IMAGE COLLECTOR II: A SYSTEM FOR GATHERING MORE THAN ONE THOUSAND IMAGES FROM THE WEB FOR ONE KEYWORD

Keiji Yanai

Department of Computer Science, The University of Electro-Communications

Abstract

We propose a system that enables us to gather more than one thousand images from the World Wide Web. The system is called Image Collector II. The Image Collector, which we proposed previously, can gather only several hundreds images. We made the two following improvements to extend the ability of our previous system in terms of the number of gathered images and their precision: (1) We extracted some words appearing with high frequency from all HTML files embedding output images in an initial image gathering, and using them as keywords, we made a second image gathering again. Through this, we obtained more than one thousand images for one keyword. (2) The more images we gathered, the more he precision of gathered images decreased. To raise the precision, we introduced word vectors of HTML files embedding images into the image selecting process in addition to image feature vectors.

1. INTRODUCTION

Due to the explosion in growth of the World Wide Web, we can easily access a large number of images. Therefore, we can consider the web as a huge image database. However, most of those images are not categorized in terms of their content and are not labeled with related keywords. We can use commercial image search engines such as Google Image Search and Ditto to search the web for image files by giving them keywords. However, most of image search engines search for images based only on keywords in HTML documents that include images, and they do so without analyzing the content of the images. As a result, they tend to return many irrelevant images to the given keywords.

As a method of an image search, content-based image retrieval (CBIR) has been researched [1]. In CBIR, the similarity between images is computed using features extracted from images, and we can search similar images to query images.

To achieve an image search for the web based on not only keywords but also the content of images, we proposed an automatic image gathering system for the web that is constructed by integrating keyword-based search and CBIR methods, which is called Image Collector [2]. In the system, a user first gives query keywords to the system, and then it obtains images associated with the keywords. First, using the existing commercial web search engines for HTML documents, the system gathers images embedded in HTML documents related to the query keywords. Next, the system selects output images from collected images based on extracted image features. The Image Collector, however, can gather only several hundreds images. Our initial objective for this system was to gather a large number of images for web image mining research, but the number of gathered images turned out to be insufficient[3].

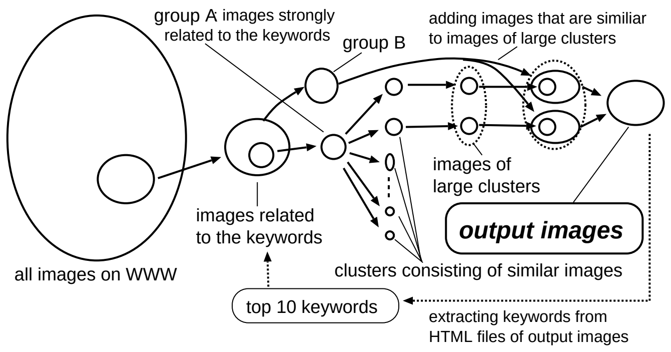
group A: images strongly related to the keywords
group B
adding images that are similiar to images of large clusters
images of large clusters
images related to the keywords
output images
all images on WWW
clusters consisting of similar images
top 10 keywords
extracting keywords from HTML files of output images

Figure 1: Processing flow of Image Collector II.

Then, we made the two following improvements on it to extend the power of our system:

(1) We extracted words that appear with high frequency from all HTML files with embedded output images in an initial image gathering and used them as keywords for gathering images from the web again. Through this, we obtained more than one thousand images using one keyword.
(2) The more images we gathered, the more the precision of the gathered images decreased. To raise the precision, we introduced word vectors of HTML files with embedded images into the image selecting process in addition to image feature vectors.

2. OVERVIEW

Figure 1 shows the processing flow. Because an image on the web is usually embedded in an HTML document that explains its content, the system uses keywords to exploit some existing commercial web text search engines such as Google and Lycos, and it gathers URLs of HTML documents related to the query keywords. In the next step, using those gathered URL-s, the system fetches HTML documents from the web, and evaluates the relevancy of the keywords for images by analyzing them. If the images are judged to be related to the keywords, the image files are fetched. Based on the extent of the relevancy to the keywords, fetched images are divided into two groups. Images put in group A have stronger relation to the keywords than those put in group B. Image features are computed for all images,

In the CBIR, a user has to provide query images to the system because it searches for images based on the similarity of image features between the query images and the images in an image database. In our system, instead of providing query images, a user only needs to provide keywords. It selects images strongly related to the keywords as group A images, removes noise images from them, and regards them as query images only by examining the keywords. Noise images are removed by eliminating images that belong to relatively small clusters in the results of image feature clustering for group A images. Images that are not eliminated are regarded as being appropriate to the query keywords and are stored as output images. Our preference for larger clusters was based on the following heuristic observation. An image that has many similar ones is usually more suitable for an image represented by keywords than for one that has only a few. Next, our system selects images that are similar to the query images from group B in the same way as the CBIR and adds them to the output images.

