Image Retrieval with Generative Model for Typicality

Taro Tezuka and Akira Maeda
College of Information Science and Engineering, Ritsumeikan University, Kusatsu, Japan
E-mail: {tezuka, amaeda}@media.ritsumei.ac.jp

Abstract—One of the most common image retrieval tasks is to find the most typical image that depicts the object specified by a query. Existing image search engines cannot efficiently do this since their search results are often a mixture of images belonging to various semantic concepts. Therefore, we introduce a probabilistic model for typicality. Our model consists of images, symbolic features, and latent semantic concepts (aspects). The aspect with highest probability is assumed to represent typicality. By collecting a large number of images, we can estimate parameters using the EM algorithm. The estimated parameters are used to quantify the level of typicality for each image. Based on the proposed method, we have implemented a system for ranking images by their typicality. Experiments using both artificial and real data showed the effectiveness of our method.

Index Terms—image retrieval, typicality, bag-of-features, generative model, aspects

I. INTRODUCTION

Advances in database and network technology have made the World Wide Web an image repository of enormous size. To fully utilize such a large data source, an effective method of image ranking is required. In image retrieval, one of the most common tasks is to find a typical image that depicts the object specified by a query. Some of the applications of retrieving typical images are as follows.

- The user wants to know the look of an object. For example, the user heard of a tropical fruit “naranjilla” and wants to know what it looks like.
- The user is driving a car and looking for a building. The car navigation system tells the location of the building but not its visual feature, so it is difficult to find the building unless its image is provided.
- The user is preparing presentation slides and wants to put an image or illustration best suited for the topic he will talk about. For example, the user is making a slide on “economic crisis” and wants to find an image representing such concept.

Existing image search engines are often not efficient for these tasks, since their search results are mixtures of images belonging to various semantic concepts. Many of the high-ranked results of image search engines are not typical. We present and evaluate a method that extracts typical images from the results of a web image search by applying a generative model, a type of probabilistic model. Although typicality is a difficult concept to capture, in this paper we define it as follows.

Definition: An image \( I \) is a typical image for a query \( Q \) if \( Q \) is an appropriate label for \( I \), given that the evaluator has enough knowledge on the object referred to by \( Q \).

We aim to provide a method that can efficiently retrieve a set of images that are appropriate under this criterion. Our proposed method estimates “aspects” expressed in a set of images and selects an aspect assumed to express typicality. We then rank the images using conditional probability. One characteristic of our method is that it expresses typicality using discrete probabilistic variables. Many models for classification and dimension reduction use continuous variables, including k-means and principal component analysis. Our model uses discrete variables only. In this sense, it is an intrinsically symbolic approach. The method can be used to obtain a large set of images with labels. This set has a wide range of applications, such as to create a general-use visual encyclopedia. It can also be utilized in a car navigation system to provide the user with the exterior image of a destination.

We implemented a system based on our proposed method and named it “Typi” after “typicality”. Related work is described in Section II. Section III presents our method in detail. Section IV illustrates our implementation, and Section V describes the results of evaluation. Section VI is the conclusion.

II. RELATED WORK

In this section, we describe related work, classified into web image searches, the bag-of-features approach, and retrieval of typical images.

A. Web image search

Several web-based image search engines are currently available, such as Google Image Search, Yahoo! Image Search, and Bing Images. There have also been works applying object identification to images on the web, e.g., WebSeek by Smith and Chang[1]. Web image searching is useful when the user wants to know what items actually look like, especially when shopping on e-commerce sites. To meet such a need, Jing and Baluja proposed a method of ranking images by using a graph structure based on image similarity[2]. Rui et al. proposed a method for
attaching additional annotation to images found in web pages. Their method gathers the text surrounding images within a web page, extends the set of relevant terms using a web image search engine, and filters them by reinforcement learning[3].

An interesting application of a web image search is proposed by Liu et al.: the large set of images available on the web is utilized to retrieve images from a personal photo archive[4]. When a query is given, the system uses it to search images from the web. These images are assumed to be textually annotated, for example, through surrounding texts. They are then used to rank images in a personal photo archive by similarity of visual features.

B. Bag-of-features approach

Recently, the bag-of-features approach, originated from the bag-of-words approach used in text information retrieval, has gained much attention. Vogel and Schiele used a combination of local features to represent higher order concepts such as objects, and they evaluated the precisions of various image retrieval methods[5]. Fei-Fei and Perona applied a generative model used in text analysis to classify natural scene images[6]. Barnard et al. focused on inferring correspondence between regions in an image with terms that are known to be related to the objects appearing in the image[7]. Such inference is especially useful for web image searches, where an image on a web page is usually surrounded by relevant text. They applied a multi-modal extension of the aspect model so that images and co-occurring text could be generated. While a bag-of-features model has been successful for image retrieval and image classification tasks, we used it for finding the typical characteristics of images.

