*IOSR Journal of Computer Engineering (IOSR-JCE)*
*e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 13, Issue 1 (Jul. - Aug. 2013), PP 44-50*
*www.iosrjournals.org*

# Improving Web Image Search Re-ranking

L. Ramadevi[1], Ch. Jayachandra[2], D.Srivalli[3]
[1]*M.Tech (S.E), VCE, Hyderabad, India,*
[2]*M.Tech (C.S.E), VCE, Hyderabad, India,*
[3]*M.Tech (S.E), VCE, Hyderabad, India,*

***Abstract:*** *Nowadays, web-scale image search engines (e.g. Google Image Search, Microsoft Live Image Search) rely almost purely on surrounding text features. This leads to the ambiguous and noisy results. We propose an adaptive visual similarity to re-rank the text based search results. A query image is first categorized as one of several predefined intention categories, and a specific similarity measure has been used inside each of category to combine the image features for re-ranking based on the query image. The Extensive experiments demonstrate that using this algorithm to filter output of Google Image Search and Microsoft Live Image Search is a practical and effective way to dramatically improve the user's experience.*

***Keywords****: Content Based Image Retrieval, Image Ranking, Image Searching, Semantic Matching, Visual Re-ranking, Image Ranking and Retrieval Techniques.*

## I. Introduction

With the number of digital images in the WWW increasing explosively, efficient image search in large scale datasets has attracted great interest from both academia and industry. However, image retrieval is currently far less efficient than text retrieval because images are unstructured and much more difficult to process than texts. The approaches of retrieving and ranking images from large scale datasets can be largely divided into the following three categories:

### 1.1 Text-Based Approaches:

The search engine returns corresponding images by processing. The associated textual information, such as file name, surrounding text, URL, etc., according to keywords input by users. Most of popular commercial Web image search engines like Google and Yahoo! adopt this method. While text-based search techniques have been verified to perform well in textual documents, they often result in mismatch when applied to the image search. The reason is that metadata cannot represent the semantic content of images. For example, a search by the keyword "tiger" nets a large number of images of a golf player Tiger Woods and the animal tigers in the meantime.

### 1.2 Content-Based Approaches:

This search engine extracts semantic information from image content features, such as colour, shape, texture, spatial location of objects in images, etc [4]-[8]. The extracted visual information is natural and objective, but completely ignores the role of human knowledge in the interpretation process. As the result, a red flower may be regarded as the same as a rising sun, and a fish the same as an airplane etc.

Hybrid approaches: recent research combines both the visual content of images and the textual information obtained from the Web for the WWW image retrieval (Fig.1). Such methods exploit the usage of the visual information for refining the initial text-based search result. Especially, through user's relevance feedback, i.e., the submission of desired images or visual content-based queries, the re-ranking for image search results can achieve significant performance improvement.

In this paper, we propose an automatic annotation method by hybriding decision tree (DT) and support vector machine (SVM) is proposed and a novel inverted file is used to rank the search result. Experiments of both word search and image search in a Corel dataset and a Yahoo! dataset are performed. The preliminary result is satisfied and promising.

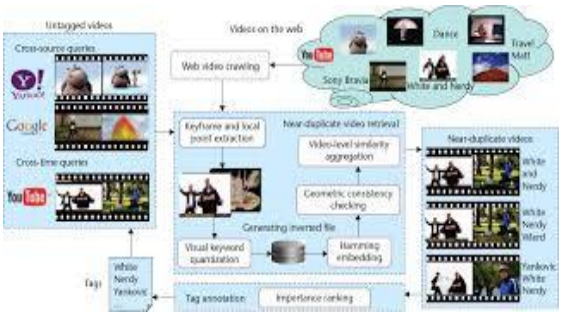


Fig. 1 Bag-based image re-ranking framework for large-scale TBIR.

