Global Journal of Engineering Science and Research Management

SEMANTIC SIGNATURE BASED IMAGE SEARCH IN WEB

V.Praveena*, S.Karthik

*M.E Department of Computer Science and Engineering Krishnasamy college of Engineering and Technology.

Associate Professor Department of Computer Science and Engineering Krishnasamy college of Engineering and Technology.

KEYWORDS: Approximate Relevance Model, Evaluation Metric, Relevance Model Estimation.

ABSTRACT

Web image retrieval may be a difficult task that needs efforts from image process, link structure analysis, and net text retrieval. Since content-based image retrieval continues to be thought-about terribly difficult, most current large-scale net image search engines exploit text and link structure to “understand” the content of the online pictures. However, lo- cal text info, like caption, filenames and adjacent text, isn't invariably reliable and informative. Therefore, international info ought to be taken under consideration once an internet image retrieval system makes connection judgment. During this paper, we have a tendency to propose a re-ranking methodology to enhance net image retrieval by rearrangement the pictures retrieved from a picture computer programme. The re-ranking method is predicated on a connection model that may be a probabilistic model that evalu-ates the connection of the hypertext markup language document linking to the image, and assigns a likelihood of connection. The experiment results showed that the re-ranked image retrieval achieved higher performance than original net image retrieval, suggesting the effectiveness of the re-ranking methodology.

INTRODUCTION

As World-Wide Web develops in a blasting rate, web crawlers get to be basic instruments for any clients who search for data on the Internet, and web picture inquiry is no exemption. Web picture recovery has been investigated and created by scholastic specialists and in addition commercial organizations, including scholarly models (e.g. Vi-sualSEEK [20]), extra hunt measurement of existing web internet searchers (e.g. Google Image Search [10], Al-taVista Image [1], particular web picture web indexes (e.g. Same [8], PicSearch [18]), and web interfaces to com-mercial picture suppliers (e.g. Getty Images [9], Corbis [6]). In spite of the fact that ability and scope differ from framework to framework, we can arrange the web picture internet searchers into three flavors regarding how pictures are ordered. The first is content based list. The representation of the im-age incorporates filename, inscription, encompassing content, and content in the HTML record that shows the picture. The sec-ond one is picture based file. The picture is spoken to in visual elements, for example, shading, composition, and shape. The third one is cross breed of content and picture list. In any case, content based list is by all accounts the common decision now in the event that anybody arrangements to fabricate a huge scale web picture recovery framework. Possible reasons include: content data interface permits clients to ex-press their data require more effortlessly than picture between face, (requesting that clients give a specimen picture or drawing a scratch is from time to time doable), picture comprehension is still an open exploration issue, and picture based record are typically of high dimensionality.

Most web image search engines provide a text input in-terface (like HTML tag <INPUT>) that users can type key-words as a query. The query is then processed and matched against the indexed web images, and a list of candidate im-ages are ranked in the order of relevance before results are returned to users, as illustrated in Figure 1.

However, textual representation of an image is often am-biguous and non-informative of the actual image content. Filenames may be misleading, adjacent text is difficult to define, and a word may contain multiple senses. All these factors confound the web image retrieval system. More con-text cues should be taken into consideration when the web image retrieval systems managed to disambiguates and rank images.

One piece of information in the HTML documents that can help make relevance judgment is link structure. Sophisticated algorithms such as PageRank [3] , “Hub and Au-

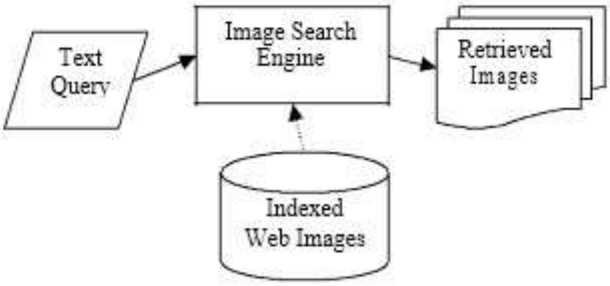


Figure 1. A high-level overview of web image search engine

thorities" [14] rank documents by analyzing the link struc-ture between documents. A document is more important if it links many "good" pages, and many "good" pages link it. Similar ideas have been applied to web image retrieval (e.g. PicASHOW [16]), and images are ranked by consider-ing the web page is an image container or a hub. However, for outsiders to make use of link structure, the index infor-mation of the web image search engine must be publicly accessible, which is unlikely and sometimes impossible.

In the paper, we propose a re-ranking process to reorder the retrieved images. Instead of accepting the results from a web image search engine, the image rank list as well as associated HTML documents are fed to a re-ranking pro-cess. The re-ranking process analyzes the text of the HTML document associated with the images to disambiguate doc-ument/image relevance using a relevance model. The rele-vance model is built automatically through a web text search engine. The re-ranking process (above the dashed line) is il-lustrated in Figure 2.

