Int J Digit Libr (2004) 4: 258–271 / Digital Object Identifier (DOI) 10.1007/s00799-004-0090-z

Effective image annotation for searches using multilevel semantics

Pu-Jen Cheng, Lee-Feng Chien

Institute of Information Science, Academia Sinica, 128 Academy Rd, Sec. 2, Nankang, Taipei 115, Taiwan, R.O.C.
e-mail: {pjcheng, lfchien}@iis.sinica.edu.tw

Published online: 11 November 2004 –

Abstract. There is an increasing need for automatic image annotation tools to enable effective image searching in digital libraries. In this paper, we present a novel probabilistic model for image annotation based on content-based image retrieval techniques and statistical analysis. One key difficulty in applying statistical methods to the annotation of images is that the number of manually labeled images used to train the methods is normally insufficient. Numerous keywords cannot be correctly assigned to appropriate images due to lacking or missing information in the labeled image databases. To deal with this challenging problem, we also propose an enhanced model in which the annotated keywords of a new image are defined in terms of their similarity at different semantic levels, including the image level, keyword level, and concept level. To avoid missing some relevant keywords, the model labels the keywords with the same concepts as the new image. Our experimental results show that the proposed models are effective for annotating images that have different qualities of training data.

Keywords: Image annotation – Keyword clustering

1 Introduction

With the rapid growth of images in digital libraries, there is an increasing need for automatic tools to help people annotate images. Given a set of images that have been labeled by keywords, the annotation problem is how to assign the keywords to new images. This task facilitates further image management such as text-based image retrieval, the creation of metadata, and the organization and presentation of image content.

Most studies on automatic image annotation focus on inferring high-level semantic information from low-level image features. Some [8, 14, 24] make use of image recognition techniques to classify images into semantically meaningful categories and then label the images by the keywords that have been manually assigned to those categories. Image recognition methods are appropriate for constrained domains, but they are not reliable for many complicated applications. Meanwhile, other researchers apply relevance feedback methods to obtain keywords from a set of labeled images [11, 16]. Generally, relevance feedback methods have the advantage of providing better accuracy than image recognition since more accurate keywords can be obtained by user interaction.

For many domains, there are numerous digital libraries that contain well-labeled images, e.g., American Memory (`http://memory.loc.gov`), U.S. National Library of Medicine (`http://wwwihm.nlm.nih.gov`), Corbis (`http://pro.corbis.com`) and Corel image libraries (`http://www.corel.com`), and Yahoo! Picture Gallery (`http://gallery.yahoo.com`). We want to determine whether it is possible to automatically assign effective keywords to a new image based on similar images that have been labeled already. Unlike previous works, our goal is to integrate content-based image retrieval techniques and statistical analysis for image annotation. We propose a novel probabilistic model for ranking a set of keywords according to their importance in representing the semantic of a given image. The model is based on the hypothesis that similar images may partially share the same keywords. For example, Fig. 1 shows six images pertaining to historical artifacts in the National Palace Museum (`http://www.npm.gov.tw`). Although it is very difficult to recognize the objects appearing in these images, content-based retrieval technologies [7, 15] provide a way to find similar images based on low-level features such as colors, textures, and shapes. In this case, the similar images are (a) and (b), (c) and (d), and (e) and (f). Suppose (a) has been labeled by the keyword *china*; then (b) will probably be labeled by *china* because it is similar to (a).

Fig. 1. Example of similar images

One key difficulty in finding keywords based on similar images is that the number of manually labeled images is usually insufficient. In general, digital libraries contain numerous collections of images and keywords, but a considerable number of keywords in the labeled images might be missing. This problem is even more serious when images have few similar images. Also, it is difficult to obtain effective keywords from a small set of similar images that have insufficiently labeled keywords.

