Probabilistic Web Image Gathering

Keiji Yanai
Department of Computer Science,
The University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi,
Tokyo, 182-8585 JAPAN
yanai@cs.uec.ac.jp

Kobus Barnard
Computer Science Department,
University of Arizona
Tucson, AZ, 85721 USA
kobus@cs.arizona.edu

ABSTRACT

We propose a new method for automated large scale gathering of Web images relevant to specified concepts. Our main goal is to build a knowledge base associated with as many concepts as possible for large scale object recognition studies. A second goal is supporting the building of more accurate text-based indexes for Web images. In our method, good quality candidate sets of images for each keyword are gathered as a function of analysis of the surrounding HTML text. The gathered images are then segmented into regions, and a model for the probability distribution of regions for the concept is computed using an iterative algorithm based on the previous work on statistical image annotation. The learned model is then applied to identify which images are visually relevant to the concept implied by the keyword. Implicitly, which regions or the images are relevant is also determined. Our experiments reveal that the new method performs much better than Google Image Search and a simple method based on more standard content based image retrieval methods.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Design, Experimentation

Keywords

Web image mining, Web image search, image selection, probabilistic method

1. INTRODUCTION

Because of the recent growth of the World Wide Web, we can easily gather substantive quantities of image data. Our goal is to mine such data for visual content. In particular, we wish to build a large scale data set consisting of many highly relevant images for each of thousands of concepts. We present below a method for achieving substantively better relevance than using either text, or the combination of text and standard content-based image retrieval (CBIR) methods based on simple image descriptors.

In the case of text data, there are many studies about how to gather data and use it as "knowledge" effectively. While such text "Web mining" is a difficult endeavor and is an active research area, mining image data poses additional challenges and has seen less research activity. The problem with mining images for knowledge is that it is not known how to reliably automatically determine semantics from image data. This has been refereed to as the semantic gap. For this reason, commercial image search engines such as Google Image Search and Altavista Image Search rely on text associated with the images as determined from surrounding HTML data by a variety of heuristics. The approach supports fast retrieval, and is somewhat successful, partly because the user can select from a number of choices.

We contrast the search activity with mining for "visual knowledge" which is the topic of this paper. To mine the data we are willing to expend more resources to achieve a more precise result. The results can then be used to improve indexes for image search, but our main focus here is to build a training set for generic image recognition. This data set will go beyond what is available from commercial image collections such as Corel images which are represent only a subset of the kinds of images that we need to study. Web images are as diverse as real world scenes, since Web images are taken by a large number of people for various kinds of purpose. It can be expected that diverse training images enable us to classify/recognize diverse real world images. We believe that use of image data on the Web, namely visual knowledge on the Web, is promising and important for resolving real world image recognition.

The key improvement in our gather approach over earlier work [16, 17] is the construction of a probabilistic model for the relevant part of the images. A model for regions relevant to the keyword is learned simultaneously with one for irrelevant (background) regions. This irrelevant region model is also meant to absorb all regions from irrelevant images. This process substantively improves the accuracy of the set of relevant images. A large data set of images accurately classified into thousands of categories will provide a training data for computer vision methods to further automatically determine content based on visual information.

To obtain candidate labeled images we could use relevance feedback or as a semi-supervised was done by H. Feng et al.[6]. In their experiments, they selected 50 relevant images by hand for each search, and then used that for bootstrapping further gathering. However, to achieve the scale that we hope for, a fully automated method is desirable. Thus

MIR'05, November 10–11, 2005, Singapore.

we propose starting with images evaluated as highly relevant ones by analyzing associated HTML texts as training images. In our previous work [16], we revealed that images whose file name, ALT tag or link word includes a certain keyword "X" are relevant to the keyword "X" with around 75% precision on average. Although the images include 25% irrelevant images, and many of the remaining 75% are not a desired canonical example, they provide an adequate starting point for our approach. We then build a model of a visual concept associated to the keyword "X". We use a generative model based on the Gaussian mixture model (GMM) to represent "X" model, and estimate the model with the EM algorithm. Next, we recognize images evaluated as highly or medium relevant by analyzing associated HTML texts with the model, and select "X" images from them. By repeating this image selection and model estimation for several times, we can refine the "X" model and finally obtain "X" images with the high accuracy. In the experiments, we show that the new method performs much better than Google Image Search and the simple method based on CBIR (Content-Based Image Retrieval) we employed in the previous system [16].

The rest of this paper is organized as follows: In Section 2 we review related work. In Section 3 we overview our framework, and in Section 4 we describe the probabilistic framework to select images gathered from the Web. In Section 5 we presents the experimental results and evaluations, and in Section 6 we conclude this paper.

2. RELATED WORK

At present, some commercial image search engines on the Web such as Google Image Search, Ditto and AltaVista Image Search are available. Their accuracy of search results is, however, not always sufficient since they employs only keyword-based search. Therefore, to overcome such drawback, some integrated Web image search engines employing both keyword-based search and content-based image retrieval have been proposed. WebSeer [9], WebSEEk [14] and Image Rover [13] are representative systems employing both visual and textual information. These systems search for images based on query keywords, and then a user selects query images from search results. After this selection by the user, the systems search for images that are similar to the query images based on image features. These three systems carry out their search in an interactive manner. They can be regarded as the combination of text-based Web image search and CBIR. An interactive way is suitable for "search" not for "gathering", since it needs human's intervention during the process.