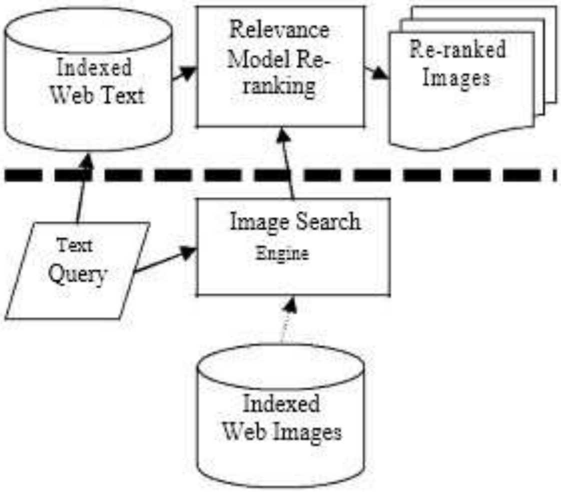


Figure 2. An overview of web image retrieval re-ranking

The basic idea of re-ranking is that the text part of HTML documents (i.e., after removal of all HTML tags in the HTML documents) should be relevant to the query if the image displayed in the document is relevant. For example, when a user input a text query "Statue of Liberty" to a web image search engine, we expect the web pages with images relevant to query is more likely to be history or travel infor-mation for "Statue of Liberty", but less likely to be pages describing a ship happening to be named after "Statue of Liberty".

We describes the relevance model, the key component in the re-ranking process, in Section 2. Experiments are con-ducted to test the re-ranking idea in Section 3. The connec-tion between relevance model re-ranking to the Information Retrieval techniques are discussed in Section 4. Finally we conclude the paper, and present some directions of future works.

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 un-derstanding 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 docu-ment, 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 [2].

By Bayes' Theorem, the probability can be rewritten as follows,

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

Since Pr(*D*) is equal for all documents and assume ev-ery document is equally possible, only the relevance model Pr(*D*|*R*) is needed to estimate if we want to know the rele-vance of the document, which consequently implies the rel-evance of the image within.

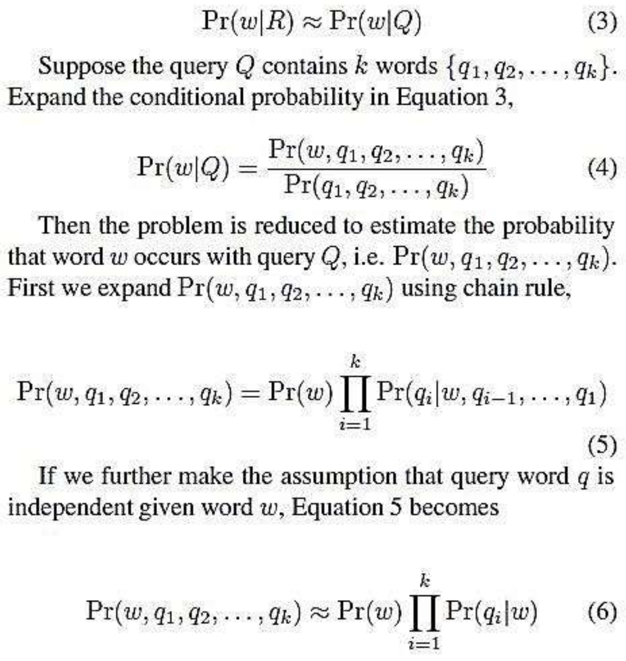
Suppose the document *D* is consisted of words $\{w_1, w_2, \ldots, w_n\}$. By making the common word indepen-dence assumption [21],

$$\Pr(D|R) \approx \Pr(w_i\,|R) \qquad (2)$$

Pr(*w*|*R*) can be estimated if training data are available, i.e. a collection of web pages that are labeled 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 every day is huge.

2.1. Approximate Relevance Model

A method, proposed by Lavrenko and Croft [15], of-fers 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 rele-vant document sampling from relevant documents,



$$\Pr(w|R) \approx \Pr(w|Q) \qquad (3)$$

Suppose the query Q contains k words $\{q_1, q_2, \ldots, q_k\}$. Expand the conditional probability in Equation 3,

$$\Pr(w|Q) = \frac{\Pr(w, q_1, q_2, \ldots, q_k)}{\Pr(q_1, q_2, \ldots, q_k)} \qquad (4)$$

Then the problem is reduced to estimate the probability that word w occurs with query Q, i.e. $\Pr(w, q_1, q_2, \ldots, q_k)$. First we expand $\Pr(w, q_1, q_2, \ldots, q_k)$ using chain rule,

$$\Pr(w, q_1, q_2, \ldots, q_k) = \Pr(w) \prod_{i=1}^{k} \Pr(q_i|w, q_{i-1}, \ldots, q_1) \qquad (5)$$