To overcome the insufficiency (or sparseness) problem of manually labeled images, in the proposed model the annotated keywords of a new image are determined in terms of their similarity at different semantic levels, including the image level, keyword level, and concept level. The set of candidate keywords is mainly selected from the labeled keywords of the images that are visually similar to the new image. To avoid missing some relevant keywords in the candidate set, the concepts constituted by the candidate keywords are considered. Based on the concept level, the proposed model has the capacity to suggest keywords with the same concepts. As a result, keywords not appearing in the labeled images can be extracted, while other keywords irrelevant to the overall concepts of the similar images can be eliminated. Consider the example again. Suppose most images similar to (e) and (f) contain the keyword *penmanship* and the authors' names. The set of the authors' names constitutes the concept of *calligrapher*. Though it is difficult to identify an image's author automatically, our model can recommend a set of calligraphers as candidate keywords. To satisfy the need for automatic image annotation, we have also designed an algorithm to generate the concept level automatically. Our method clusters relevant keywords based on statistical analysis of their occurrences and co-occurrences in a corpus.

To evaluate the model's performance, we developed a prototype system that extracts both colors and textures for global and local features. It also supports relevance feedback strategies to automatically adjust the meaning of similarities among different features. The experiments with the Corel image library show that the proposed method is effective and can achieve 54.6% recall and 55.1% precision.

The rest of the paper is organized as follows. Section 2 presents related work on automatic image annotation. Section 3 describes the problem and its challenges. Sections 4 and 5 introduce the proposed methods and implementation issues, respectively. Section 6 describes the performance evaluation. Section 7 contains a discussion. Finally, in Sect. 8, we present our conclusions.

2 Related work

A number of works that explore automatic image annotation based on low-level image features assume that semantically relevant images have similar visual features. They attempt to classify given images into categories and label them by the keywords associated with those categories. Wang et al. [24] classify images into semantic categories such as textured or nontextured and graph or photograph, which enhances image retrieval by permitting semantically adaptive search methods. Meanwhile, Paek et al. [14] combine visual and textual features. Visual objects are identified by the clustering of segmented regions from images and represented with the *tf-idf* scheme [19] by which images can be classified into two categories, indoor and outdoor, according to their visual objects. Other types of image classes include city vs. landscape [23] and portrait vs. nonportrait [8]. Chang et al. [2] present semantic visual templates comprised of a set of example objects that represent the semantic associated with the templates. The templates need to be generated semiautomatically. Due to the accuracy limitations of computer vision and pattern recognition technologies, most of these methods focus on certain semantic types whose features have a high degree of discrimination for particular user-defined classes.

Other researchers apply relevance feedback methods to obtain keywords from a set of labeled images. Minka et al. [13, 16] introduce an interactive annotation method to find the association between semantic labels and primitive image features by using positive and negative examples. Meanwhile, Lu et al. [11] utilize user feedback to incorporate additional keywords for image annotation. When a user advises that some images are relevant to a query, the method updates the annotation of the images by linking the query with the images. Although it has been proved that relevance feedback techniques are generally more efficient than manual annotation and more accurate than image recognition, they still require much user interaction.

Less attention has been devoted to automatic image annotation based on statistical analysis. Barnard et al. [1] present a statistical model for hierarchically modeling

the statistics of word and feature occurrence and co-occurrence and organizing image collections that simultaneously integrate semantic and visual information. Higher levels of the model provide more general terms, while lower levels provide specific ones. This model supports automatic image classification as well as the association of terms with images. However, the method needs human assistance to design the hierarchy's topology, which is very sensitive to real data and difficult for people to determine.

3 The problem and challenges

In this section, we will define the image annotation problem and introduce its challenges in real-world applications.

The image annotation problem can be formally defined as follows. Let $I = \{I_1, I_2, \ldots, I_m\}$ be a set of images and $K = \{K_1, K_2, \ldots, K_n\}$ be a set of keywords for labeling image $I_i \in I$. Let P be a probabilistic function mapping $I \times K$ to real numbers varying from 0 to 1. $P(K_j|I_i) = v$ represents the conditional probability of keyword K_j given image I_i. The probability $P(K_j|I_i)$ indicates the degree of importance of keyword K_j in representing the semantic of image I_i. Suppose we have database images D and new images T such that $I = D \cup T$ and $D, T \neq \emptyset$, where database images D are the images that have been labeled already and new images T refer to the images to be annotated. For each new image $I_Q \in T$, the image annotation problem is to rank each keyword $K_j \in K$ according to the conditional probability $P(K_j|I_Q)$, which is estimated through $P(K_j|I_i)$ for all $I_i \in D$. Note that once image I_Q is labeled ($D = D \cup \{I_Q\}$ and $T = T - \{I_Q\}$), subsequent new images in new T will be assessed based on new D. In general, the labeled image I_Q should be further verified by human experts to avoid error propagation. Since in the verification process only incorrect keywords need to be filtered, the cost of labeling a large number of images is still affordable.

