking feature of an image. Ranking K-means is among the most popular learning to rank algorithms. This algorithm is widely used K-means clustering is to handle a ranking problem. The basic idea has to decompose a ranking into a set of pair wise preferences and then to reduce the ranking-learning problem into a pair-wise classification problem. The basic idea was to decompose a ranking into a set of pair-wise preferences and then to reduce the ranking-learning problem into a pair-wise classification problem.

Standard efficient approaches are to learning K-means clustering, such that a sequential minimal optimization, it can be directly employed for learning the Ranking K-means. Moreover, the fast algorithm, e.g., the cutting-plane algorithm, can be adopted to speed up the training of a linear Ranking K-means.

The reason is why the learned reranking model described above can be generalized across queries beyond those used for the training was that the model weights are not related to specific images but to their rank positions in the text-based search result. The separation of the model weights from specific images is the key to ensure that there ranking model only needs to be learned once and then it can be applied to any arbitrary query.

The existing learning is to-rerank methods, including the supervised-reranking and query-relative classifier, design the reranking model based on the hand designed ranking features defined at a higher abstraction level or on the ordered visual words, respectively. When compared to them, the prototype-based learning to rerank method learns how likely the images at each of the ranked position in by text-based result are to be relevant to the query.

4. Constructing Meta Rerankers

One of the key steps in the Multiple Set Prototype image reranking method is the construction of meta rerankers. The computed scores are used to input for the reranking model to estimate the ultimate ranking scores to determine the rank position of the images in the reranked result. There are three types to construct meta rerankers, depending on how the prototypes are generated from the initial text-based search result. Single image prototype, Multiple average prototype and multiple set prototype are the three algorithms for constructing meta rerankers.

Single-Image Prototype:

A straightforward way to generate a set of

A prototype is to select top images from the text based result, as illustrated in fig: 4.

If we denote this set as $\{P^S{}_i\}_{i=1}^{L}$ then the meta reranker can be built simply based on the visual similarity S(.) between the prototype P_i^S and the image Ij to be reranked as

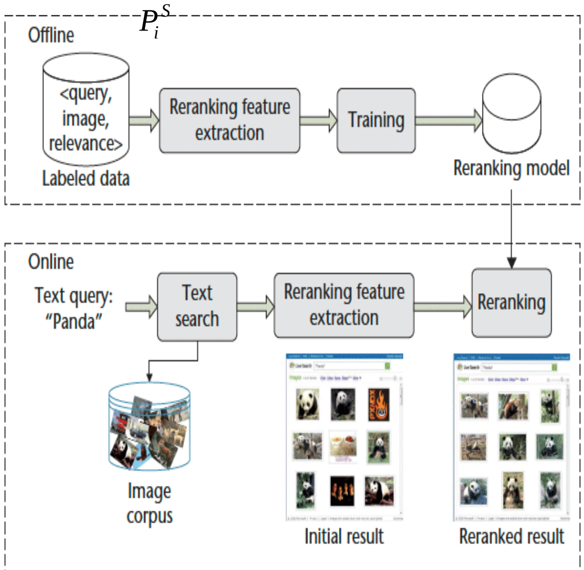


Figure 4 Single-Image prototype

$$M^S\left(I_j / P_i^s\right) = S\left(I_j, P_i^s\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

The score vector aggregating the values (1) from all meta rerankers is then used as input to the linear reranking model in order to compute the definitive ranking score for image I_j :

$$R^S(I_j) = \sum_{i=1}^{L} W_i x S(I_j, P_i^s) \ldots\ldots\ldots \ldots\ldots\ldots \ldots\ldots (2)$$

Where Wi are the individual weights from the model weight vector W.