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

$$\Pr(w, q_1, q_2, \ldots, q_k) \approx \Pr(w) \prod_{i=1}^{k} \Pr(q_i|w) \qquad (6)$$

We sum over all possible unigram language models *M* in the unigram universe Ξ to estimate the probability Pr(*q*|*w*), as shown in Equation 7 . Unigram language model is de-signed 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*.

$$\Pr(w, q_1, q_2, \ldots, q_k) = \Pr(w) \prod_{i=1}^{k} \sum_{M \in \Xi} \Pr(q_i, M|w) \quad (7)$$

In practice, we are unable to sum over all possible uni-gram models in Equation 7, and usually we only consider a subset. In this paper, we fix the unigram models to top-ranked *p* documents returned from a text web search engine given a query *Q*.

If we further assume query word *q* is independent of word *w* given the model *M* , Equation 7 can be approximated as follows,

$$\Pr(w, q_1, q_2, \ldots, q_k) \approx \Pr(w) \prod_{i=1}^{k} \sum_{j=1}^{p} \Pr(M_j|w) \Pr(q_i|M_j) \quad (8)$$

The approximation modeled in Equation 8 can be re-garded as the following generative process: we pick up a word *w* according to Pr(*w*), then select models by condi-tioning on the word *w*, i.e. Pr(*M* |*w*), and finally select aquery word *q* according to Pr(*q*|*M*).

There are still some missing pieces before we can actu-ally compute the final goal Pr(*D*|*R*). Pr(q_1, q_2 , . . . , q_k) in Equation 4 can be calculated by summing over all words in the vocabulary set *V*,

$$\Pr(q_1, q_2, \ldots, q_k) = \sum_{w \in \mathcal{V}} \Pr(w, q_1, q_2, \ldots, q_k) \quad (9)$$

where $\Pr(w, q_1, q_2, \ldots, q_k)$ is obtained from Equation 8, $\Pr(w)$ in Equation 8 can estimated by summing over all unigram models,

$$\begin{aligned}\Pr(w) &= \sum_{j=1}^{p} \Pr(M_j, w) \\ &= \sum_{j=1}^{p} \Pr(M_j) \Pr(w|M_j)\end{aligned} \quad (10)$$

It is not a good idea here to estimate the unigram model $\Pr(w|M_j)$ 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,

$$\Pr(w|M_j) = \lambda \frac{c(w,j)}{\sum_{v \in \mathcal{V}(j)} c(v,j)} + (1-\lambda) \frac{c(w,G)}{\sum_{v \in \mathcal{V}(G)} c(v,G)} \quad (11)$$

where G is the collection of all documents, $c(w,j)$ is the number of times that word w occurs in the document j, $\mathcal{V}(j)$ is the vocabulary in the document j, and λ 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 favor 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,

$$D(\Pr(w|D_i)||\Pr(w|R)) = \sum_{v \in \mathcal{V}} \Pr(v|D_i) \log \frac{\Pr(v|D_i)}{\Pr(v|R)} \quad (12)$$

where $\Pr(w|D_i)$ is the unigram model from the document associated with rank i image in the list, and $\Pr(w|R)$ is the aforementioned relevance model, and $\mathcal{V}$ 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 $\{q_1, q_2, \ldots, q_k\}$ 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.

3. Experiments

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 labeled into three categories: relevant, ambiguous, and irrelevant. An image is labeled as relevant if it is clearly a natural, non-synthesized image with desired objects described by the query, and can be identified instantly by human judges. If the image is obviously a wrong match, it will be labeled irrelevant, otherwise will be labeled 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.

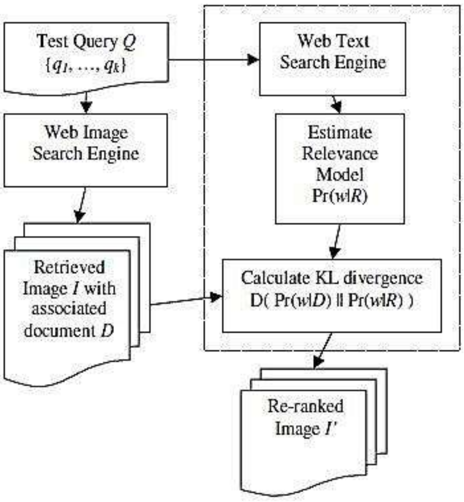

Test Query Q
$\{q_1, \ldots, q_k\}$
Web Text Search Engine
Web Image Search Engine
Estimate Relevance Model $\Pr(w|R)$
Retrieved Image I with associated document D
Calculate KL divergence $D(\Pr(w|D) \parallel \Pr(w|R))$
Re-ranked Image I'

Figure 3. A 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 PageRank 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 rep-resentative, 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]. Top-ranked 200 web documents, i.e. p equals 200 in Equation 8, in the list are further fetched using a web crawler. 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 per-formance. The relevance model is estimated in the same way described before. The smoothing parameter λ in Equa-tion 11 is empirically set to 0.6.