The image annotation problem encounters two challenges in real-world applications. First, new image I_Q may have few similar images that have been labeled in database D. Second, the similar images may miss some satisfactory keywords to label them because manual labeling is usually incomplete. Thus, the labeled images in D probably contribute an incomplete set of keywords as candidates for later annotation processing. This explains why low-recall queries appear easily, even in digital libraries with millions of images.

4 Image annotation processing

4.1 Probabilistic model

In this section, we use the *semantic network* to represent the relationships between database images D and keywords K in the annotation problem and propose a basic probabilistic model based on this representation.

The basic semantic network has two levels of nodes that provide empirical associations between database images and keywords. More precisely, the basic semantic network is defined as a tuple (D, K, M_{IK}) where

- D is a set of database images $\{Ii\}$ ($D \subset I$) specifying the *image level*. Image I_i is a raw image, e.g., a JPEG image.
- K is a set of predefined keywords $\{K_j\}$ ($j = 1 \ldots n$) specifying the *keyword level*.
- M_{IK} is a set of weights $\{\beta_{ij}\}$ ($0 \leq \beta_{ij} \leq 1$) on the links between database images D and keywords K. β_{ij} denotes the importance of keyword K_j to database image I_i.

Figure 2 illustrates an example of the basic semantic network, where $D = \{I_1, I_2, I_3\}$, $K = \{Dog, Grass, Lawn, Pet, Running\}$, and $M_{IK} = \{\beta_{11} = 1.0, \beta_{12} = 0.8, \ldots\}$. For simplicity, two nodes are connected if the weight on their link is nonzero.

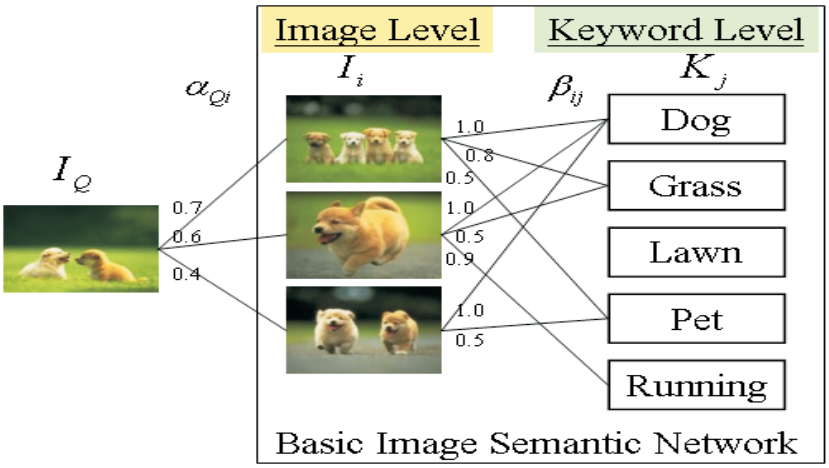


Fig. 2. Example of a basic image semantic network

For each new image I_Q, candidate keywords can be selected according to the labeled keywords of its similar images. α_{Qi}, which varies from 0 to 1, represents the similarity value between images I_Q and I_i. The higher the value, the more similar the images will be. Numerous works [3, 9, 18] on content-based image retrieval provide an automatic means to evaluate the similarity value and retrieve similar images. For example, in Fig. 2, $I_{1\sim3}$ are the similar images of I_Q, where $\alpha_{Q1} = 0.7$, $\alpha_{Q2} = 0.6$, and $\alpha_{Q3} = 0.4$. *Dog*, *Grass*, *Pet*, and *Running* are candidate keywords, but *Lawn* is not considered as a candidate at this stage. We assume the annotation is incomplete here.