## II. Relevance Model

Let us formulate the web image retrieval re-ranking problem in a more formal way. For each image I in the rank list returned from a web image search engine, there is one associated HTML document D displaying the image, that is, the HTML document D contains an <img> tag with src attribute pointing to the image I. Since both image understanding and local text information are exploited by the image search engine, we wonder if we can re-rank the image list using global information, i.e. text in the HTML document, to improve the performance. In other words, can we estimate the probability that the image is relevant given text of the document D, i.e. Pr(R|D)? This kind of approach has been explored and called Probability-Based Information Retrieval. By Bayes' Theorem, the probability can be rewritten as follows,

$$Pr(\mathrm{R} \mid \mathrm{D}) = \frac{\Pr(\mathrm{D}|\mathrm{R})\Pr(\mathrm{R})}{\Pr(D)} \quad (1)$$

Since Pr(D) is equal for all documents and assume every document is equally possible, only the relevance model Pr(D|R) is needed to estimate if we want to know the relevance of the document, which consequently implies the relevance of the image within. Suppose the document D is consisted of words {w1, w2, . . . ,wn}. By making the common word independence assumption [21],

$$\Pr(D \mid \mathrm{R}) \approx \prod_{i=1}^{n} \Pr(\omega_i \mid \mathrm{R}) \quad (2)$$

Pr(w|R) can be estimated if training data are available ,i.e. a collection of web pages that are labelled as relevant to the query. However, we cannot afford to collect training data for all possible queries because the number of queries to image search engines everyday is huge.

### 2.1 Approximate Relevance Model

A method, proposed by Lavrenko and Croft [15], offers a solution to approximate the relevance model without preparing any training data. Instead of collecting relevant web pages, we can treat query Q as a short version of relevant document sampling from relevant documents,

$$\Pr(\omega \mid R \approx \Pr(\omega \mid Q) \quad (3)$$

Suppose the query Q contains k words {q1, q2, qk}.Expand the conditional probability in Equation 3,

$$\Pr(\omega \mid Q) = \frac{\Pr(\omega, q_1, q_2, ..., q_k)}{\Pr(q_1, q_2, ..., q_k)} \quad (4)$$

Then the problem is reduced to estimate the probability that word w occurs with query Q, i.e. Pr(w, q1, q2, . . . , qk).First we expand Pr(w, q1, q2, . . . , qk) using chain rule,

$$\Pr(\omega \mid Q) = \Pr(\omega) \prod_{i=1}^{k} \Pr(q_1 \mid \omega, q_{i-1}, ..., q_1) \quad (5)$$

If we further make the assumption that query word q is independent given word w, Equation 5 becomes

$$\Pr(\omega, q_1, q_2, ..., q_k) \approx \Pr(\omega) \prod_{i=1}^{k} \Pr(q_i|\omega) \quad (6)$$

We sum over all possible unigram language models M in the unigram universe to estimate the probability Pr (q|w), as shown in Equation 6. Unigram language model is designed to assign a probability of every single word. Words that appear often will be assigned higher probabilities. A document will provide a unigram language model to help us estimate the co-occurrence probability of w and q.

$$\Pr(\omega, q_1, q_2, ..., q_k) \approx \Pr(\omega) \prod_{i=1}^{k} \sum_{j=1}^{p} \Pr(M_j|\omega) \Pr(q_i|M_j) \quad (7)$$

The approximation modelled in Equation 7 can be regarded as the following generative process: we pick up a word w according to Pr(w), then select models by conditioning on the word w, i.e. Pr(M|w), and finally select a query word q according to Pr(q|M). There are still some missing pieces before we can actually compute the final goal Pr(D|R). Pr(q1, q2, . . . , qk) in Equation 4 can be calculated by summing over all words in the vocabulary set V,

$$\Pr(q_1, q_2, ..., q_k) = \sum_{\omega \in V} \Pr(\omega, q_1, q_2, ..., q_k) \quad (8)$$