C. Retrieval of typical images

There has also been research on finding typical images of objects. Kennedy and Naaman proposed a method of extracting typical images of landmarks[8]. In addition to using visual features, their method used location metadata attached to images contributed on image-sharing sites. In contrast, our method relies on visual features alone. Wu and Yang proposed a system for finding street landmarks, such as signs, based on extracting object fingerprints from images[9]. Their method requires a user to prepare a correct image (a model image) before extraction. Our method uses unsupervised learning and does not require a correct image beforehand.

van Leuken et al. proposed a method for obtaining diversified image search results[10]. Their goal is roughly opposite to ours, but the reason for proposing the method is similar. Since most existing image search engines heavily rely on textual annotation, the visual features of images must be utilized to obtain images fulfilling a specific need, such as being diverse or typical. van Zwol and Sigurbjornsson proposed an image retrieval system based on “facets,” which are mappings between objects[11]. For example, a city is linked to its different sightseeing spots through facets. Their motivation is that since the results of image search engines are too diversified, they must be classified using facets in order to provide a better user interface. Facets are similar to the notion of “aspect” that we use. The most significant facet is likely to correspond to the aspect representing typicality. The difference is that while facets must be manually constructed, aspects are automatically obtained from given data.

Our previous paper proposed a method of retrieving typical images using hierarchical clustering and the aspect model[12]. In this paper, we have extended the approach based on the aspect model. Specifically, we added a filtering mechanism for removing irrelevant aspects using entropy. We performed further experiments to verify the effectiveness of our proposed method using both artificial and real data.

III. Method

In this section, we describe our proposed method. Our proposed algorithm is as follows.

Algorithm

1. Collect a set of images using an existing web image search engine.
2. For each image, perform the following.
   a. Create histograms of RGB and brightness from the image.
   b. Binarize the image by discriminant analysis of the histograms.
   c. If the ratio of the regions obtained by the discriminant analysis exceeds the threshold, use the p-tile method for binarization instead.
   d. Trace the border and obtain the object’s boundary.
3. Estimate parameters of the generative model using dyadic data consisting of pixels in the object region and color features.
4. Select the “top aspect,” which is assumed to express typicality.
5. Rank images using conditional probability under the top aspect.

Figure 1 shows the flow of our method.

Since our previous paper described the methods of extracting object regions in more detail[12], the rest of this section mainly describes the method of selecting the “top aspect” assumed to express typicality.

A. Model for typicality

The results of a web image search usually contain various objects that are relevant to the query. For example, for a query “iris,” search results would contain a type
observed pair (yellow, or white. of “iris,” for example, the object region can be purple, expressed by a mixture of typical features. In the case apply this model because we assume typicality to be

consistent of discrete observable and latent variables. We express using a mixture model, which is a probabilistic

image, and a part of the eye. In addition, irises are of various colors, ranging from purple to yellow. Images may also show a field with irises. Such variety can be well

may also show a field with irises. Such variety can be well

Our model is based on the aspect model, a model

consisting of discrete observable and latent variables. We apply this model because we assume typicality to be

expressed by a mixture of typical features. In the case of “iris,” for example, the object region can be purple, yellow, or white.

The aspect model[13] is a model that assumes an observed pair \((x, y)\) is conditionally independent under a discrete latent variable \(z\). We modify it to include parameters representing the concept for typicality. Our model is illustrated in Figure 2.

\[
\phi_{ij} = \frac{\phi_{ij}}{\sum_{j'} \phi_{ij'}}
\]

\(\psi\) is also a set of \(\dim(z)\) probabilistic vectors \(\psi_i\) generated in the same manner as \(\phi_i\).

In contrast, \(\mu\) is a vector with \(\mu_1 = c\), where \(c\) is a constant indicating the probability that the “aspect for typicality” \(z\) occurs. \(c\) is set to be a larger value than the other components of \(\mu\), e.g., to 0.5. The other components of \(\mu\) (there are \(\dim(z) - 1\) of them) are obtained by a random division of \(1 - c\). They can be obtained by randomly dividing 1 into partitions in the same manner as \(\phi_i\) and \(\psi_i\) and multiplying each partition by \(1 - c\).

The graphical model in Figure 2 indicates the following conditional independence.

\[
p(x, y, z) = p(x|z)p(y|z)p(z)
\]

Another way to look at it is as the generative model of observed data \((x, y)\). An aspect \(z\) is generated from a distribution \(p(z|\mu)\). \(x\) and \(y\) are generated under conditioned probabilities \(p(x|z, \phi)\) and \(p(y|z, \psi)\). If we express the observed frequency of the pair \((x, y)\) by \(n(x, y)\), the log-likelihood of the dyadic data \(L(x, y)\) is expressed as \(\ln \prod_{x,y} p(x, y)^{n(x,y)}\). On the basis of the above-mentioned conditional independence, we can transform the equation as follows.

\[
L(x, y) = \sum_{x,y} n(x, y) \ln p(x, y)
\]

When we estimate the parameters of the model using maximum likelihood, it is easier to maximize the Q function. We therefore use the EM algorithm. The Q function is the expectation of the log-likelihood of the complete data set[14]. In the following equation, \(q(z)\) is the distribution of \(z\) using the estimated value of \(z\).
$E_q(z)$ refers to obtaining the expectation using distribution $q(z)$. By substituting our generative model, we obtain the following.

\[
Q(p, q) = E_q(z) \left[ \sum_{x, y} n(x, y) \ln p(x, y, z) \right]
\]

\[
= \sum_z \sum_{x, y} q(z)n(x, y)(\ln p(x|z) + \ln p(y|z) + \ln p(z))
\]

By maximizing the $Q$ function, we obtain the following E-step and M-step.