Furthermore, the three research systems quoted above require crawling over the Web in advance for gathering Web images and making large indices of images on the Web. Hence, they require a large-scale web-crawling mechanism for the whole Web and continuous web-crawling to keep their indices up-to-date for practical use. However, they limited crawled Web sites in their experiments, and did not make large indices covering the whole Web. This shows difficulty to make these system more practical like Google image search. Therefore, using the commercial Web search engines which have been indexing whole the World Wide Web such as Google as the index of the Web is much more practical approach. In [16], we proposed using the Web text search engines to gather images on the Web. In contrast to the existing systems, due to exploiting of existing keyword-based search engines and on-demand image-gathering, our system, the Image Collector, did not require a large-scale web-crawling mechanism and making a large index in advance, so that it can be used practically unlike the existing Web image search system quoted above. In fact, at that time, Google Image Search has not got in service yet, so it was a very good way to search whole the World Wide Web for images. Now we can use Google Image Search or other commercial image search engines on the Web. So recently most of studies related to Web image search focus on how to refine results of Web image search engines, which use only textual information as clues to index Web images but index most part of the extremely huge World Wide Web.

As one of such studies, H. Feng et al. proposed a new method to refine results of Web image search engines [6], which employed a co-learning method [2] that is a kind of semi-supervised learning. They used both image features and word vectors extracted from associated HTML documents. In their paper, they claimed only 50 relevant images were needed to be supervised by hand in case of using co-learning with both visual and textual features. They called their method a bootstrapping approach. In their experiment, they used 5418 images associated to 15 concepts, and obtained an F-measure of 54%.

X. Song et al.[15] proposed building automatic visual model from the result of Google image search without human's intervention. They employed multiple instance learning[10] as an image learning method. It seems to be similar to our framework, "Web image mining" in terms of learning from Web images with no supervision. However, they focused on only frontal face images and used a face detector which was learned in advance, although they claimed that they can apply their proposed method to generic Web images by using the supervised framework they proposed before. Since they used a face detector for which they used many labeled face images to learn a face model, we can consider that their method is not truly "automatic" and it is a kind of supervised methods. Their method can build face models of famous individuals, after detecting face regions by the face detector. In their paper, they did not mention how many images they obtained from Google Image Search for each person.

The two systems mentioned above also aims to "search" the Web for images. On the other hand, the objective of our image gathering is absolutely different from ones of the other Web image search systems including commercial Web image search engines. Their objective is searching for highly relevant but relatively a small number of images. That is why they have adopted interactive or supervision-needed ways. Unlike these system, we aim to gather a large number of relevant images, and actually we plan to gather enormous and various kinds images associated to more than one thousand kinds of concepts. In such case, even a very little interaction should be avoided, since we have to repeat the same kinds of interaction more than one thousand time. It must be a troublesome job. Therefore, we adopt non-interactive search without user intervention during the gathering process. This enables the gathering process to be done as a batch job. To gather many kinds of images from the Web, what we have to do are just providing keywords related to concepts of images we like to gather and waiting.

In work possibly most similar to ours, P. Fergus et al. applied their probabilistic method [7] to filtering of results of Google Image Search [8]. Their method can model object categories in an unsupervised way. Unlike the two above studies, they used imperfect training data which includes outliers, and in addition they used negative training data which consists of only irrelevant images. Although it is similar to our method in terms of not requiring supervision and using negative training images, there are some important

differences: (1) They used all images obtained from Google Image Search as training images, and HTML analysis never affected image analysis process, while we use only images judged as highly relevant ones by HTML analysis, yielding better training data. (2) They used image patches or image curves as units to be modeled, while we use image regions generated by a region segmentation algorithm as the original modeling algorithms that the respective methods are based on [7, 1] are different.

3. OVERVIEW OF IMAGE GATHERING

Our proposed method consists of two stages, which are a collection stage and a selection stage (Figure.1). Most of recent studies related to Web image search focus on only how to select relevant images from results of Web image search engines. In contrast, we make much of a collection stage as well as a selection stage. We have indicated that we can collect images related to given keywords from the Web with only Web *text* search engine and we do not need Web image search engines to do that [16]. Therefore, in this paper we also use Web text search engine to fetch images from the Web, although we can use Web image search engines instead of Web text search engines in our framework.

First of all, we provide keywords which represent the visual concept of images we like to obtain. For example, "lion", "dog" and "cat". If we use polysemous words as keywords, we can add subsidiary keywords which restrict the meaning of the main keyword. For example, when we like to obtain images of "bank" of a river, we should use "river" as a subsidiary keyword in addition to the main keyword "bank".

In the collection stage, we use the method we proposed before [16, 17] to collect images from the Web. Since an image on the Web is usually embedded in an HTML document that explains its content, we exploit some existing commercial *text* Web search engines and gather URLs of HTML documents related to the keywords. In the next step, using those gathered URLs, we fetch HTML documents from the Web, and evaluate the relevancy of images only by analyzing associated HTML documents. If it is judged that images are related to keywords, the image files are downloaded from the Web. According to the relevancy of images to the given keywords, we divide fetched images into two groups: images in group A are highly relevant to the keywords, and others are classified into group B. For all gathered images, we perform region segmentation by JSEG[4] and extract image features from each region. Moreover, we extract word vectors from the HTML documents associated to all the downloaded images. The detail is described in [16, 17].