Where Pr(w, q1, q2, . . . , qk) is obtained from Equation 8, Pr(w) in Equation 8 can estimated by summing over all unigram models,

$$\begin{aligned} \Pr(\omega) &= \sum_{j=1}^{p} \Pr(M_j, \omega) \\ &= \sum_{j=1}^{p} \Pr(M_j) \Pr(\omega \mid M_j) \end{aligned} \quad (9)$$

It is not a good idea here to estimate the unigram model Pr(w|Mj) directly using maximum likelihood estimation, i.e. the number of times that word w occurs in the document j divided by the total number of words in the document, and some degree of smoothing is usually required. One simple smoothing method is to interpolate the probability with a background unigram model,

$$\Pr(\omega \mid M_j = \lambda \frac{c(\omega, j)}{\sum_{v \in V(j)} c(v, j)} + (1 - \lambda) \frac{c(\omega, G)}{\sum_{v \in V(G)} c(v, G)} \quad (10)$$

Where G is the collection of all documents, c (w, j) is the number of times that word w occurs in the document j, V (j) is the vocabulary in the document j, and $\lambda$ is the smoothing parameter between zero and one.

**2.2 Ranking Criterion**

While it is tempting to estimate Pr(w|R) as described in the previous section and re-rank the image list in the decreasing order of Pr(D|R), there is a potential problem of doing so. Let us look at Equation 2 again. The documents with many words, i.e. long documents, will have more product terms than short documents, which will result in smaller Pr(D|R). Therefore, using Pr(D|R) directly would favour short documents, which is not desirable. Instead, we use Kullback-Leibler (KL) divergence [7] to avoid the short document bias. KL divergence D(p||q) is often used to measure the "distance" between two probability distributions p and q, defined as follows,

$$D(\Pr(\omega \mid D_i \Pr(\omega \mid R)) = \sum_{v \in V} \Pr(v|D_i) \log \frac{\Pr(v, D_i)}{\Pr(v|R)} \quad (11)$$

Where Pr (w|Di) is the unigram model from the document associated with rank i image in the list, and Pr(w|R) is the a fore-mentioned relevance model, and V is the vocabulary. We estimate the unigram model Pr(w|D) for each document associated with an image in the image list returned from image search engine, and then calculate the KL divergence between the Pr(w|D) and Pr(w|R). If the KL divergence is smaller, the unigram is closer to the relevance model, i.e. the document is likely to be relevant. Therefore, the re-ranking process reorders the list in the increasing order of the KL divergence.

We summarize the proposed re-ranking procedure in Figure 3, where the dashed box represents the "Relevance Model Re-ranking" box in Figure 2. Users input a query consisting of keywords {q1, q2, . . . , qk} to describe the pictures they are looking for, and a web image search engine returns a rank list of images. The same query is also fed into a web text search engine, and retrieved documents are used to estimate the relevance model Pr(w|R) for the query Q. We then calculate the KL divergence between the relevance model and the unigram model Pr(w|D) of each document D associated with the image I in the image rank list, and re-rank the list according to the divergence.

## III. Performance And Experimental Results

We tested the idea of re-ranking on six text queries to a large-scale web image search engine, Google Image Search [10], which has been on-line since July 2001. As of March 2003, there are 425 million images indexed by Google Image Search. With the huge amount of indexed images, there should be large varieties of images, and testing on the search engine of this scale will be more realistic than on an in-house, small-scale web image search system. Six queries are chosen, as listed in Table 1, which are among image categories in Corel

Image Database. Corel Database is often used for evaluating image retrieval [5] and classification [17]. Each text query is typed into Google Image Search, and top 200 entries are saved for evaluation. The default browsing setting for Google Image Search is to return 20 entries per page, and thus 200 entries takes users ten time "Next" button clicks to see all the results, which should reasonably bound the maximum number of entries that most users will check.