Multiple average prototypes:

Prototype P_i^{MA} can be construct by first selecting the top L images in the initial search result list and then by cumulatively averaging the features of all images ranked starting from the topmost position to the position i , as illustrated in Fig. 4. In other words, the prototype P_i^{MA} can be defined as

$$P_i^{MA} = \frac{1}{i}\sum_{j=1}^{i} I_j \ldots\ldots\ldots \ldots\ldots\ldots \ldots\ldots\ldots \ldots\ldots\ldots \ldots (3)$$

Then, this prototype can be employed to compute the scores of individual meta rerankers by again computing the visual similarity between a prototype and the image to be reranked:

$$M^{MA}\left(I_j / P_i^{MA}\right) = S\left(I_j, P_i^{MA}\right) \ldots\ldots\ldots\ldots\ldots (4)$$

ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)

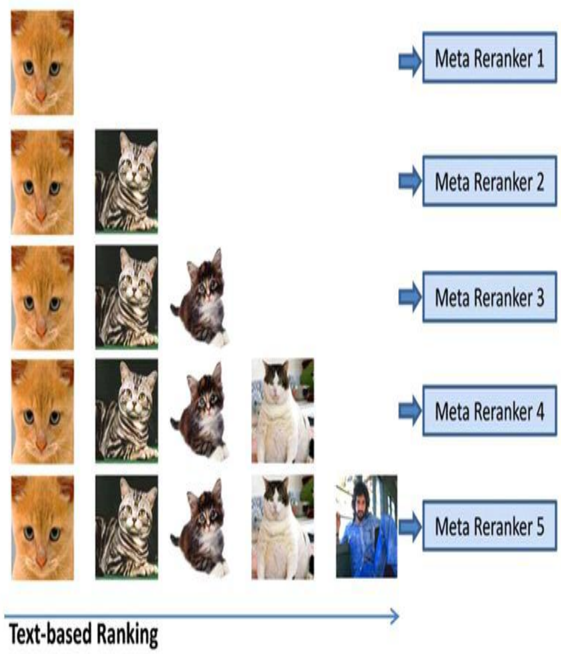


Figure 5 Multiple Average prototype

Multiple set prototype:

The multiple-set prototype P_i^{MS} at rank i is defined as a bag of images ranked from the topmost position to the rank i, as illustrated in Fig.6.

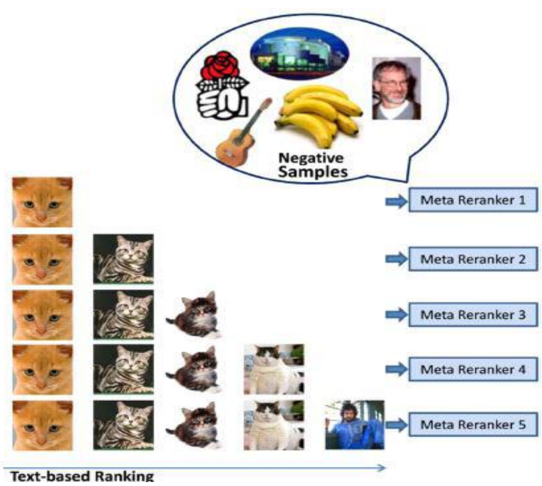


Figure 6 Multiple set prototype

$$P_i^{MS} = \{I_J\}_{i=1}^{i} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

The multiple-average prototype is the average of features for the images in the multiple-set prototype and can be seen as a special case of this prototype. The multiple-set prototype is a more flexible representation, which can support the development of more types of Meta rerankers. Given a multiple-set prototype P_i^{MA} can learn a visual classifier by regarding all the images in P_i^{MA} as positive samples, which is then employed as meta reranker and the prediction score is used as the meta reranking score. Since a discriminative learning method is usually more effective for learning a visual model, there is K-means in this paper. However, it needs not only positive samples but also negative samples. The Meta reranker with a

multiple-set prototype can be defined as follows:

$$M^{MS}(I_j / P_i^{MS}) = p(I_j / \theta) \ldots\ldots\ldots\ldots\ldots\ldots (6)$$

$$\theta = \arg\max_{\theta} \rho(P_i^{MS} / \theta) \ldots\ldots\ldots\ldots\ldots\ldots (7)$$