In the selection stage, we employ a probabilistic method to select relevant images from all the downloaded images. In general, to use a probabilistic method or other machine learning methods to select true images, we need labeled training images. However, we do not want to pick up good images by hand. Instead, we regard images classified into group A as training images, although they always include some irrelevant images. In our probabilistic framework, we allow training data to include some irrelevant data and we can remove them by repeating both estimation of a model and selection of relevant regions of images from all the regions of images in group A and group B. We use a generative model based on the Gaussian mixture model to represent models associated to keywords, and estimate models with the EM algorithm. After estimating the model, we "recognize" relevant region out of all regions in group A and B with the model. We repeat this model estimation and region selection. After the second iteration, we use regions selected in the previous iteration as training data for estimating a model.

4. SELECTION STAGE

In the selection stage, the system selects relevant images to the concept which the keywords represent out of all the downloaded images in the collection stage.

4.1 Overview of Probabilistic Approach for Image Selection

As a method to select images, we adopt a probabilistic method with a Gaussian mixture model. This approach is based on our method for learning to label image regions from images with associated text without the correspondence between words and images regions [5, 1]. That method uses a mixture of multi-modal components, each combining a multinomial for words and a Gaussian over image features. Here, we simplify things a bit, and build models of the distribution of image features for a given concept for regions which are obtained by a region segmentation algorithm.

To get a model of regions associated to a certain concept, we need training images. As mentioned before, our basic policy is no human intervention, so that we propose using images in group A as training images. Most of images in group A are relevant ones, but they always includes outliers due to no supervision. Moreover, in general, images usually include backgrounds as well as objects associated with the given concept. Therefore, we need to eliminate outlier images and regions unrelated to the concept such as backgrounds, and pick up only the regions strongly associated with the concept in order to make a model correctly. We use only the regions expected to be highly related to the concept to estimate a model. In our new method, we need negative training images in addition to group A and B images. We prepare about one thousand images by fetching them from the Web randomly as negative training images in advance.

Our method to find regions related to a certain concept is an iterative algorithm similar to the expectation maximization (EM) algorithm applied to missing value problems. Initially, we do not know which region is associated with a concept "X", since an image with an "X" label just means the image contain "X" regions. In fact, with the images gathered from the Web, even an image with an "X" label sometimes contains no "X" regions at all. So at first we have to find regions which are likely associated with "X". To find "X" regions, we also need a model for "X" regions. Here we adopt a probabilistic generative model, namely a mixture of Gaussian, fitted using the EM algorithm.

In short, we need to know a model for "X" regions and which regions are associated with "X" simultaneously. However, each one depends on each other, so we proceed iteratively. Once we know which regions corresponds to "X", we can regard images containing "X" regions as "X" images, and therefore we can compute the probability of an "X" image for each image. Finally, we select the images which have the high probability as final results.

In addition to image selection with only image features, we also propose the extended method using not only image features but also textual features extracted from associated HTML documents. In fact, some existing studies have integrated both visual and textual features [13, 16, 6]. To realize that, we compute the probability of "X" and "non-X" in terms of word vectors in the same way as image features, and integrate them.

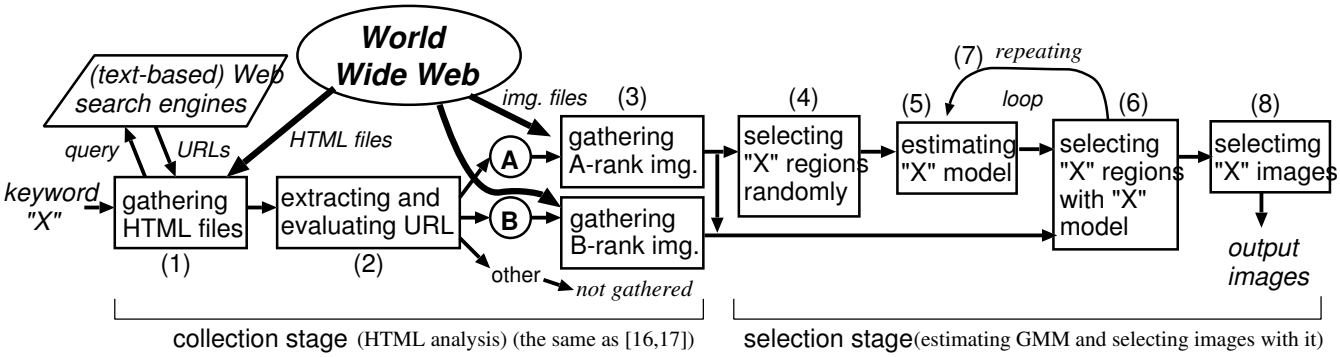

Figure 1: Processing flow of image-gathering from the Web employing a probabilistic method, which consists of the collection stage and the selection stage.

4.2 Segmentation and Image Feature Extraction

For the images gathered from the Web as "X" images, we carry out the region segmentation. In the experiment, we use JSEG [4]. After segmentation, we extract image features from each region whose size is larger than a certain threshold. As image features, we prepare three kinds of features: color, texture and shape features, which include the average RGB value and its variance, the average response to the difference of 4 different combination of 2 Gaussian filters, region size, location, the first moment and the area divided by the square of the outer boundary length. An image feature vector we use in this paper is totally 24-dimension.

4.3 Textual Feature Extraction

In addition to image selection based on only image features, we make experiments with not only image features but also textual features, namely word vectors which are extracted from associated HTML documents.