Given image I_Q and a candidate keyword K_j, the basic probabilistic model P_1 is defined as follows:

$$P_1(K_j|I_Q) = \frac{W_j^{(1)}}{\sum_{j'=1\ldots n} W_{j'}^{(1)}}, \tag{1}$$

where $W_j^{(1)}$ is the weight of keyword K_j, which is computed as the weighted sum of β_{ij}:

$$W_j^{(1)} = \sum_{\forall i} \alpha_{Qi} \times \beta_{ij}\,. \tag{2}$$

Taking Fig. 2 as an example, the basic probabilistic model produces $P_1(Dog|I_Q) = 0.466$, $P_1(Grass|I_Q) = 0.236$, $P_1(Pet|I_Q) = 0.15$, $P_1(Running|I_Q) = 0.148$, and $P_1(Lawn|I_Q) = 0$. The labeled keywords of the images similar to I_Q can be taken as a "virtual document," which yields a possible semantic interpretation of image I_Q. Each keyword in the document has a different weight $W_j^{(1)}$, which is calculated according to its importance to the similar images and the similarity between I_Q and the similar images. The basic probabilistic model P_1 provides a possible means to rank the keywords, but some rankings are unsatisfactory. For example, $P_1(Lawn|I_Q) = 0$ because no similar images contain the keyword *Lawn*. In addition, the keyword *Running* is a noise and difficult to filter. *Pet* is probably better than *Running* for labeling image I_Q in this example. An enhanced model that is more accurate is, therefore, needed.

4.2 Enhanced model

To alleviate the inaccuracy mentioned above, we now explore the use of the semantic concepts implicit in a set of keywords to build an enhanced probabilistic model. We assume that an image has some concepts constituted by its relevant keywords. Based on the conceptual information, the semantic network in Fig. 2 can be extended to Fig. 3. The extended semantic network is defined as a tuple $(D, K, C, M_{IK}, M_{KC})$ where

- D, K, and M_{IK} are the same as the definitions given in the basic semantic network.
- C is a set of concepts (or semantic categories) $\{C_k\}$ $(k = 1\ldots q)$ specifying *the concept level*. Concept C_k denotes the group of keywords pertaining to a certain concept.
- M_{KC} is a set of weights $\{\gamma_{jk}\}$ $(0 \le \gamma_{jk} \le 1)$ on the links between keywords K and concepts C. γ_{jk} denotes the relevance of keyword K_j to concept C_k.

In the extended network shown in Fig. 3, the concept level mirrors the other keyword level K' $(= K)$ on the right-hand side. Weight γ'_{kj} on the link between concept C_k and keyword K'_j is equivalent to weight γ_{jk}. This helps us explain how to compute the ranking for each keyword later. Moreover, concept C and weights γ_{jk} and γ'_{kj} can be automatically generated with conventional term clustering [4] and categorization methods, respectively. We give the details in Sect. 5.

Like the basic probabilistic model, we assume the relevant keywords of image I_Q constitute another "virtual document," which provides a possible interpretation of image I_Q at the concept level. Given image I_Q and a candidate keyword K_j, the enhanced probabilistic model P_2 is defined as follows:

$$P_2(K_j|I_Q) = \frac{W_j^{(2)}}{\sum_{j'=1\ldots n} W_{j'}^{(2)}}, \tag{3}$$

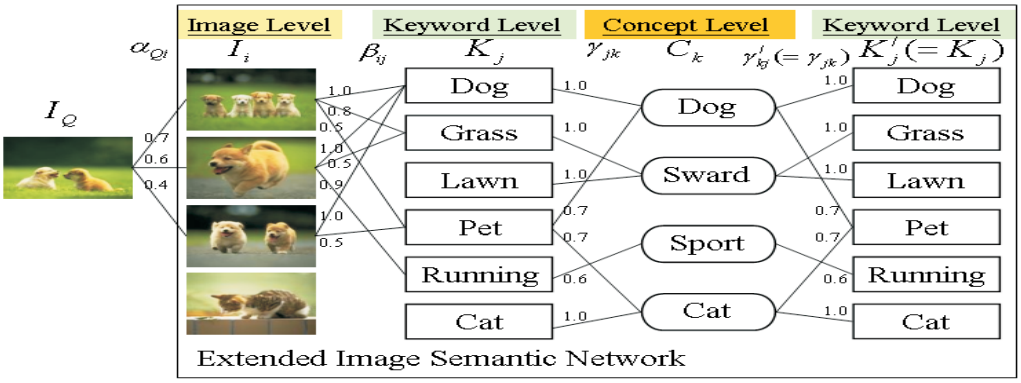


Fig. 3. Example of an extended image semantic network

where weight $W_j^{(2)}$ is computed as

$$W_j^{(2)} = \sum_{K=1\ldots q} \gamma'_{kj} \times \left(\sum_{j'=1\ldots n} W_{j'}^{(1)} \times \gamma_{j'k} \right) , \qquad (4)$$

where $W_j^{(2)}$ is the weighted sum of the concept weights and γ'_{kj} denotes the weight and the formula in parentheses denotes the concept weights.

Obviously model P_1 is a special case of model P_2, where each keyword corresponds to a concept with weight 1. However, this seldom happens in real applications. In the example in Fig. 3, the enhanced probabilistic model produces $P_2(Dog|I_Q) = 0.44$, $P_2(Pet|I_Q) = 0.22$, $P_2(Grass|I_Q) = 0.155$, $P_2(Lawn|I_Q) = 0.155$, and $P_2(Running|I_Q) = 0.03$. Specifically, the weight of the concept *Dog* is 1.975, that of *Sward* is 0.774, and that of *Sport* is 0.27. The idea behind the enhanced model is that if keyword K_j's weight $W_j^{(1)}$ is low, or even zero, but K_j is highly relevant to the concept of its similar images, the ranking of K_j should be higher, e.g., $P_2(Pet|I_Q)$ and $P_2(Lawn|I_Q)$. On the other hand, a keyword will be ignored if it is irrelevant to the concepts of the similar images, e.g., the keyword *Running* and its concept *Sport*. Model P_2 prefers to raise the rankings of the keywords that have the same concepts as the image.

One possible drawback of the method is that it may carry other noisy information into the annotation process. Annotation accuracy relies on the performance of similarity estimation between images and the performance of relevance estimation between keywords. Therefore, linear combination of the two models would help to support a more flexible and reliable method. The solution to the image annotation problem is computed as

$$P(K_j|I_Q) = \omega \times P_1(K_j|I_Q) + (1-\omega) \times P_2(K_j|I_Q) , \qquad (5)$$

where ω is a weighting parameter between 0 and 1. Given image I_Q, function $P(K_j|I_Q)$ serves as the ranking algorithm for each keyword $K_j \in K$.

5 Implementation issues

We have developed a prototype system based on the proposed probabilistic models P_1 and P_2 for image annotation. In this section, we present two major implementation issues: (1) searching for similar images and (2) automatically grouping keywords into concepts. Figure 4 shows our system architecture, which consists of three main modules: *image similarity*, *keyword categorization*, and *keyword ranking*.

In the image similarity module, the *feature extraction* function produces a feature vector for each visual feature of the input image based on image processing technologies. The features considered here include colors and textures for global and local features. The *similarity measure* function calculates similarity α_{Qi} for each image $I_i \in I$ by estimating the weighted sum of the Euclidean distances among the extracted features. Details of visual feature extraction are described in Sect. 5.1.

In the keyword categorization module, the *feature extraction* function generates a feature vector for each keyword by gathering the statistics of documents retrieved from a corpus. These feature vectors are then passed to the *keyword-clustering* function, which generates a set of keyword categories (or concepts) by clustering the keywords based on the similarity between corresponding feature vectors. The centroid feature vector of each concept is regarded as the feature vector of the concept. Weight γ_{jk} is determined by the similarity between the feature vectors of keyword K_j and concept C_K. Details of keyword-feature extraction and keyword clustering are described in Sects. 5.2 and 5.3, respectively.