3.2. Evaluation Metric

Recall and precision are common metrics used to eval-uate information retrieval systems. Given a rank list with length n, precision is defined as $\frac{r}{n}$, recall as $\frac{r}{R}$, 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 collec-tion. 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.

We choose precision at DCP over traditional Recall-Precision curve is because DCP precision will reflect more closely the browsing behavior of users on the Internet. 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 rele-vant documents in the top 20 results (a specific DCP), rather than at 20% recall which is Precision-Recall curve calcu-lation 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 Figure 4. The average precision at the top 50 documents, i.e. in the first two to three re-sult pages of Google Image Search, has remarkable 30% to 50% increases (recall from original 30-35% to 45% after re-ranking). 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 cues to judge the relevance of the image.

The improvement at the high ranks is a very desirable property. 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.

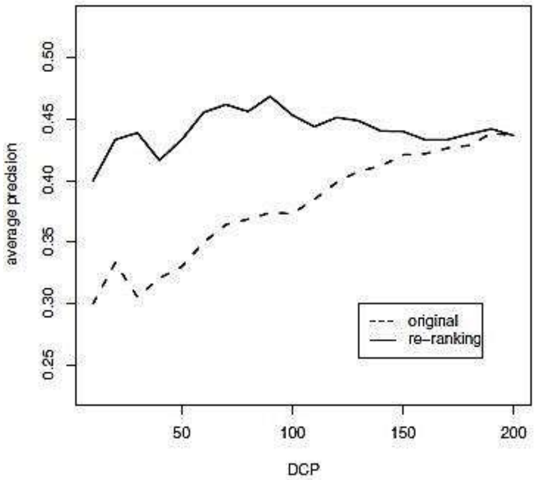


Figure 4. The average precision at DCP over six queries showed re-ranking process made remarkable improvements, especially at the higher ranks

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 ap-pears that many words that are semantics related to the query words are assigned with high probability by the rele-vance model. For example, in Query 3 "fish", there are marine (marin in stemmed form), aquarium, seafood, salmon, bass, trout, shark, etc. In Query 1 "birds", we can see birdwatch, owl, parrot, ornithology (ornitholog in stemmed form), sparrow, etc. It is the ability to correctly assign prob-ability 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 docu-ments should be very much related to the query. The same idea with a different name called pseudo relevance feed-back has been proposed and shown performance improve-ment 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 widly used in Information Re-trieval community. By adding more related terms in the query, the system is expected to retrieve more relevant doc-uments, 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 doc-uments on the Internet, and we do not need to prepare any training documents.

CONCLUSIONS AND FUTURE WORKS

Re-ranking web image retrieval can improve the perfor-mance of web image retrieval, which is supported by the experiment results. The re-ranking process based on rele- vance 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 of queries. At the same time, it will be infeasible to manually label thousands of im-ages retrieved from a web image search engine. An alterna-tive 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 func-tion.

Although we apply the idea of re-ranking on web im-age 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. Altavista image. http://www.altavista.com/image/.
2. R. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval. Addison-Wesley, 1999.
3. S. Brin and L. Page. The anatomy of a large-scale hypertex-tual web search engine. In Proceedings of 7th International World-Wide Web Conference, 1998.
4. J. P. Callan, W. B. Croft, and S. M. Harding. The IN-QUERY retrieval system. In Proceedings of 3rd Interna-tional Conference on Database and Expert Systems Appli-cations, 1992.
5. C. Carlson and V. E. Ogle. Storage and retrieval of feature data for a very large online image collection. Bulletin of the Technical Committee on Data Engineering, 19(4), 1996.
6. Corbis. http://www.corbis.com/.
7. T. M. Cover and J. A. Thomas. Elements of Information Theory. Wiley-Interscience, 1991.

8. Ditto. http://www.ditto.com/.
9. Getty images. http://creative.gettyimages.com/.
10. Google Image Search. http://images.google.com/.
11. Google Web APIs. http://www.google.com/apis/.
12. Google Web Search. http://www.google.com/.
13. D. Hull. Using statistical testing in the evaluation of retrieval experiments. In Proceedings of the 16th SIGIR Conference, pages 329–338, 1993.
14. J. Kleinberg. Authoritative sources in a hyperlinked envi-ronment. Journal of ACM, 46(5):604–632, 1999.
15. V. Lavrenko and W. B. Croft. Relevance-based language models. In Proceedings of the International ACM SIGIR Conference, 2001.