To make a word vector for each HTML document, we eliminates HTML tags and extracts surrounding ten words (only nouns, adjectives, and verbs) before and after the link tag to the image file, link words and words in the ALT tag from HTML documents associated with downloaded images. We count the frequency of appearance of all the extracted words, select the top 300 words in terms of the frequency, and make a 300-dimensional word vector whose elements are word frequencies weighted by TFIDF (Term Frequency and Inverse Document Frequency) [12] for each of the images. Moreover, to shorten the vectors and to equate words having similar meanings, we perform the LSI methods (Latent Semantic Indexing) [3] to the 300-dimensional word vectors. These methods compress word vectors with singular value decomposition which is similar to the principal component analysis. We compress a 300-dimensional word vector into a 100-dimensional vector, and treat with this 100-dimensional vector as a word vector in this paper.

4.4 Detecting Regions and Estimating a Model Associated with "X" and "non-X"

To obtain $P(X|r_i)$, which represents the probability of how much the region is associated with the concept "X", and some parameters of the Gaussian mixture model, which represents a generative model of "X" regions, at the same time, we propose an iterative algorithm. Note that "X" corresponds to a certain concept associated to given keywords.

At the initial iteration, we regard images in group A which are expected to be highly relevant to the concept "X" by HTML analysis as positive training images, and prepare negative training images by gathering images from the Web in advance. To gather negative training images, we provided Google Image Search with randomly selected 200 adjective keywords which have no relation to noun concepts, and collected 4000 negative training images.

Next, we select n "X" regions randomly from group A images, and select n "non-X" regions randomly from regions which come from negative training images, respectively. In the experiment, we set n as 1000.

Taking positive and negative regions together, we apply the EM algorithm, which is a kind of a probabilistic clustering algorithm, to $2n$ image feature vectors of the regions selected from positive and negative initial training images, and obtain the Gaussian mixture model.

To select positive components and negative components from all components of the mixture model, we compute $P_0(c_j|X)$ which represents the ratio that the j-th component of the mixture model, c_j, contributes to the concept "X" within the obtained GMM, according to the following formula:

$$P_0(c_j|X) = (1/n_X)\sum_{i=1}^{n_X} P(c_j|r_i^X, X) \quad (1)$$

$$= (\alpha/n_X)\sum_{i=1}^{n_X} P(X|c_j, r_i^X)P(c_j) \quad (2)$$

where n_X is the number of positive regions, r_i^X is the i-th "X" region, and α is a constant for the normalization. As the same way, we also compute $P_0(c_j|nonX)$. Here we regard all the regions selected from positive images as "X" regions and all the regions selected from negative images as "non-X" regions, and substitute their feature vectors to the above formula.

Next, we compute p_j^X and p_j^{nonX} for all components j as follows:

$$p_j^X = \frac{P_0(c_j|X)}{P_0(c_j|X) + P_0(c_j|nonX)} \quad (3)$$

$$p_j^{nonX} = \frac{P_0(c_j|nonX)}{P_0(c_j|X) + P_0(c_j|nonX)} \quad (4)$$

We select components where $p_j^X > th_1$ as positive components and components where $p_j^{nonX} > th_1$ as negative components. Positive components and negative components means Gaussian components associated with the concept "X" and Gaussian components strictly not to associated with "X", respectively. The key point in this component selection process is that mixing positive samples and negative samples together before applying the EM, and throwing

away neutral components which belongs to neither positive nor negative components, since neutral components are expected to be associated with image features included in both positive and negative samples and to be useless for discrimination between "X" and "non-X". This is different from other work (e.g.[11]) which estimates two GMMs separately with EM to model positive and negative image concepts.

We regard the mixture of only positive components as an "X" model and the mixture of only negative components as a "non-X" model as. With these models of "X" and "non-X", we can compute $P(X|r_i)$ and $P(nonX|r_i)$ for all the regions extracted from images in both group A and group B. First, we compute $p1(X|r_i)$ which is the output of the model of "X' and $p2(nonX|r_i)$ which is the output of the model of "non-X" for each region r_i:

$$p1(X|r_i) = \sum_{k=1}^{m_1} w_{1,k} \frac{1}{\sqrt{(2\pi)^N |\Sigma_{1,k}|}} \exp^{-\frac{1}{2}(x_i-\mu_{1,k})^T \Sigma_{1.k}^{-1}(x_i-\mu_{1,k})} \quad (5)$$

$$p2(nonX|r_i) = \sum_{k=1}^{m_2} w_{2,k} \frac{1}{\sqrt{(2\pi)^N |\Sigma_{2,k}|}} \exp^{-\frac{1}{2}(x_i-\mu_{1,k})^T \Sigma_{2,k}^{-1}(x_i-\mu_{2,k})} \quad (6)$$

where x_i is the image feature vector of region r_i, N is the dimension of image features, m_1 is the number of positive components and $w_{1,k}$, $\mu_{1,k}$ and $\Sigma_{1,k}$ represent the weight, the mean vector and the covariant matrix of k-th positive component, respectively, on the condition of $\sum w_{1,k}{}_{k=1}^{m/1} = 1$. Finally, we obtain $P(X|r_i)$ and $P(nonX|r_i)$ as follows:

$$P(X|r_i) = \frac{p1(X|r_i)}{p1(X|r_i) + p2(nonX|r_i)} \quad (7)$$

$$P(nonX|r_i) = \frac{p2(nonX|r_i)}{p1(X|r_i) + p2(nonX|r_i)} \quad (8)$$

For the next iteration, we select the top n regions regarding $P(X|r_i)$ as "X" regions and the top $\frac{2}{3}n$ regions regarding $P(nonX|r_i)$ as "non-X" regions. In addition, we add $\frac{1}{3}n$ regions randomly selected from negative images gathered from the Web in advance to the "non-X" regions. We repeat the processing described above with n positive regions and n negative regions for several times.