E-step:

\[
q(z|x, y) = \frac{q(x|z)q(y|z)q(z)}{\sum_{z'} q(x|z')q(y|z')q(z')}
\]

M-step:

\[
q(x|z) = \frac{n(x, y)q(z|x, y)}{\sum_{x', y} n(x', y)q(z|x', y)}
\]

\[
q(y|z) = \frac{n(x, y)q(z|x, y)}{\sum_{x, y'} n(x, y')q(z|x, y')}
\]

\[
q(z) = \frac{1}{N} \sum_{x, y} n(x, y)q(z|x, y)
\]

We repeat the process until the convergence condition is fulfilled. As a result, we obtain the estimated distributions $q(x|z)$, $q(y|z)$, and $q(z)$. In our proposed method, $x$ indicates an image and $y$ indicates an image feature, described in the next subsection.

B. Image features

In our proposed method, we model typicality as a mixture of image features. Although we focus on color features in this paper, future work will extend to other features such as texture and shape. To express the color feature as a vector, we divide the color space into color regions. From now on, we refer to each color region as a "color." Similar colors are grouped into a component of the vector. Hue is divided into finer details, since it is usually intrinsic to the object, while brightness and saturation vary depending on lighting. Colors with low brightness are considered as black, and those with low saturation are considered as either dark gray, light gray, or white.

For each color, the system counts how many pixels have that color and creates an HSV vector having the numbers as its components. The fact that an image feature $y$ appears in the object region of an image $x$ is considered to be observed datum $(x, y)$.

We used HSV color space for creating feature vectors, since it is known to represent human perception better than RBG color space. There are other color spaces, but we chose HSV for computational simplicity and also because it is intuitively understandable. Since dimensions of HSV (hue, saturation and brightness) have intuitive meanings, it is easier to adjust the system using our knowledge on each color feature. For example, we divided hue more finely than saturation and brightness since we assumed that hue is less likely to change under different lighting, when compared to the other two dimensions.

C. Extracting aspect expressing typicality

An aspect $z$ with higher $q(z)$ is more likely to be observed, therefore it is considered to be more important in the image set. $q(y|z)$ indicates the probability that an image feature $y$ is generated from an aspect $z$.

If the aspect captures the typical characteristics of an object, it is likely to consist of several colors rather than of a single color. In a pre-experiment, we observed that aspects with high probability of a single color are less likely to express typicality. However, if an aspect generates all colors equivalently, it does not have any characteristics and is not appropriate as an expression of typicality, even if it has high $q(z)$. We therefore introduce entropy $H[q(y|z)]$ for filtering such inappropriate colors.

Aspects are sorted by $H[q(y|z)]$, and those that come below or over threshold ranks are removed. This filtering can be expressed as follows.

\[
\alpha|Z| < \text{ranking}(H[q(y|z)]) < \beta|Z|
\]

\[
|Z| \text{ is the number of aspects, and} \text{ranking}(H[q(y|z)]) \text{ is the rank of an aspect } z \text{ when sorted by the decreasing order of entropy } H[q(y|z)]. 0 \leq \alpha, \beta \leq 1 \text{ are the coefficients for determining the range that the aspects to be used fall within.}
\]

From the set of aspects that fulfill the condition of entropy indicated by Expression 5, we use $z_{(1)}$ to represent an aspect $z$ with the highest value of $q(z)$ and call it the "top aspect." By use of this aspect, the "typicality" of an image is calculated as follows.

\[
typicality(x) = q(x|z_{(1)})
\]
region where the saturation is below 0.2, if the brightness is between 0.2 and 0.6, it is dark gray. If the brightness is between 0.6 and 0.8, it is light gray. If the brightness is over 0.8, it is white. For the remaining region, brightness is divided into 3 ranges, saturation into 3, and hue into 18. The resulting HSV vector has \((18 \times 3 \times 3) + 4 = 166\) components. The division method is illustrated below.

Division of color space

\[
\text{if } (V < 0.2) \\
\quad \text{color} = \text{black} \\
\text{else if } (S < 0.2) \\
\quad \text{if } (V < 0.6) \\
\quad \quad \text{color} = \text{dark gray} \\
\quad \text{else if } (V < 0.8) \\
\quad \quad \text{color} = \text{light gray} \\
\quad \text{else} \\
\quad \quad \text{color} = \text{white} \\
\text{else} \\
\quad \text{assign into any of 162 color regions using the values of H, S, V}
\]

For filtering aspects by entropy, we used \(\alpha = 0.4\) and \(\beta = 0.8\). The aspect \(z\) can take 10 values. This is set based on a pre-experiment indicating that when 100 images obtained as search results were clustered, the number of groups consisting of more than 2 elements was usually less than 10.