Each entry in the rank list contains a filename, image size, image resolution, and URL that points to the image. We build a web crawler program to fetch and save both the image and associated HTML document for each entry. After total 1200 images for six queries are fetched, they are manually labelled into three categories: relevant, ambiguous, and irrelevant (Fig.1). If the image is obviously a wrong match, it will be labelled irrelevant, otherwise will be labelled as ambiguous. Both irrelevant and ambiguous are considered as "irrelevant" when we evaluate the performance. As shown in the third column of Table 1, the number of the relevant images varies much from query to query, indicating the difficulty of the query.

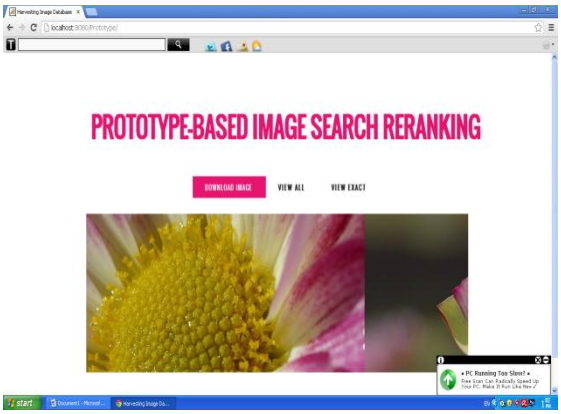


Fig.1 Home Page

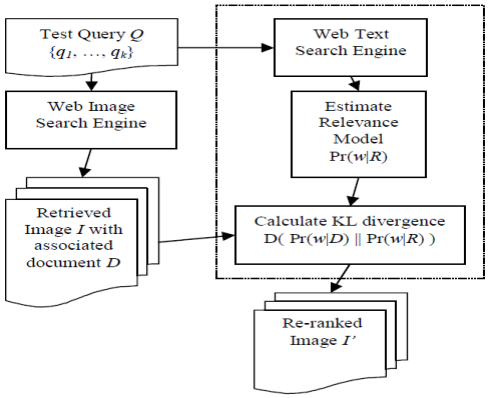


Fig 2. Pictorial Summary of relevance model estimation

### 3.1. Relevance Model Estimation

We also feed the same queries to a web text search engine, Google Web Search [12], to obtain text documents for estimating relevance model.

Google Web Search, based on Page Rank algorithm [3], is a large-scale and heavily-used web text search engine. As of March 2003, there are more than three billions of web pages indexed by Google Web Search. There are 150 millions queries to Google Web.

Table 1.six search queries

| Query No. | Text Query | Number of Relevant Images In Top 200 |
|---|---|---|
| 1 | Birds | 51 |
| 2 | Food | 117 |
| 3 | Fish | 73 |
| 4 | Fruits and vegetables | 117 |
| 5 | Sky | 78 |
| 6 | Flowers | 90 |

Search every day. With the huge amounts of indexed web pages, we expect top-ranked documents will be more representative, and relevance model estimation will be more accurate and reliable for each query, we send the same keywords to Google Web Search and obtain a list of relevant documents via Google Web APIs [11]. Before calculating the statistics from these top-ranked HTML documents, we remove all HTML tags, filter out words appearing in the INQUERY [4] stop word list, and stem words using Porter algorithm [19], which are all common pre processing in the Information Retrieval systems [2], and usually improve retrieval performance. The relevance model is estimated in the same way described before. The smoothing parameter $\lambda$ in Equation 11 is empirically set to 0.6.

### 3.2 Evaluation Metric

Recall and precision are common metrics used to evaluate information retrieval systems. Given a rank list with length n, precision is defined as r/n, recall as r/R where r is the number of documents that is truly relevant in the list, and R is the total number of relevant documents in the collection. The goal of any retrieval system is to achieve as higher recall and precision as possible. Here we choose precision at specific document cut-off points (DCP) as the evaluation metric, i.e. calculate the precision after seeing 10, 20,. . . ,200 documents. In the web search setting, users usually have limiting time to browse results, and different methods should be compared after users spend the same efforts of browsing. It should be more reasonable to praise a system that can find more relevant documents in the top 20 results (a specific DCP), rather than at 20% recall which is Precision-Recall curve calculation is based on, because 20% recall can mean different numbers of documents that have to be evaluated by users.