Some web image search systems such as WebSeer[4], WebSEEk[5], and Image Rover[6] have been reported so far. These systems can be regarded as an integration of keyword-based search and content-based image retrieval. They search for images based on the keywords, and then a user selects query images from the search results. After this selection, the systems search for images that are similar to the query images based on the image features. These three systems search in an interactive manner. Our system is different from those in that it only needs a one-time input of query keywords. It is able to gather a large number of various images related to the keywords because a user does not need to indicate query images during the processing and because the whole processing is executed automatically. The three aforementioned systems require gathering images over the web in advance and making big indices of images on it. In contrast to those systems, ours does not require making a large index in advance because it exploits existing text search engines.

3. COLLECTION AND SELECTION

The processing of Image Collector II consists of collection and selection stages. In addition, it extracts additional keywords and repeats two kinds of stages during a second image gathering.

3.1. Collection Stage

In the collection stage, the system obtains URLs using some commercial web search engines, and by using those URLs, it gathers images from the web. The algorithm is as follows.

1. A user provides the system with two kinds of query keywords. One is a main keyword that best represents an image, and the other is an optional subsidiary keyword. For example, when we gather "lion" images, we use "lion" as a main keyword and "animal" as a subsidiary keyword.
2. The system sends the main and subsidiary keywords as queries to the commercial search engines and obtains the URLs of the HTML documents related to the keywords.
3. It fetches the HTML documents indicated by the URLs.
4. It analyzes the HTML documents, and extracts the URLs of images embedded in the HTML documents with image-embedding tags ("`IMG SRC`" and "`A HREF`"). For each of those images, the system calculates a score that represents the intensity of the relation between the image and the query keywords. The score is calculated by checking the following conditions.

   Condition 1: Every time one of the following conditions is satisfied, 3 points are added to the score.
   - If the image is embedded by the "`SRC IMG`" tag, the "`ALT`" field of the "`SRC IMG`" includes the keywords.
   - If the image is linked by the "`A HREF`" tag directly, the words between the "`A HREF`" and the "`/A`" include the keywords.
   - The name of the image file includes the keywords.

   Condition 2: Every time one of the following conditions is satisfied, 1 point is added.
   - The "`TITLE`" tag includes the keywords.
   - The "`H1, ..,H6`" tags include the keywords, assuming these tags are located just before the image-embedding one.
   - The "`TD`" tag including the image-embedding tag includes the keywords.
   - Ten words just before the image-embedding tag or ten words after it include the keywords.

   If the final score of an image is higher than 3, the image is classified into group A. If it is higher than 1, the image is classified into group B. The system only fetches files whose images belong to either group A or B.

3.2. Selection Stage

In the selection stage, the system selects more appropriate images for the query keywords out of the ones gathered in the collection stage. The selection is based on the image features described below.

1. The system first makes image feature vectors for all the collected images. Our system uses a $6 \times 6 \times 6$ color histogram in the $Lu^{*}v^{*}$ color space. The dimensions of two kinds of features are 216.
2. For images in group A, the distance (dissimilarity) between two images is calculated based on the quadratic form distance [7].
3. Based on the distance between images, images in group A are grouped by the hierarchical cluster analysis method. Our system uses the farthest neighbor method (FN). In the beginning, each cluster has only one image, and the system repeats merging clusters until all distances between them are more than a certain threshold.
4. It throws away small clusters that have fewer images than a certain threshold value, regarding them as being irrelevant. It stores all images in the remaining clusters as output images.
5. It selects images from group B whose distances to the images in the remaining clusters of group A are small and adds them to the output images.

After this image-feature-based selection, our system carries out the second selection for group B images by using word vectors extracted from the HTML documents with embedded images. Introducing the word vectors enables it to eliminate images embedded in the HTML documents whose topics are irrelevant and to ignore them.