Criteria for judging convergence are that the difference is below \(10^{-5}\) or have updated the estimated parameters for over 300 times. Since the log-likelihood of our model has local maxima, we run five trials starting from different initial values. We use the set of parameters with the highest log-likelihood.

C. Interface

Figure 3 is a system snapshot. The results of ranking by Google Image Search and of our method are presented. Figure 4 shows the image evaluation mode. The evaluator can click on the images and classify them as correct or incorrect, enabling evaluation with less effort.

V. EVALUATION

A. Evaluation using artificial data

Before evaluating our proposed method using real data, we carried out a set of experiments with artificial data. Samples were generated from the model illustrated in Figure 2. Real parameters were set randomly, and the estimated parameters were compared to the real ones using the Kullback-Leibler (KL) divergence. In this subsection, we use \(q(x|z), q(y|z),\) and \(q(z)\) to indicate the “real” distributions used for generating samples. In contrast, we use \(q(x|z), q(y|z),\) and \(q(z)\) for the estimated distributions.

The dimension of vector \(z\) for real distribution is indicated by \(\dim(z)_{\text{real}}\). The dimension of vector \(z\) used in the estimated model is indicated by \(\dim(z)_{\text{est}}\). Since the value \(\dim(z)_{\text{real}}\) is unknown when doing an estimation, it is important that any value of \(\dim(z)_{\text{real}}\) can be handled. In the latter half of the experiments, we compared the results for different values of \(\dim(z)_{\text{real}}\) while \(\dim(z)_{\text{est}}\) remained unchanged. We use \(\hat{z}\) to indicate a value of vector \(z\) with \(z_1 = 1\). We also use \(z^{(1)}\) to indicate a value of vector \(z\) with the highest \(q(z)\). In our model, \(\hat{z}\) is the real aspect for typicality and \(z^{(1)}\) is the estimated aspect for typicality.

In the experiment, we used \(\dim(x) = 100\) and \(\dim(y) = 100\). We altered \(\dim(z)_{\text{real}}\) and \(\dim(z)_{\text{est}}\) to determine the effect of varying the number of aspects.

We first set \(p(z = \hat{z}) = \mu_1\) to a certain value \(c\). We then generate the other components for \(\mu\) based on a sampling from the uniform distribution in [0, 1]. Parameters \(\phi\) and \(\psi\) are generated as well. They are the “real parameters” of the model. Dyadic data \((x, y)\) are
generated from these real parameters. In the experiment, we generated 2,000,000 samples for the training set and 1,000 samples for the test set. By performing the EM algorithm described in subsection III-A on the samples, we obtain the estimated parameters \( \hat{\mu} \), \( \hat{\phi} \), and \( \psi \) for \( q(x|z), q(y|z) \), and \( q(z) \), respectively.

For evaluation, we used the KL divergence between the true distribution and the estimated distribution. It is common to use the KL divergence to measure the similarity between probability distributions. When distributions are identical, the KL divergence is zero. A larger KL divergence value indicates a larger level of difference between distributions. We measure \( KL(p(x|z = \hat{z}), q(x|z = z_{(1)})) \).

The whole process of real parameters setting, sample generation, and parameter estimation was repeated 100 times, and the average KL divergences were calculated. Figure 5 illustrates the flow of evaluation. Each large box indicates that the steps inside it are repeated.

Figure 5. Flow of evaluation using artificial data

In the first set of experiments, we observed the change in the KL divergence for different values of \( \mu_1 \) ranging from 0.1 to 0.9. Figures 6 – 9 illustrate the results. The dimensions of \( z \) were set to \( \text{dim}(z)_{\text{real}} = 8 \) and \( \text{dim}(z)_{\text{est}} = 10 \). \( q'(x|z = z'_{(1)}) \) and \( q'(y|z = z'_{(1)}) \) are distributions with randomly generated parameters, while \( q(x|z = z_{(1)}) \) and \( q(y|z = z_{(1)}) \) are distributions with estimated parameters. \( z'_{(1)} \) represents the aspect with the highest value of \( q'(z) \). The y-axis labels are abbreviated for simplicity. In Figure 6, for example, the y-axis label is \( KL(p(x|z), q(x|z)) \). This actually represents that \( KL(p(x|z = \hat{z}), q(x|z = z_{(1)})) \) is being compared with \( KL(p(x|z = \hat{z}), q'(x|z = z'_{(1)}) \).

The results show that the system effectively estimates the real parameters, since the KL divergence is much smaller for the estimated distributions. Figures 6 – 7 show that as the value of \( \mu_1 \) increases, the KL divergence decreases for the estimated parameters, whereas such a decrease does not occur for randomly generated parameters. This indicates that when the aspect for typicality is likely to appear in the data, estimation is more accurate. In contrast, when the aspect for typicality is unlikely to appear (e.g., \( \mu_1 = 0.1 \)), estimation is not as accurate.