4.5 Computing the Probability of "X"

After iterating both detection of regions and estimation of models for several times, in order to decide which images are "X", finally, we compute the probability of "X" $P(X|I_j)$, where I_j represents j-th image.

To estimate $P(X|I_j)$, we select top T regions in terms of $P(X|r_i), (r_i \in I_j)$, and average them as follows:

$$P(X|I_j) = \frac{1}{T} \sum_{k=1}^{T} P(X|r^i_{top_k}) \quad (9)$$

where $r^i_{top_k}$ is the j-th largest region within image I_i in terms of $P(X|r_i)$. This estimation of $P(X|I_j)$ is based on the heuristic that an image having regions whose $P(X|r_i)$ are high is likely to be an "X" image. Since images usually includes backgrounds as well as target objects, background regions or unrelated regions should be ignored for estimating $P(X|I_j)$. Therefore, we use not all regions but only several regions with high probability. Finally, we select images whose $P(X|I_j)$ is more than a threshold th_2 as final output images. In the experiment, we set T as 2.

4.6 Selection by Textual Feature

To use textual features in addition to image features, we compute the probability of "X" and "non-X" in terms of word vectors in the same way as image features, and integrate them in two kinds of ways. Note that a word vector corresponds with an image, while an image feature vector corresponds with a region.

After selecting "X" regions, we need to select "non-X" regions, and need to compute the probability of "X" images and "non-X" images for each image, respectively, after each iteration of image selection by image feature. Since we do not have negative samples of word vectors unlike the case of image features in the beginning, we make use of results of selection by image features. We regard the word vectors of the images labeled as "X" by image-feature-selection as positive training samples, and regard the word vectors of the images labeled as "non-X" as negative training samples.

In the next step, using both positive and negative training vectors, we build an "X" model and a "non-X" model in the similar way as image features. We perform the EM algorithm for the word vectors of "X" and "non-X" images, select Gaussian components corresponding to "X" and "non-X", and compute the probability of "X" and "non-X" in terms of word vectors.

We prepare two methods to use the probability $P_{word}(X|I_j)$. One is computing the weighted sum $P_{total}(X|r_i)$ of word-vector-based probability $P_{word}(X|I_j)$ and image-feature-based probability $P(X|r_i)$, and the other is the two step selection that word-vector-based selection is carried out after image-feature-based selection.

In the weighted-sum selection, we compute $P_{total}(X|r_i)$ for all r_i as follows:

$$P_{total}(X|r_i) = wP_{word}(X|I_j) + (1-w)P(X|r_i) \quad (10)$$

where $r_i \in I_j$, and select positive and negative training regions for the next iteration using $P_{total}(X|r_i)$ instead of just $P(X|r_i)$. In the experiment, we set w as 0.25.

In the two-step selection, using a threshold th_3, we eliminate region r_i whose $P_{word}(X|I_j)$ is low, where $r_i \in I_j$ from the positive regions after image-feature-based region selection.

4.7 Proposed Algorithm

To summarize our method we described above, the algorithm is as follows:

(1) Carry out region segmentation for all the images and extract image features from each region of each image. In this paper, we use JSEG[4] to perform region segmentation. In case of using word vector, we also generate word vectors for all the images in both group A and B from associated HTML documents in this step.

(2) At the first iteration, regard images in group A as positive training images which are associated with the concept "X" and images gathered from the Web with non-noun keywords in advance as negative training images.

(3) Select n "X" regions randomly from positive images, and select n "non-X" regions randomly from negative images, respectively (Figure 1 (4)).

(4) Applying the EM algorithm to the image features of regions which are selected as both positive and negative regions, compute the Gaussian mixture model for the distribution of both "X" and "non-X" (Figure 1 (5)).

(5) Find the components of the Gaussian mixture which contributes "X" regions or "non-X" regions greatly. They are regarded as "X" components or "non-X" components, and the rest are ignored. The mixture of only

"X" regions is a model of "X" regions, and the mixture of only "non-X" is a model of "non-X" regions.

(6) Based on the mixture of "X" components and the mixture of "non-X" components, compute $P(X|r_i)$ and $P(nonX|r_i)$ for all the regions which come from "X" images, where r_i is the i-th region.

(7) Select the top n regions in terms of $P(X|r_i)$ as new positive regions and the top $\frac{1}{3}n$ regions in terms of $P(nonX|r_i)$ as new negative regions. Add $\frac{1}{3}n$ regions randomly selected from the negative training images to new negative regions.

(8) Repeat from (4) to (7) with newly selected positive and negative regions (Figure 1 (7)).

(9) After repeating several times, compute $P(X|I_j)$ for each image i based on $P(X|r_i)$ and select images whose $P(X|I_j)$ is larger than th_2 as final output images (Figure 1 (8)).

When using word vectors, the following steps are carried out instead of Step (8).

(8-1) Compute $P(X|I_j)$ and $P(nonX|I_j)$ for each image i based on $P(X|r_i)$ and $P(nonX|r_i)$, respectively, and select images whose $P(X|I_j)$ or $P(nonX|I_j)$ is larger than th_2 as "X" or "non-X" images.

(8-2) Applying the EM algorithm to the word vectors associated to the images selected as both "X" and "non-X", compute the Gaussian mixture model for the word vectors of both "X" and "non-X".

(8-3) Find the components of the Gaussian mixture which contributes "X" images or "non-X" images greatly in terms of word vectors, and regard the mixtures of selected components for "X" and "non-X" as the model of "X" and "non-X" images.

(8-4) Compute $P_{word}(X|I_j)$ for all images in group A and group B.