6. The system eliminates HTML tags and extracts words (only nouns, adjectives, and verbs) from HTML documents with embedded images selected by the aforementioned image feature selection. It counts the frequency of appearance of each word in the HTML documents, selects the top 500 words in terms of the frequency, and makes a 500-dimensional word vector whose elements are word frequencies weighted by Term Frequency and Inverse Document Frequency (TFIDF)[8] for each of the images. All word vectors W_i are normalized so that $|W_i| = 1$.

   In addition, we also made experiments using the Latent Semantic Indexing (LSI) methods [9]. These methods compress word vectors with singular value decomposition like Principal Component Analysis (PCA). We compressed a 500-dimensional word vector into one of 100 dimensions.
7. For selected images in group A, it clusters their word vectors based on the k-means clustering method. In the experiments, we set the number of clusters generated in the image-feature-based selection for group A to k. We used the inner product as the dissimilarity (distance) between word vectors.
8. From selected images in group B, it picks up images whose distance to the masses of clusters in group A is less than a certain threshold in terms of word vectors. They are output images of group B found by the word-feature-based selection.

3.3. Second Image-gathering

In our image-gathering method, the more URLs of HTML documents we obtained, the more images we could gather. However, for one set of query keywords, the number of URLs obtained from Web search engines was limited because commercial search engines restrict the maximum number of URLs returned for one query. Thus, we propose a method to generate automatically new sets of query keywords for search engines.

The system extracts the top ten words (only nouns, adjectives, and verbs) with high frequency except for initial query keywords from all HTML files with embedded output images of the initial image gathering, and regards them as subsidiary query keywords. It generates ten sets of query keywords by adding each of ten subsidiary words to a main keyword, and then obtains a large number of URLs for the ten sets of query keywords. Then, for the second image gathering, using obtained URLs, the system goes through the collection and selection stages again.

4. EXPERIMENTS

We implemented an experimental system in C++ and Perl on a Linux-based PC (CPU: AthlonXP 1.6Ghz, memory: 1GB). In the experiments, we limited the Web sites to access only Japanese (.jp) domains.

The experimental results of the initial image gathering for eight keywords are shown in Table 1, which describes the number of URLs of the HTML documents obtained from the search engines, the number of images collected from the web, and the number of selected images using three methods, image-feature-based, a combination of image-feature-based and word-feature-based, and a combination of image-feature-based and word-feature-based with LSI compression. The table also describes the precision rate of the collected and selected images in the parentheses. The precision represents the ratio of relevant images and was computed by the subjective evaluation.

In the collection stage, we used four major Japanese search engines, Google Japan, Goo, Infoseek Japan, and Excite Japan to obtain URLs related to query keywords and merged the search results of these four engines by omitting duplications. In each experiment, we obtained about 2000 URLs.

In the selection stage, the average precision rose from 57.7% to 65.2% after the image-feature-based selection. Moreover, after the word-vector-based selection and the LSI-based selection, it rose to 68.0% and 70.1%, respectively. This shows the effectiveness of introducing word vectors.

Table 2 shows the experimental results of the second image gathering for eight keywords. The number of URLs obtained from the search engine on average was 5.1 times that of the initial image gathering. For "lion", we extracted ten words as new subsidiary key-

Table 1: Results of initial image gathering.

	num.	collected images			selected images				
keyword	URLs	A	B	A+B	A	B	A+B	word vec.	LSI
apple	2247	268 (74)	414 (65)	682 (68.9)	223 (73)	140 (65)	363 (70.1)	355 (70.9)	340 (72.5)
lion	2127	126 (90)	200 (41)	326 (60.8)	113 (90)	72 (43)	185 (72.3)	154 (75.8)	144 (78.9)
baby	2312	529 (69)	265 (46)	794 (60.5)	455 (72)	89 (53)	544 (67.6)	527 (69.3)	512 (71.6)
notebook PC	2151	479 (66)	945 (48)	1424 (55.7)	340 (69)	462 (54)	802 (60.8)	677 (62.8)	620 (66.2)
ramen †	2225	901 (78)	1695 (55)	2596 (67.0)	699 (78)	793 (59)	1492 (71.4)	1434 (71.4)	1385 (71.6)
Shinkansen ††	2181	496 (68)	1146 (40)	1642 (54.1)	428 (70)	604 (43)	1032 (59.3)	744 (64.2)	665 (65.9)
Kinkaku temple	2133	497 (69)	521 (27)	1018 (47.6)	403 (72)	181 (24)	584 (56.6)	521 (60.5)	485 (65.0)
Nomo ‡	2039	96 (72)	124 (25)	220 (46.9)	83 (72)	25 (32)	108 (63.4)	95 (68.9)	90 (69.3)
average	2177	424 (73)	663 (43)	1087 (57.7)	343 (74)	295 (47)	638 (65.2)	563 (68.0)	530 (70.1)