For example, 20% recall means the top 10 documents for the Query 1, but means the top 23 documents for Query 2.In the low DCP, precision is more accurate than recall [13].Since possible relevant images on the Internet are far larger than we retrieved, 200 documents are regarded as a very low DCP, and therefore only precision is calculated.

### 3.3 Results

The comparison of performance before and after re-ranking is shown in (Fig 3). The average precision at the top 50 documents, i.e. in the first two to three result pages of Google

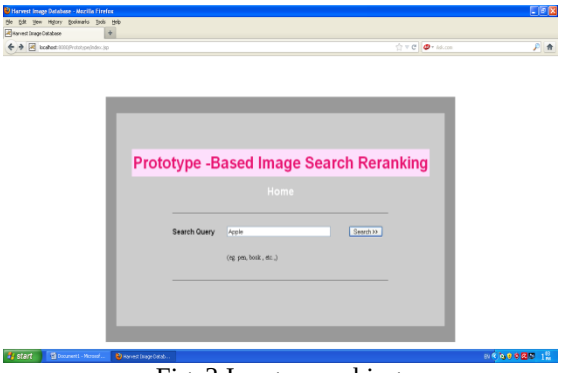


Fig. 3 Image searching

Image Search, has remarkable 30% to 50% increases (recall from original 30-35%to 45%after re-ranking).

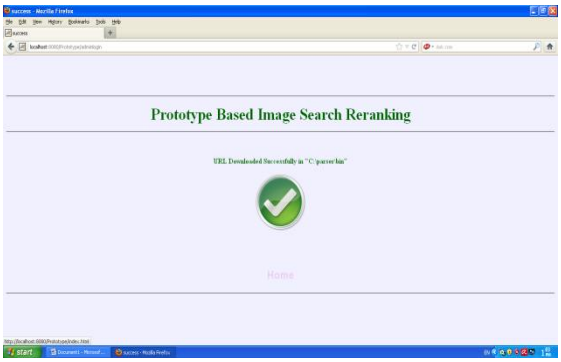


Fig.4 Deleting the irrelevant images.

Even testing on such a high-profile image search engine, the re-ranking process based on relevance model still can improve the performance, suggesting that global information from the document can provide additional clues to judge the relevance of the image. Internet users are usually with limit time and patience, and high precision at top-ranked documents will save user a lot of efforts and help them find relevant images more easily and quickly.

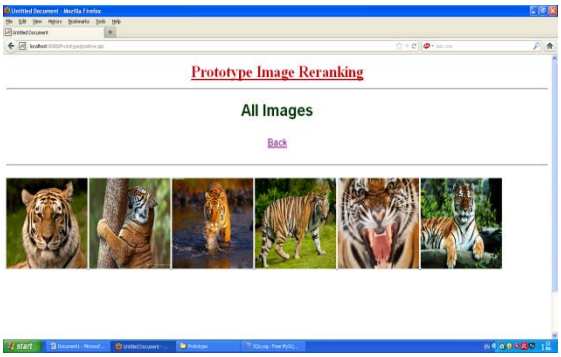


Fig.5 finally the score for all meta re-rankers.

## IV. Discussions

Let us revisit at the relevance model Pr(w|R), which may explain why re-ranking based on relevance model works and where the power of the relevance model comes from. In Appendix A, top 100 word stems with highest probability Pr(w|R) from each query are listed. It appears that many words that are semantics related to the query words are assigned with high probability by the relevance model. For example, in Query 3 "fish", there are marine (marine in stemmed form), aquarium, seafood, salmon, bass, trout, shark, etc.

In Query 1 "birds", we can see bird watch, owl, parrot, ornithology (ornithology in stemmed form), sparrow, etc. It is the ability to correctly assign probability to semantic related terms that relevance model can make a good guess of the relevance of the web document associated with the image.