(8-5) We have two strategies to use $P_{word}(X|I_j)$:

(1) Compute $P_{total}(X|r_i)$ for all r_i as follows:

$$P_{total}(X|r_i) = wP_{word}(X|I_j) + (1-w)P(X|r_i) \quad (11)$$

where $r_i \in I_j$, and re-execute Step (7) using $P_{total}(X|r_i)$ instead of using $P(X|r_i)$.

(2) Using a threshold th_3, eliminate region r_i where $P_{word}(X|I_j) < th_3$, $r_i \in I_j$ from the positive images selected by image features in Step (7).

(8-6) Repeat from (4) to (8-5) for several times.

In the following subsection, we explain the details of each step of the algorithm described above.

5. EXPERIMENTAL RESULTS

We made experiments for the following ten concepts independently: beach, sunset, flower, waterfall, mountain, lion, apple, baby, note-PC, and Chinese noodle. For only "lion" and "apple", we added subsidiary keywords "animal" and "fruit" to restrict its meaning to "lion of animal" and "apple of fruit" in the collection stage, respectively.

In the collection stage, we gathered around 5000 URLs for each concept from Google Search and Yahoo Web Search which are not "image search" but "text search". The exact numbers vary depending on concepts, since we excluded duplicate URLs from the URL list for each category.

Table 1 shows the results of the collection stage, namely raw images, and we added to it the evaluation of the results of Google Image Search and our previous system which employs the CBIR-based image selection method [16] for comparison. The results of the collection stage consists of the number of images downloaded from the Web with only HTML analysis and their precision. To compute the precision and the recall, we randomly selected 500 images from the images of each concepts and checked their relevancy by the subjective evaluation. Note that for the downloaded images we cannot estimate the recall, since the denominator to estimate it corresponds to the number of images associated to the given concept on whole the Web and we cannot get to know it. Regarding the results of Google Image Search, we show the precision of output images ranked between 1 and 500 in the table. The average precision of raw images, 62.2%, was slightly superior to the average precision of top 500 results of Google images, 58.6%, while we collected about 3000 images a concept. This shows that our original image collection method is better than Google Image Search.

In the selection stage, we selected images with high probability of "X" from raw images collected in the collection stage. In the experiment, we set th_1, th_2 and th_3 as 0.85, 0.5 and 0.2, respectively, which are decided from the results of the preliminary experiments. We define the number of Gaussian components for image feature and the number of Gaussian components for word vectors as m and n. Here, we set n as 30. Regarding m, we describe how we decided the value for the experiment.

We show the average precision (lines in the figure) and average number (bars in the figure) of final output images in case of m=50, 100, 150, 200 and 300 in Figure 2. This indicates that m should be more than 150, since the precision is the best on the average in case of $m = 150$, so that we decided to set m as 150. Regarding the number of iteration, we selected 2 as the iteration times, because there was not big changes after the second iteration in terms of the number and the precision of selected images, while the number of images increased slightly between the first and the second iteration.

Table 2 shows the number, the precision and the recall of the results by the three proposed method, which are the method by only image feature, by image and word features with linear combination of both probability and by image and word features with two-step selection. Here, we describe these three methods as Only-image, Word(1) and Word(2), and describe the raw images and the CBIR-based method as RAW and CBIR.

Regarding the average results, Word(1) outperformed the results of CBIR perfectly in terms of both precision and recall, and Google Image Search in terms of precision. The 72.0% precision and the 59.1% recall were obtained with Word(1), while the 66.0% precision and the 57.7% recall were obtained with conventional CBIR-based methods we proposed in [16] and the 58.6% precision was obtained by Google Image Search. In addition, the result of Work(1) also outperforms 54% F-measure by H. Feng et al.[6], and the 65.9% precision and the 15% recall by P. Fergus et al.[8], although the keywords used in the experiments are different each other and it is difficult to compare their results.

The difference between the results of Only-image and Word (1) was small, so that the use of word vector is not indispensable but helpful to improve the results. Regarding Word(2), the recall was not good, since two-step selection always reduces the number of output images and in this case it was reduced too much.

For each results in case of Word(1), "sunset" and "mountain" were really good in the precision, which are 88.6% and 86.7since the precision of the raw images of "mountain" was high and the model was effective for "sunset", most of the scenes of which are relatively simple.

Figure 3 shows the number of images and the precision

Table 1: Results of the collection stage and the selection stage by the CBIR-based method [16]. This table describes the precision of the 500 output images of Google Image Search which are ranked from 1 to 500, the number of raw images collected from the Web, the number of selected images out of them by the existing CBIR-based method. Numerical values in () represent the precision and the recall.