Figure 6. \( KL(p(x|z = \hat{z}), q(x|z = z_{(1)})) \) by \( \mu_1 \)

In the second set of experiments, we observed changes in the KL divergence by altering \( \text{dim}(z)_{\text{real}} \) from 2 to 20 (by increments of 2). We set \( \text{dim}(z)_{\text{est}} = 10 \) and \( \mu_1 = 0.5 \). Figures 8 – 9 show the results of the experiments. They indicate that when the number of aspects in the real model is small (e.g., \( \text{dim}(z)_{\text{real}} = 2 \)), estimation is not accurate. For higher values of \( \text{dim}(z)_{\text{real}} \), they show that estimation is accurate.

Figure 8. \( KL(p(x|z = \hat{z}), q(x|z = z_{(1)})) \) by \( \text{dim}(z)_{\text{real}} \)
B. Evaluation using real data

In the experiments using real data, we used 30 queries from the category “flowers.” For each query, 100 images were collected, resulting in 3,000 images in total. We used Google Image Search to collect images[16]. We evaluated the averaged and individual top-k precisions. Table I lists the queries used in the experiment. For queries consisting of two or more words (such as “scarlet pimpernel”), the query was put into quotations marks, enabling a phrase search.

C. Evaluation criteria

To judge whether an image is typical or not, we need a unified criterion. As mentioned in Section I, we defined “image I is a typical image for a query Q if Q is an appropriate label for I.” Therefore, it is a necessary condition that the object specified by the query appears in the image.

If an image contains more than one object and it is not sufficient for determining the object’s visual characteristics, we considered it to be an incorrect image. For example, for the query “iris,” the image should have an iris that is large enough for the evaluator to identify as an iris. In the evaluation, we assume that the evaluator has sufficient knowledge on the object being queried.

D. Ranking example

In this subsection, we exemplify the top-ranked images by the rankings by Google Image Search and by the top aspect according to our method. In Figures 10 – 14, images are ordered from the top left, going left to right and then down. We added an “x” to images that were judged to be incorrect under our evaluation scheme.

The method allows a wider range of irises with iris also means a part of the eye, the top-ranked images by the search engine contained pictures of this (10th and 15th images in Figure 10). This is one weakness of the text-based ranking mechanism used by the image search engine. Figure 11 shows the images obtained by our method. More images were correct than those obtained by the search engine, resulting in higher precision.

To judge whether an image is typical or not, we need a unified criterion. As mentioned in Section I, we defined “image I is a typical image for a query Q if Q is an appropriate label for I.” Therefore, it is a necessary condition that the object specified by the query appears in the image.

If an image contains more than one object and it is not sufficient for determining the object’s visual characteristics, we considered it to be an incorrect image. For example, for the query “iris,” the image should have an iris that is large enough for the evaluator to identify as an iris. In the evaluation, we assume that the evaluator has sufficient knowledge on the object being queried.

Figures 10 – 12 are the results for the query “iris.” Since iris also means a part of the eye, the top-ranked images by the search engine contained pictures of this (10th and 15th images in Figure 10). This is one weakness of the text-based ranking mechanism used by the image search engine. Figure 11 shows the images obtained by our method. More images were correct than those obtained by the search engine, resulting in higher precision.

To judge whether an image is typical or not, we need a unified criterion. As mentioned in Section I, we defined “image I is a typical image for a query Q if Q is an appropriate label for I.” Therefore, it is a necessary condition that the object specified by the query appears in the image.

If an image contains more than one object and it is not sufficient for determining the object’s visual characteristics, we considered it to be an incorrect image. For example, for the query “iris,” the image should have an iris that is large enough for the evaluator to identify as an iris. In the evaluation, we assume that the evaluator has sufficient knowledge on the object being queried.

Figures 10 – 12 are the results for the query “iris.” Since iris also means a part of the eye, the top-ranked images by the search engine contained pictures of this (10th and 15th images in Figure 10). This is one weakness of the text-based ranking mechanism used by the image search engine. Figure 11 shows the images obtained by our method. More images were correct than those obtained by the search engine, resulting in higher precision.

To judge whether an image is typical or not, we need a unified criterion. As mentioned in Section I, we defined “image I is a typical image for a query Q if Q is an appropriate label for I.” Therefore, it is a necessary condition that the object specified by the query appears in the image.

If an image contains more than one object and it is not sufficient for determining the object’s visual characteristics, we considered it to be an incorrect image. For example, for the query “iris,” the image should have an iris that is large enough for the evaluator to identify as an iris. In the evaluation, we assume that the evaluator has sufficient knowledge on the object being queried.
buildings, the results by the search engine contain many such images. In contrast, the results of ranking by the top aspect contain mainly flowers, resulting in high precision. In contrast, the second aspect is assumed to represent buildings in China.

Figures 16 – 17 show the results for the query “bleeding heart.” The search engine results contain images for bleeding heart flower and a heart literally bleeding. The results by our method contain flowers only.

E. Average top-k precisions

For each of 30 queries, we performed experiments using 100 images obtained by an existing web image search engine. Using the rankings by the search engine (Google Image Search) and our proposed method, we evaluated the top-k precisions. The results are shown in Figure 18.