The keyword ranking module accepts the weights necessary for the semantic network and estimates the relevance degree of each keyword K_j to image I_Q based on the probabilistic models P_1 and P_2. If a keyword's relevance degree is larger than a given threshold, it will be selected to annotate image I_Q.

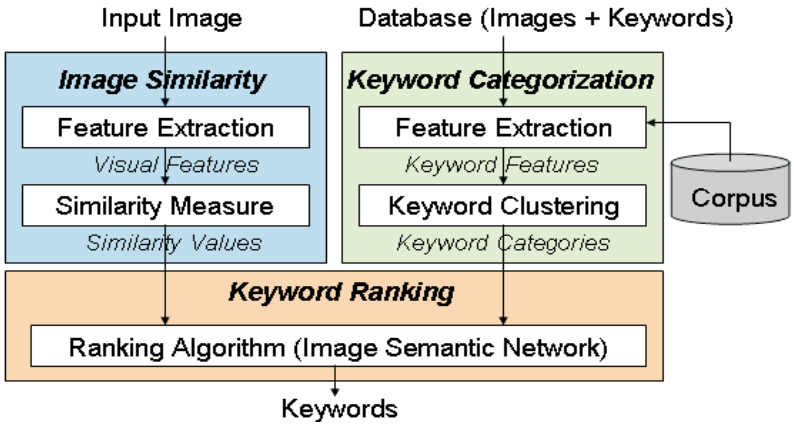


Fig. 4. System architecture of the prototype system

5.1 Visual feature extraction

We employ both colors and textures for global and local image features. Local features representing the spatial layout of the colors and textures in an image are useful for providing object-level visual information. Given an image, we first rescale it to the 256×256 resolution and divide it into 256 16×16 square blocks. Then color and texture features are calculated for each block and the whole image.

For the color features, we make use of a palette of 166 colors by uniformly quantizing the cylindrical HSV color space into 18 hues, 3 saturations, and 3 values, augmented by 4 gray levels. The quantized color histogram emphasizes hue because hue is more sensitive to human perception than the others. Then we represent the histogram as a feature vector with 166 dimensions. The value of each feature is varied from 0 to 1.

For the texture features, we apply symmetric Gabor filters [12] with 3 scales and 4 orientations defined in the spatial domain. The Gabor filters are defined as

$$f_{mn}(x,y) = \frac{1}{2\pi\alpha_m^2} e^{-(x^2+y)/2\alpha_m^2} \times \cos(2\pi(u_{0m}x\cos\theta_n + u_{0m}y\sin\theta_n)), \quad (6)$$

where m indexes filter scales, n indexes their orientations, and u_{0m} denotes the center frequency. The half peak radial bandwidth is chosen to be one octave, which determines α_m. The highest center frequency is chosen as $u_{01} = 0.5$, and $u_{0m+1} = u_{0m}/2$. The four orientations are $\theta_0 = 0$ and $\theta_{n+1} = \theta_n + \pi/4$. The resultant bank of 12 filters has satisfactory coverage over the frequency domain. The mean energy of each filter is computed and quantized into 10 levels, varied from 0 to 1. Then the 12 filters are represented as a feature vector with 12 dimensions. Each dimension corresponds to one quantized level.

5.2 Keyword-feature extraction

Given a set of keywords K, keyword-feature extraction aims to create an N-dimensional feature space with term vocabulary $S = \{t_1, t_2, \ldots, t_N\}$ and then generate a feature vector $f = \langle a_1, a_2, \ldots, a_N \rangle$ for each keyword $K_j \in K$. An extra training corpus is required to examine the statistics of term occurrence and co-occurrence. If a term frequently appears in the corpus with two keywords, the two keywords may be relevant to each other (because they have common co-occurring context terms). Note that the corpus should reflect user knowledge about various applications. For instance, the labeled images in the National Palace Museum provide information about historical artifacts, and the Web provides abundant information about proper nouns such as personal and company names. Herein we take the Web as an example and adopt *Google* (`http://www.google.com`) as our backend search engine for providing the corpus. To obtain term vocabulary S, each keyword $K_j \in K$ is submitted to Google and the top 200 most relevant search results including titles and page descriptions are returned. The search results of keyword K_j can be treated as a document corresponding to K_j. The training corpus collects all $|K|$ documents for all keywords. We then use character/word bi- and trigrams together to extract feature terms from the corpus. Bi- and trigrams denote consecutive two and three words/characters in the corpus, respectively. The top N most frequent feature terms are chosen as our term vocabulary S.