Where θ is the learned model and

Here is the analysis of the properties of the reranking method based on the multiple-average prototype. By using the dot product as the similarity measure, a corresponding meta reranker, leads to the following expression:

$$R_i^{MA}(I_j) = \sum_{i=1}^{L}\left(w_i x \frac{1}{i} \sum_{k=1}^{i} S(I_k, I_j) \right) \ldots\ldots\ldots (8)$$

$$= \sum_{i=1}^{L} \alpha_i x S(I_{i,} I_j) \ldots\ldots\ \ldots\ldots\ \ldots\ldots\ \ldots\ldots\ ..(9)$$

where

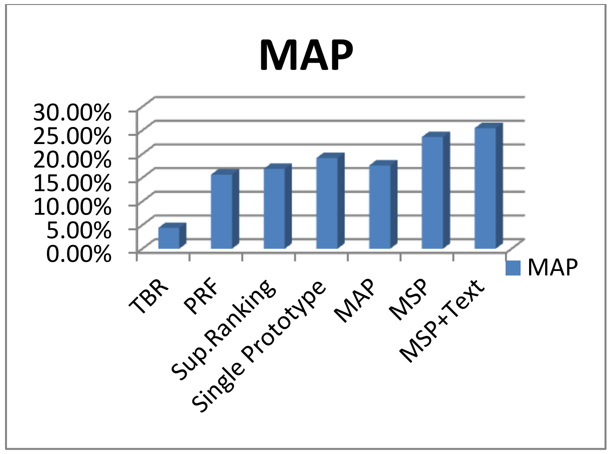
$$\alpha_i = \ = \sum_{k=i}^{L} \frac{w_k}{k} \ldots\ldots\ \ldots\ldots\ \ldots\ldots\ \ldots\ldots\ \ldots\ldots (10)$$

The above expressions transform the model based on a multiple average prototype on to the model based on a single image prototype, however, with different weights. It states that the ranking in the text-based search result represents the ordering of the importance for each individual image to be used as a prototype for reranking. In other words, there ranking

ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)

based on a multiple-average prototype will rely more on the initial text based result than that based on a single image prototype.

Weights for individual images by the reranking based on a multiple-average prototype will decline gradually with the decreasing ranks. This may make this reranking model less aggressive and more robust than the one based on a single image prototype. Meanwhile, it makes the reranking model learned by the multiple-average prototype-based reranking method hardly over-fitting to the training queries.



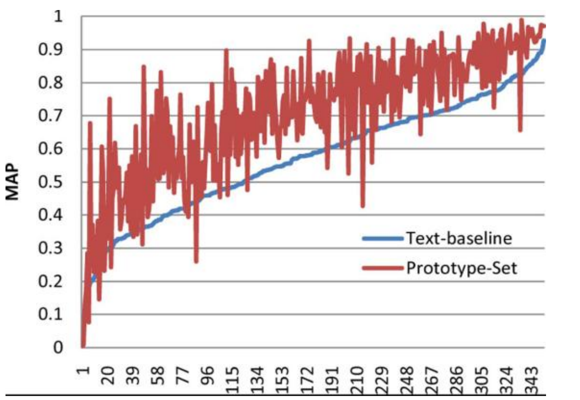
5. Comparison and Results

a) Comparison

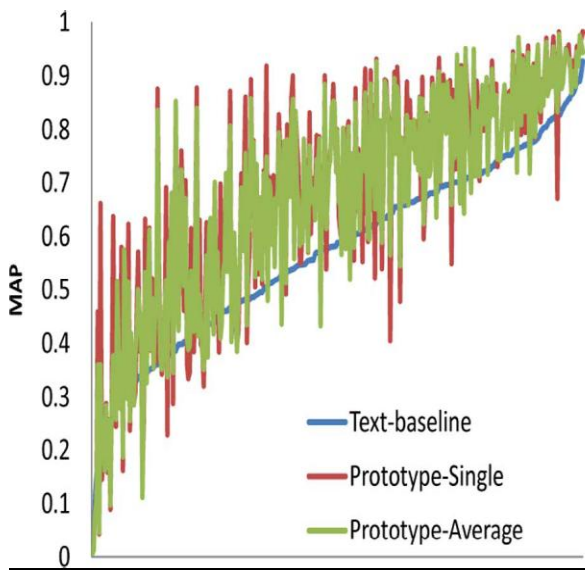
TABLE I Performance comparison of various reranking methods.