concepts	Google prec.	raw images A	raw images B	raw images A+B	selected images by the CBIR-based method A	selected images by the CBIR-based method B	selected images by the CBIR-based method A+B
sunset	79.8	790 (67)	710 (44)	1500 (55.3)	644 (72, 88)	184 (26, 15)	828 (62.2, 62.1)
mountain	48.8	1950 (88)	3887 (71)	5837 (79.2)	1750 (89, 91)	1673 (70, 42)	3423 (82.6, 61.2)
Chinese noodle	65.2	901 (78)	1695 (55)	2596 (66.6)	699 (78, 78)	793 (59, 51)	1492 (71.0, 61.3)
waterfall	72.4	2065 (71)	2584 (70)	4649 (70.3)	1884 (70, 90)	1397 (75, 58)	3281 (71.4, 71.7)
beach	63.2	768 (69)	1155 (62)	1923 (65.5)	685 (70, 90)	443 (61, 38)	1128 (67.3, 60.3)
flower	65.6	576 (72)	1418 (67)	1994 (69.6)	487 (81, 95)	465 (76, 37)	952 (79.3, 54.4)
lion	44.0	511 (87)	1548 (49)	2059 (66.0)	369 (86, 71)	598 (51, 40)	967 (71.0, 50.5)
apple	47.6	1141 (78)	2137 (59)	3278 (64.3)	744 (75, 63)	751 (64, 38)	1495 (68.8, 48.8)
baby	39.4	1833 (56)	1738 (53)	3571 (54.5)	1207 (52, 62)	624 (61, 41)	1831 (55.1, 51.8)
notebook PC	60.2	781 (57)	1756 (32)	2537 (43.6)	535 (57, 69)	755 (34, 47)	1290 (46.9, 54.6)
TOTAL/AVG.	58.6	11316 (72)	18628 (56)	29944 (62.2)	9004 (73, 80)	7683 (58, 41)	16687 (66.0, 57.7)

Table 2: Results of the selection stage by the image-feature-based method and by the selection method integrating image feature and word vector in two kinds of ways. This table describes the number of selected images from group A, group B, and all the downloaded images by the image-feature-based method, the number of images selected by the image-and-word method with the linear combination, and the number of images selected by the image-and-word method with the two-step filtering. Numerical values in () represent the precision and the recall.

concepts	image selection by only image feature A	image selection by only image feature B	image selection by only image feature A+B	w/ word vector(1) A+B	w/ word vector(2) A+B
sunset	387 (96, 72)	249 (83, 68)	**636 (91.0, 70.2)**	**697 (88.6, 75.6)**	433 (93.9, 49.5)
mountain	1237 (93, 65)	2273 (85, 65)	**3510 (89.0, 65.0)**	**3504 (86.7, 61.2)**	2880 (86.4, 50.8)
Chinese noodle	453 (85, 49)	813 (69, 59)	**1266 (77.0, 53.2)**	**1339 (74.8, 56.2)**	817 (76.0, 33.8)
waterfall	1569 (77, 73)	1935 (76, 77)	**3504 (76.8, 74.6)**	**3406 (78.6, 69.8)**	2345 (76.8, 40.5)
beach	414 (74, 61)	569 (73, 64)	**983 (73.3, 62.5)**	**1163 (73.5, 75.8)**	583 (75.9, 42.1)
flower	282 (77, 50)	476 (66, 33)	**758 (71.9, 41.0)**	**889 (73.2, 50.6)**	427 (73.2, 24.6)
lion	234 (88, 59)	477 (55, 48)	**711 (69.4, 53.6)**	**846 (66.7, 65.5)**	487 (70.3, 38.8)
apple	499 (79, 45)	753 (60, 33)	**1252 (67.2, 37.7)**	**1362 (65.7, 39.5)**	691 (62.9, 18.4)
baby	801 (66, 49)	537 (60, 41)	**1338 (63.9, 45.9)**	**1619 (62.4, 50.1)**	983 (66.4, 34.8)
notebook PC	295 (68, 46)	572 (44, 50)	**867 (56.0, 47.6)**	**954 (49.8, 46.5)**	437 (59.1, 25.7)
TOTAL/AVG.	6171 (80, 57)	8654 (67, 54)	**14825 (73.5, 55.1)**	**15779 (72.0, 59.1)**	10083 (74.1, 35.9)

of the results of RAW, CBIR, Only-image, Word(1) and Word(2). The lines of the precision of Only-image and Word(1) have the same tendency, and they are almost overlapped, while the lines of RAW and CBIR are similar. In case of "sunset", the difference between RAW and Only-image are large, so that this shows that the proposed model matched "sunset" images very well.

We show some images selected as output images in Figure 4 and 5. Each image has a label on the center of the region whose $P(X|r_i)$ is the largest within the image. For "sunset" and "waterfall" we obtained good results, since most of their images are characteristic and relatively similar to each other. For example, "sunset" images consist of orange sky and black silhouette, and "mountain" images consist of white, gray or blue sky and green or brown ground. Therefore, the proposed model based on the Gaussian mixture could represent their image features well, and we achieved the high precision for them.

On the other hand, "lion" and "notebook PC" were difficult to classify with the proposed method. "Lion" images have many kinds of poses or scales of lions and are sometimes hard to be segmented, while "notebook PC" are sometimes placed on a messy office desk and segmentation tends to fail. In short, "objects" are not easy to be distinguished by the proposed method, while the proposed model can represent "scenes" well.

We have prepared the Web site to show the experimental results we provided in this paper. The URL is as follows:
`http://img.cs.uec.ac.jp/mm05/`

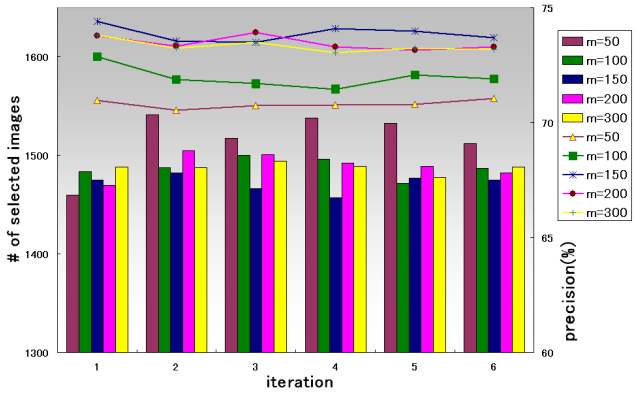

Figure 2: The average precision and average number of final output images in case that m=50, 100, 150, 200 and 300.