†. Chinese noodles ††. a super express train in Japan ‡. a Japanese major league baseball player

Table 2: Results of second image gathering.

	num.	collected images			selected images				
keyword	URLs	A	B	A+B	A	B	A+B	word vec.	LSI
apple	12757	1141 (78)	2137 (59)	3278 (64.7)	744 (75)	751 (64)	1495 (69.3)	1313 (70.4)	1245 (70.9)
lion	15128	511 (87)	1548 (49)	2059 (66.5)	369 (86)	598 (51)	967 (71.5)	745 (75.1)	646 (76.7)
baby	14554	1833 (56)	1738 (53)	3571 (54.9)	1207 (52)	624 (61)	1831 (55.5)	1595 (57.3)	1500 (57.4)
notebook PC	9371	781 (57)	1756 (32)	2537 (44.1)	535 (57)	755 (34)	1290 (47.3)	1069 (50.6)	1033 (51.4)
ramen	10849	1945 (78)	4307 (55)	6252 (66.5)	1148 (75)	1852 (52)	3000 (64.8)	2935 (65.0)	2871 (65.3)
Shinkansen	10841	877 (67)	2272 (44)	3149 (55.1)	647 (66)	1129 (51)	1776 (58.9)	1266 (61.1)	1088 (64.3)
Kinkaku temple	9583	720 (59)	800 (25)	1520 (41.8)	518 (65)	253 (27)	771 (51.8)	689 (56.3)	668 (56.6)
Nomo	6288	107 (87)	232 (17)	339 (40.5)	92 (89)	59 (17)	151 (61.4)	125 (70.9)	119 (72.7)
average	11171	989 (71)	1848 (42)	2838 (54.3)	657 (71)	752 (45)	1410 (60.1)	1217 (63.3)	1146 (64.4)

words. They were "safari", "zoo", "park", "elephant", "tiger", "Africa", "group", "giraffe", "mane", and "head". Finally, we obtained 1171 images with a 64.4% precision on average for the LSI-based selection. Note that we gathered relatively fewer images of "Nomo" than images of other keywords, since "Nomo" is a person's name and fewer images of him exist on the web.

5. CONCLUSIONS

In this paper, we described a method, an implementation, and the experiments of an automatic image-gathering system for the web. We gathered more than one thousand images for one keyword and achieved the high precision of about 70% without any knowledge about target images by using word vectors and the LSI method.

In this implementation, we used simple image features for the image selection. In future work, we plan to exploit more sophisticated image features to improve the precision rate. Moreover, we will apply gathered images for our web image mining project.

6. REFERENCES

[1] V. N. Gudivada and V. V. Raghavan, "Content-based image retrieval-systems," *IEEE Computer*, vol. 28, no. 9, pp. 18–22, 1995.

[2] K. Yanai, "Image collector: An image-gathering system from the World-Wide Web employing keyword-based search engines," in *Proc. of IEEE Int'l Conf. on Multimedia and Expo*, 2001, pp. 704–707.

[3] K. Yanai, "An experiment on generic image classification using Web images," in *Proc. of the Third IEEE Pacific-Rim Conference on Multimedia (Springer LNCS no.2532)*, 2002, pp. 303–310.

[4] C. Framkel, M. J. Swain, and V. Athitsos, "WebSeer: An image search engine for the World Wide Web," Tech. Rep. TR-96-14, University of Chicago, 1996.

[5] J. R. Smith and S. F. Chang, "Visually searching the Web for content," *IEEE Multimedia*, vol. 4, no. 3, pp. 12–20, 1997.

[6] S. Sclaroff, M. LaCascia, et al., "Unifying textual and visual cues for content-based image retrieval on the World Wide Web," *CVIU*, vol. 75, no. 1/2, pp. 86–98, 1999.

[7] J. Hafner, H. S. Sawhney, et al., "Efficient color histogram indexing for quadratic form distance functions," *IEEE Trans. on PAMI*, vol. 17, no. 7, pp. 729–736, 1995.

[8] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[9] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society of Information Science*, vol. 41, no. 6, pp. 391–407, 1990.