Method	Precision (Mean Avg. Precision)
Text based Ranking	4.39%
Pseudo Relevance Feed Back	15.64%
Supervised Reranking	16.87%
Single Prototype	19.16%
Multiple Average Prototype	17.57%
Multiple Set Prototype	23.60%
Multiple Set Prototype +Text	25.45%

ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)



Performance comparison of *Prototype-Set* and *Text-baseline* from the search engine. The query is arranged in the ascending order of the performance of *Text-baseline*.



MAP comparison of *Prototype-Single* and *Prototype-Average*
methods.

b) Results

- **Query Image**

When we are searching for an image in search engines, the corresponding images are loaded. Among them uncategorized images are also spotted. However, by producing such databases containing a large number of images and with high precision is still an arduous manual task.

Generally Image search engines apparently provide an effortless route. But they are presently limited by returned image precision. Our objective in this work is to re rank a large number of images of a particular class automatically, and to achieve this with the high precision.

Image clusters for each topic are formed by selecting images where nearby text is top ranked by the topic. The user then partitions the clusters into positive and negative for the class. Then second, images and the associated text from these clusters are used as exemplars to train a classifier based on voting on visual (shape, color, and texture) and text features.

Fig 5.1 Query Image

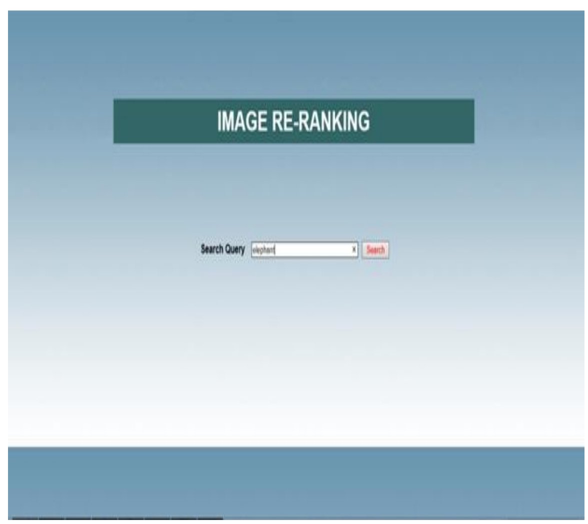


Fig 5.2 Giving Query

- **Download Associate Images, Url Parsing**

We are using Google search engine to downloading images from the Web.

Image Search gives a very low precision (only about 4 percent) and is not used for the harvesting experiments. This low precision is probably due to the fact that Google selects many images from Web gallery pages which contain images of all sorts. Google is able to select the in-class images from those pages, e.g., the ones with the object-class in the filename; however, if we use those Web pages as seeds, the overall precision greatly decreases. Therefore, we only use Web Search and Google Images, which are merged into one data set per object class. Table 2 lists the 18 categories downloaded and the corresponding statistics for in-class and non-class images. The overall precision of the images downloaded for all 18 classes is about 29 percent.

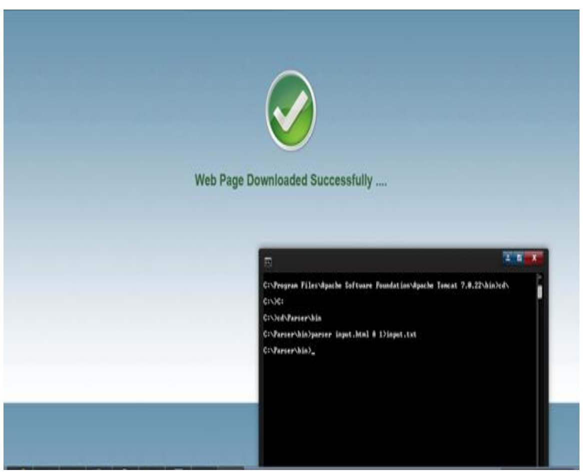


Fig 5.3 Downloading Images

- **Apply Re-Ranking Algorithm**

Now describe the re ranking of the returned images based on text and metadata alone. Here, we follow and extend the method proposed by using a set of textual attributes whose presence is a strong indication of the image content.