Since the original set consists of 100 images for each query, the top-100 precision is the ratio of correct images to the whole set. In this case, since 61% of 3000 images are correct, the average top-100 precision is 61%. While
the top-5 precision of the search engine’s ranking is 71%, that of the ranking by the top aspect is 84%. For the top-30 precision, the search engine’s ranking is 68% and our method is 80%.

Semi-supervised PCA had 55% for top-5 precision and 63% for top-30 precision. It was inferior to the result of the image search engine. The reason is assumed to be because the image search engine utilizes information other than image features. For example, in case of web image retrieval, it often uses texts that surround the image in the web page it is embedded in. Since PCA does not use such external information, it has lower precisions than the original search engine ranking. Also, the first principal component was likely to consist of features that distinguish black-and-white images to colorful images. Since such a feature has only a small relevance to typicality, it accounted for PCA’s low precision.

The dotted line in Figure 18 indicates the top-k precision of the ranking obtained from the aspect with the highest $q(z)$. The graph also shows that the proposed method has higher precision than the method that does not filter aspects by entropy in the way indicated by Expression 5.
Table II lists the top-k precision for each query used in the experiment.

The result for query “baby blue eyes” is an example where our method fails. The top-30 precision for this query is zero, because the images we obtained in the top ranks were pictures of blue-eyed babies. Such ambiguous phrases often lower the precision of the results.

Figures 19 – 21 show what ratio of the queries fell into different ranges of top-10 precisions. In the graph, \( P \) refers to precision. The size of each region in each circle indicates the number of queries having precisions within the corresponding range.

In the results of the search engine, only 27% of the queries had precisions over or equal to 0.9. In PCA, only 17% of the queries had precisions over or equal to 0.9. Using our method, 70% of the queries had precisions over or equal to 0.9.

G. Processing time

Once the parameters for a query \( q \) are estimated, our system can rank an arbitrary set of images based on typicality as an image of the object \( q \). Since the parameters are computed beforehand, ranking of a new set of image is quite fast. The typicality score for image \( I \) is obtained by a dot product of \( I \)’s feature vector \( y' \) and the estimated parameter vector \( p(y|z = z_{11}) \) for the multinomial distribution of \( y \). Computation resource required is linearly dependent on the dimension of the feature vector and the number of images to be ranked.

We have measured processing time necessary for ranking images, once the parameters were estimated using
either the EM algorithm or PCA. Computation consists of three steps: extraction of object region, construction of feature vectors, and ranking by typicality scores. We used Intel Core2 Duo 2.00GHz 2GB RAM for the experiment. Figures 22 - 23 are histograms indicating how many images required a certain amount of time for processing. The unit is in milliseconds. For the object region extraction, the mode is at around 40 milliseconds. There are some outliers, but they fall within triple the time of the mode. For the construction of feature vectors, the mode is at around 25 milliseconds. Outliers fall within double the time of the mode. The average time required for learning the aspect model for 100 images was 71.1 milliseconds.