6. CONCLUSIONS

In this paper, we described a new probabilistic method for the selection stage of Web image-gathering and introduced it into the Web image gathering system we proposed before. In the proposed method, we employed a Gaussian mixture model to model regions of images in terms of image features and estimated it by the EM algorithm. It enabled us to learn from imperfect training data which comes from the Web without human supervision. In addition, we also build a model in terms of word vectors in the same way as image

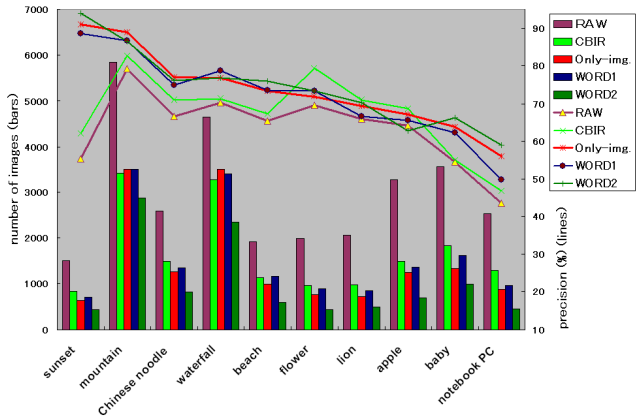

Figure 3: The precision and number of raw images (RAW) and results by the CBIR-based method (CBIR), by the proposed method with only image feature (Only-image), with image and word feature in the linear combination (WORD(1)) and with image and word in the two-step filtering (WORD(2)).

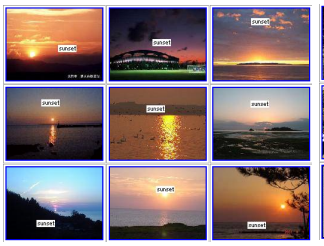

Figure 4: "Sunset".

Figure 5: "Waterfall".

features, and used both image features and word vectors together to select images related to the given concept from the raw images downloaded from the Web. In the experiments, we obtained the 73.5% precision and the 55.1% recall with only image features, and the 72.0% precision and the 59.1% recall with image and word features for ten concepts, which outperformed the results of the previous system and Google Image Search much.

The contribution of this paper is as follows:

(1) We divide raw images collected from the Web into two groups, A and B by analyzing associated HTML documents, and use images in group A which are more likely to be relevant as initial training images for the probabilistic learning method. We realized unsupervised learning in Web image-gathering without human intervention.

(2) By an iterative algorithm, we estimate a model for "X" regions and which regions are associated with "X", simultaneously. Finally, we regard the images containing regions having high probability as "X" images without summing the probability of all the regions coming from the image. In short, we proposed classifying images with only "X" regions ignoring background and other unrelated regions.

(3) We propose integrated use of word vectors with image feature vectors within a probabilistic Web image selection framework.

As future work, we plan to apply the proposed method to automatic generation of real world image corpus for generic image classification/recognition. In our method, we have already made generative models based on the Gaussian mixture which can be applied to generic image recognition. As a first step of "*Web Image Mining*" which is a new concept to make use of Web images for generic image classification/recognition [17], we are going to gather many kinds of images and build models for many kinds of concepts.

The experiments revealed that the model was effective for "scene" concepts such as sunset and mountain, but not enough for "object" concepts such as lion and notebook PC. So we need to develop new models which can represent "object" concepts as well as "scene" concepts by importing latest methods for generic object recognition.

7. REFERENCES

[1] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[2] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. of the eleventh annual conference on Computational Learning Theory*, pages 92–100, 1998.

[3] 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*, 41(6):391–407, 1990.

[4] Y. Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):800–810, 2001.

[5] P. Duygulu, K. Barnard, J. d. Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *The Seventh European Conference on Computer Vision*, pages IV:97–112, 2002.

[6] H. Feng, R. Shi, and T. Chua. A bootstrapping framework for annotating and retrieving WWW images. In *Proc. of ACM International Conference Multimedia 2004*, pages 960–967, 2004.

[7] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. of IEEE Computer Vision and Pattern Recognition*, volume 2, pages 264–271, 2003.

[8] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *Proc. of European Conference on Computer Vision*, pages 242–255, 2004.

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

[10] O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classification. In *Proc. of 15th International Conference on Machine Learning*, pages 341–349, 1998.

[11] M. R. Naphade, S. Basu, J. R. Smith, C. Y. Lin, and B. Tseng. Modeling semantic concepts to support query by keywords in video. In *Proc. of IEEE Intl. Conference on Image Processing*, pages I–145–148, 2002.

[12] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[13] S. Sclaroff, M. LaCascia, S. Sethi, and L. Taycher. Unifying textual and visual cues for content-based image retrieval on the World Wide Web. *Computer Vision and Image Understanding*, 75(1/2):86–98, 1999.

[14] J. R. Smith and S. F. Chang. Visually searching the Web for content. *IEEE Multimedia*, 4(3):12–20, 1997.

[15] X. Song, C. Lin, and M. Sun. Autonomous visual model building based on image crawling through internet search engines. In *Proc. of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 315–322, 2004.

[16] K. Yanai. Image collector: An image-gathering system from the World-Wide Web employing keyword-based search engines. In *Proc. of IEEE International Conference on Multimedia and Expo*, pages 704–707, 2001.

[17] K. Yanai. Generic image classification using visual knowledge on the web. In *Proc. of ACM International Conference Multimedia 2003*, pages pp.67–76, 2003.