The goal is to re rank the retrieved images. Each feature is treated as binary: "True" if it contains the query word (e.g., penguin) and "False" otherwise. To rerank images for one particular class (e.g., penguin), we do not employ the whole images for that class. Instead, we train the classifier using all available annotations except the class we want to rerank.

Fig 5.4 Re-Ranking

Fig 5.5 Applying Re-ranking

- **K-means Implementation**

K-means clustering is used to form the cluster of similar images. This technique will filter PNG or GIF format images. Based on threshold value the re ranking process will be done.

ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)

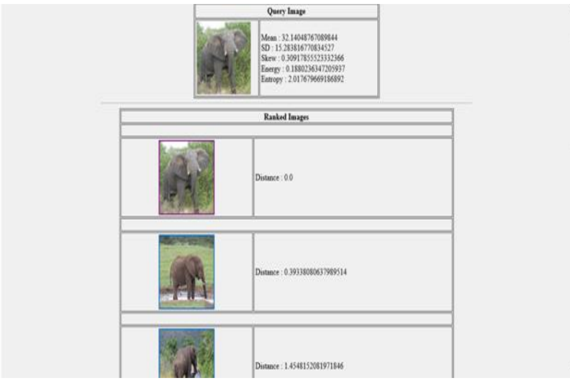


Fig 5.6 K-means Implementation

6. Conclusion

This paper proposes a prototype based image reranking by using K-means clustering, which constructs Meta rerankers corresponding to visual prototypes representing the textual query and learns the weights of a linear reranking model to combine the results of individual Meta rerankers and produce the reranking score of a given image taken from the initial text-based search result. It improves the performance by 25.48% over the text-based search result by combining prototypes and textual ranking features. The natural extension of the approach described in this paper would be to apply the proposed methods to learn concept models from image search engines in a semiautomatic fashion. Compared to the fully automatic methods, then the semi-automatic approach could learn then the concept models for any arbitrary concept much better and with only little human supervision. While the proposed methods have proved effective for reranking image search results, There was envision of two directions for future work to further improve the reranking performance. First, It could be could further speed up the Prototype-Set method variant while decreasing the precision degradation. Since top images are incrementally added into the multiple-set prototypes to train the meta rerankers, one of the possible approaches in this direction is to utilize the online learning algorithms. Next the second, although It assume that the rank position is generally correlated with the relevance value of the image found there, and while our results show that this assumption can be regarded valid in a general case, still the deviations from this expectation can occur for individual queried. One possible approach here would be to automatically estimate the query-relative reliability and accuracy of each meta-reranker and then incorporate it into the reranking model. The another approach may be to learn the reranking models for different query classes.

ACKNOWLEDGMENT

We wish to acknowledge the efforts of **Pantech Solution Pvt ltd., Hyderabad**, for guidance which helped us work hard towards producing this research work.

7. References

[1] Linjun Yang, and Alan Hanjalic, “Prototype based Image search Rrranking”, IEEE Transactions on Mltimedia, vol. 14, no. 3, June 2012.

[2] L. Yang and A. Hanjalic "Supervised reranking for web image search", Proc. ACM Multimedia, 2010

[3] R. Fergus , L. Fei-Fei , P. Perona and A. Zisserman "Learning object categories from Google's image search", Proc. ICCV, 2005.

[4] W. H. Hsu , L. S. Kennedy and S.-F. Chang "Video search reranking via information bottleneck principle", Proc.ACM Multimedia, 2006.

[5] M. Fritz and B. Schiele "Decomposition, discovery and detection of visual categories using topic models", Proc. CVPR, 2008.

[6] W. H. Hsu , L. S. Kennedy and S.-F. Chang "Video search reranking through random walk over

document-level context graph", Proc. ACM Multimedia, 2007.

[7] 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 2008 .

[8] X. Tian , L. Yang , J. Wang , Y. Yang , X. Wu and X.-S. Hua "Bayesian video search reranking", Proc. ACM Multimedia, 2008 .

[9] Y. Liu , T. Mei , X.-S. Hua , J. Tang , X. Wu and S. Li "Learning to video search rerank via pseudo preference feedback", Proc. ICME, 2008..

[10] F. Schroff , A. Criminisi and A. Zisserman "Harvesting image databases from the web", Proc. ICCV, 2007.