The典型性 score of an image $I$ is obtained by

<table>
<thead>
<tr>
<th>Query</th>
<th>Method</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>baby blue eyes</td>
<td>Search en.</td>
<td>0.20</td>
<td>0.40</td>
<td>0.40</td>
<td>0.40</td>
<td>0.38</td>
<td>0.42</td>
<td>0.34</td>
</tr>
<tr>
<td>blaze star</td>
<td>Search en.</td>
<td>0.80</td>
<td>0.50</td>
<td>0.60</td>
<td>0.55</td>
<td>0.58</td>
<td>0.62</td>
<td>0.61</td>
</tr>
<tr>
<td>bleeding heart</td>
<td>Search en.</td>
<td>0.60</td>
<td>0.50</td>
<td>0.60</td>
<td>0.67</td>
<td>0.63</td>
<td>0.64</td>
<td>0.62</td>
</tr>
<tr>
<td>buttercup</td>
<td>Search en.</td>
<td>0.60</td>
<td>0.70</td>
<td>0.62</td>
<td>0.70</td>
<td>0.67</td>
<td>0.72</td>
<td>0.78</td>
</tr>
<tr>
<td>calliopsis</td>
<td>Search en.</td>
<td>0.60</td>
<td>0.80</td>
<td>0.70</td>
<td>0.67</td>
<td>0.70</td>
<td>0.72</td>
<td>0.78</td>
</tr>
<tr>
<td>chicory</td>
<td>Search en.</td>
<td>0.60</td>
<td>0.50</td>
<td>0.50</td>
<td>0.53</td>
<td>0.50</td>
<td>0.54</td>
<td>0.56</td>
</tr>
<tr>
<td>chinese houses</td>
<td>Search en.</td>
<td>0.60</td>
<td>0.60</td>
<td>0.55</td>
<td>0.57</td>
<td>0.60</td>
<td>0.62</td>
<td>0.56</td>
</tr>
<tr>
<td>cingot</td>
<td>Search en.</td>
<td>0.60</td>
<td>0.70</td>
<td>0.85</td>
<td>0.70</td>
<td>0.58</td>
<td>0.54</td>
<td>0.39</td>
</tr>
<tr>
<td>clematis</td>
<td>Search en.</td>
<td>1.00</td>
<td>1.00</td>
<td>0.85</td>
<td>0.83</td>
<td>0.73</td>
<td>0.64</td>
<td>0.39</td>
</tr>
<tr>
<td>columbine</td>
<td>Search en.</td>
<td>0.20</td>
<td>0.20</td>
<td>0.15</td>
<td>0.20</td>
<td>0.28</td>
<td>0.30</td>
<td>0.36</td>
</tr>
<tr>
<td>daisy</td>
<td>Search en.</td>
<td>1.00</td>
<td>1.00</td>
<td>0.75</td>
<td>0.80</td>
<td>0.73</td>
<td>0.70</td>
<td>0.59</td>
</tr>
<tr>
<td>dandelion</td>
<td>Search en.</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.93</td>
<td>0.78</td>
<td>0.68</td>
<td>0.38</td>
</tr>
<tr>
<td>forget me nots</td>
<td>Search en.</td>
<td>1.00</td>
<td>0.90</td>
<td>0.90</td>
<td>0.83</td>
<td>0.83</td>
<td>0.82</td>
<td>0.72</td>
</tr>
<tr>
<td>franciscan wallflower</td>
<td>Search en.</td>
<td>0.80</td>
<td>0.70</td>
<td>0.65</td>
<td>0.73</td>
<td>0.65</td>
<td>0.56</td>
<td>0.47</td>
</tr>
<tr>
<td>harebell</td>
<td>Search en.</td>
<td>1.00</td>
<td>1.00</td>
<td>0.90</td>
<td>0.90</td>
<td>0.88</td>
<td>0.90</td>
<td>0.78</td>
</tr>
<tr>
<td>hawkweed</td>
<td>Search en.</td>
<td>1.00</td>
<td>1.00</td>
<td>0.95</td>
<td>0.87</td>
<td>0.88</td>
<td>0.86</td>
<td>0.80</td>
</tr>
<tr>
<td>ice plant</td>
<td>Search en.</td>
<td>0.40</td>
<td>0.70</td>
<td>0.75</td>
<td>0.73</td>
<td>0.68</td>
<td>0.64</td>
<td>0.52</td>
</tr>
<tr>
<td>iriss</td>
<td>Search en.</td>
<td>0.80</td>
<td>0.80</td>
<td>0.85</td>
<td>0.87</td>
<td>0.88</td>
<td>0.86</td>
<td>0.86</td>
</tr>
<tr>
<td>johnny jump up</td>
<td>Search en.</td>
<td>0.80</td>
<td>0.80</td>
<td>0.90</td>
<td>0.80</td>
<td>0.68</td>
<td>0.74</td>
<td>0.77</td>
</tr>
<tr>
<td>lady’s slipper</td>
<td>Search en.</td>
<td>1.00</td>
<td>1.00</td>
<td>0.95</td>
<td>0.90</td>
<td>0.93</td>
<td>0.90</td>
<td>0.78</td>
</tr>
<tr>
<td>morning glory</td>
<td>Search en.</td>
<td>1.00</td>
<td>1.00</td>
<td>0.95</td>
<td>0.87</td>
<td>0.88</td>
<td>0.86</td>
<td>0.80</td>
</tr>
<tr>
<td>pale flax</td>
<td>Search en.</td>
<td>0.80</td>
<td>0.50</td>
<td>0.65</td>
<td>0.70</td>
<td>0.68</td>
<td>0.66</td>
<td>0.67</td>
</tr>
<tr>
<td>pale flax</td>
<td>Search en.</td>
<td>1.00</td>
<td>0.80</td>
<td>0.80</td>
<td>0.87</td>
<td>0.80</td>
<td>0.84</td>
<td>0.67</td>
</tr>
<tr>
<td>pale flax</td>
<td>Search en.</td>
<td>1.00</td>
<td>0.90</td>
<td>0.90</td>
<td>0.80</td>
<td>0.68</td>
<td>0.64</td>
<td>0.52</td>
</tr>
<tr>
<td>pale flax</td>
<td>Search en.</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.93</td>
<td>0.93</td>
<td>0.94</td>
<td>0.87</td>
</tr>
<tr>
<td>pale flax</td>
<td>Search en.</td>
<td>1.00</td>
<td>0.90</td>
<td>0.90</td>
<td>0.80</td>
<td>0.68</td>
<td>0.64</td>
<td>0.65</td>
</tr>
<tr>
<td>pale flax</td>
<td>Search en.</td>
<td>0.80</td>
<td>0.80</td>
<td>0.75</td>
<td>0.80</td>
<td>0.80</td>
<td>0.82</td>
<td>0.60</td>
</tr>
<tr>
<td>pale flax</td>
<td>Search en.</td>
<td>0.20</td>
<td>0.40</td>
<td>0.40</td>
<td>0.47</td>
<td>0.45</td>
<td>0.42</td>
<td>0.37</td>
</tr>
<tr>
<td>pale flax</td>
<td>Search en.</td>
<td>0.80</td>
<td>0.60</td>
<td>0.50</td>
<td>0.50</td>
<td>0.43</td>
<td>0.42</td>
<td>0.35</td>
</tr>
<tr>
<td>pale flax</td>
<td>Search en.</td>
<td>0.20</td>
<td>0.10</td>
<td>0.15</td>
<td>0.17</td>
<td>0.15</td>
<td>0.18</td>
<td>0.15</td>
</tr>
<tr>
<td>pale flax</td>
<td>Search en.</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
<td>0.77</td>
<td>0.73</td>
<td>0.76</td>
<td>0.65</td>
</tr>
<tr>
<td>texas bluebonnet</td>
<td>Search en.</td>
<td>0.80</td>
<td>0.80</td>
<td>0.75</td>
<td>0.80</td>
<td>0.85</td>
<td>0.84</td>
<td>0.85</td>
</tr>
<tr>
<td>Texas bluebonnet</td>
<td>Search en.</td>
<td>0.80</td>
<td>0.90</td>
<td>0.95</td>
<td>0.93</td>
<td>0.93</td>
<td>0.88</td>
<td>0.85</td>
</tr>
<tr>
<td>tidy tips</td>
<td>Search en.</td>
<td>0.80</td>
<td>0.90</td>
<td>0.90</td>
<td>0.83</td>
<td>0.83</td>
<td>0.84</td>
<td>0.87</td>
</tr>
<tr>
<td>tiger lily</td>
<td>Search en.</td>
<td>0.60</td>
<td>0.80</td>
<td>0.75</td>
<td>0.77</td>
<td>0.78</td>
<td>0.76</td>
<td>0.80</td>
</tr>
<tr>
<td>water arum</td>
<td>Search en.</td>
<td>0.80</td>
<td>0.80</td>
<td>0.75</td>
<td>0.70</td>
<td>0.63</td>
<td>0.66</td>
<td>0.45</td>
</tr>
<tr>
<td>wild radish</td>
<td>Search en.</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
<td>0.75</td>
<td>0.70</td>
<td>0.70</td>
<td>0.75</td>
</tr>
</tbody>
</table>