If the web page contains words that are semantics relevant to the query words, the images within the page will be more likely to be relevant. Recall we feed the same text query into a web text search engine to obtain top 200 documents when we estimate the co-occurrence probability of the word w and the query Q in Equation 8. These 200 documents are supposed to highly relate to the text query, and words occur in these documents should be very much related to the query. The same idea with a different name called pseudo relevance feedback has been proposed and shown performance improvement for text retrieval [22]. Since no humans are involved in the feedback loop, it is a "pseudo" feedback by blindly assuming top 200 documents and relevant.

The relevance model estimates the co-occurrence probability from these documents, and then re-ranks the documents associated the images. The relevance model acquires many terms that are semantics related the query words, which in fact equals to query expansion, a technique widely used in Information Retrieval community. By adding more related terms in the query, the system is expected to retrieve more relevant documents, which is similar to use relevance model to re-rank the documents. For example, it may be hard to judge the relevance of the document using single query word "fish", but it will become easier if we take terms such as "marine", "aquarium", "seafood", "salmon" into consideration, and implicitly images in the page with many fish-realted terms should be more likely to be real fish. The best thing about relevance model is that it is learned automatically from documents on the Internet, and we do not need to prepare any training documents.

## V. Conclusion And Future Work

Re-ranking web image retrieval can improve the performance of web image retrieval, which is supported by the experiment results. The re-ranking process based on relevance model utilizes global information from the image's HTML document to evaluate the relevance of the image. The relevance model can be learned automatically from a web text search engine without preparing any training data. The reasonable next step is to evaluate the idea of re-ranking on more and different types queries. At the same time, it will be infeasible to manually label thousands of images retrieved from a web image search engine. An alternative is task-oriented evaluation, like image similarity search. Given a query from Corel Image Database, can we re-rank images returned from a web image search engine and use top-rank images to find similar images in the database. We then can evaluate the performance of the re-ranking process on similarity search task as a proxy to true objective function.

Although we apply the idea of re-ranking on web image retrieval in this paper, there are no constraints that re-ranking process cannot be applied to other web media search. Re-ranking process will be applicable if

the media files are associated with web pages, such as video, music files, MIDI files, speech wave files, etc. Re-ranking process may provide additional information to judge the relevance of the media file.

## References


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

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In Proceedings of 7th International World-Wide Web Conference, 1998.

[3] J. P. Callan, W. B. Croft, and S. M. Harding. The INQUERY retrieval system. In Proceedings of 3rd International Conference on Database and Expert Systems Applications, 1992.

[4] D. Hull. Using statistical testing in the evaluation of retrieval experiments. In Proceedings of the 16th SIGIR Conference, (1993), 329–338.

[5] J. Kleinberg. Authoritative sources in a hyperlinked environment. Journal of ACM, 46(5),(1999),604–632.

[6] V. Lavrenko and W. B. Croft. Relevance-based language models. In Proceedings of the International ACM SIGIR Conference, 2001.

[7] R. Lempel and A. Soffer. Pic A SHOW: Pictorial authority search by hyperlinks on the web. In Proceedings of the 10th International World-Wide Web Conference, 2001.

[8] O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classification. In Proceedings of the 15th International Confernece on Machine Learning (1998), 341–349.

[9] M. Porter. An algorithm for suffix stripping. Program, 14(3),(1980),130–137.

[10] C. J. van Rijsbergen. A theoretical basis for the use of cooccurrence data in information retrieval. Journal of Documentation33 ,( 1977),106–119

[11] X. S. Zhou and T. S. Huang, "Relevance feedback in image retrieval: A comprehensive review," Multimedia Syst., vol. 8, no. 6, (2003)536–544.

[12] X. Tian, D. Tao, X.-S. Hua, and X. Wu, "Active reranking for web image search," IEEE Trans. Image Process., vol. 19, no. 3, (Mar. 2010) 805–820.

[13] T.-Y. Liu, "Learning to rank for information retrieval," Found. Trends Inf. Retr., vol. 3,( Mar. 2009) 225–331.