The typicality score of an image $I$ is obtained by...
multiplying a vector \( p(w|z = 1) \) and the image feature vector \( w \) of image \( I \). The system ranks the images according to the scores. We measured the processing time for ranking 100 images for each query. Figure 24 compares the result for the aspect model and PCA.

The result shows that the computation speed of our proposed method is competitive.

VI. CONCLUSION

In this paper, we presented a method that ranks web image search results in the order of typicality, by extracting the top aspect. One strong point of our approach is that it is based on the probability theory. For example, since all values are actually parameters of distributions, threshold values can be set with a probabilistic basis.

Our present implementation uses color features only, but we plan to use more complex image features in future work. In our evaluation, we used flowers, which are objects that have strong color characteristics. As not all objects have such strong color features, in future work we plan to use shape and texture, in addition to color, to deal with such objects.

We plan to extend our method by using a Bayesian model, where parameter \( \mu_1 \) is no longer a constant but a part of a probabilistic vector with distribution \( p(\mu|\alpha) \). Distributions for \( \phi \) and \( \psi \) can also be specified by hyperparameters \( \beta \) and \( \gamma \). The model is shown in Figure 25. The hyperparameter \( \alpha \) can be estimated if enough data is given. This would not be difficult, as the amount of images on the World Wide Web is enormous.

One of our future goals is to build a general task image recognition engine that gives the name of an object when an arbitrary image is given. Attaching correct labels to a large set of images obtained from the web would contribute to building such a system.

Although we use a single latent probabilistic vector for typicality, it can possibly be modeled with a more detailed structure. Exploration of such a model is also future work.

ACKNOWLEDGMENTS

This work was supported in part by a MEXT Grant-in-Aid for Strategic Formation of Research Infrastructure for Private University “Sharing of Research Resources by Digitization and Utilization of Art and Cultural Materials” (Grant Number: S0991041) and MEXT Grant-in-Aid for
Young Scientists (B) “Object Identification System using Web Image Collection and Machine Learning” (Leader: Taro Tezuka, Grant Number: 21700121).

REFERENCES


Taro Tezuka is currently a lecturer at Ritsumeikan University. His main research interests are information retrieval, web mining, and machine learning. He received a Ph.D. in Informatics from Kyoto University in 2005. Prof. Tezuka is a member of the Information Processing Society of Japan (IPSJ), the Institute of Electronics, Information and Communication Engineers (IEICE), and the Database Society of Japan (DBSJ).

Akira Maeda is currently an associate professor at Ritsumeikan University. His main research interests are digital libraries and information retrieval. He received a Ph.D. in Engineering from Nara Institute of Science and Technology (NAIST) in 2000. Prof. Maeda is a member of the Information Processing Society of Japan (IPSJ), the Institute of Electronics, Information and Communication Engineers (IEICE), and the Database Society of Japan (DBSJ), the IEEE Computer Society, and the ACM.

© 2011 ACADEMY PUBLISHER