Suppose there exists a vector $\vec{t_i}$ for each feature term t_i. Keyword K_j can be represented as $\vec{K_j} = \sum_{i=1\ldots N} a_i \times \vec{t}$. a_i is defined with a *tf-idf* term weighting scheme [19] and is computed by:

$$a_i = \left(0.5 + 0.5\frac{tf_i}{\max_j ft_i}\right) \log\frac{n}{n_j}, \quad (7)$$

where tf_i is the number of occurrences of term t_i in the document corresponding to keyword K_j, n denotes the total number of documents in the corpus, and n_i is the number of documents containing t_i in the corpus.

5.3 Keyword-clustering algorithm

To cluster keywords K, we need to judge the similarity between two keywords K_1 and K_2 with feature vectors f_1 and f_2, respectively. The similarity between keywords K_1 and K_2 is defined as the cosine measure, i.e., $Sim(K_1, K_2) = cos(f_1, f_2)$.

Algorithm *ConceptGeneration* (K: keywords, δ: threshold)
  $CS \leftarrow \emptyset$
  for each keyword $K_j K$ **do**
    $CS \leftarrow CS \cup \{\{K_j\}\}$
  $L_{i-1}, L_i \leftarrow CS$
  while $|CS| \geq 2$ & $InterLevelDist(L_i, L_{i-1}) < \delta$ **do begin**
    Select two clusters C_1 and C_2 from CS such that $FarthestNeighborDist(C_1, C_2)$ is minimum
    $L_{i-1} \leftarrow CS$
    $CS \leftarrow CS - \{C_1\} - \{C_2\}$
    $CS \leftarrow CS \cup \{\{C_1 \cup C_2\}\}$
    $L_i \leftarrow CS$
  end
  return L_{i-1}.

We extend a hierarchical agglomerative clustering method to group keywords into concepts C. The algorithm *ConceptGeneration* starts with trivial clusters, each containing one keyword. A cluster in the clustering algorithm corresponds to a concept in C and vice versa. The algorithm repeats a loop in which the two "closest clusters" are merged into one cluster. The distance between two clusters C_1 and C_2 is defined as the maximum of the distances between all possible pairs of keywords in the two clusters (the complete-linkage method) and is

computed by

$$FarthestNeighborDist(C_1, C_2) = \underset{K1 \in C1, K2 \in C2}{Max} 1 - Sim(K_1, K_2)\,. \quad (8)$$

The *FarthestNeighborDist* function typically identifies compact clusters in which keywords are very similar to each other and is less affected by the presence of noise or outliers in the data. Its major disadvantage is that it may be biased in favor of small compact clusters. Each loop will reduce the number of clusters by 1 and is repeated until one global cluster is reached or an interlevel's distance exceeds a given threshold δ. C_k's size is determined by $FarthestNeighborDist(C_k, C_k)$.

The *ConceptGeneration* algorithm determines if there exists a suitable partition in each cycle based on the following principle: If merging two closest clusters maximizes the ratio of the change of the intracluster's distance to the change of the intercluster's distance, a cluster may not be cohesive and isolated from the other clusters; therefore, a partition may occur. The distance between two levels L_1 and L_2 is computed by

$$InterLevelDist(L_i, L_{i-1}) = \frac{IntraClusterDist(L_i)\,IntraClusterDist(L_{i-1})}{InterClusterDist(L_i)\,InterClusterDist(L_{i-1})}\,, \quad (9)$$

where

$$IntraClusterDist(L_i) = \frac{1}{|L_i|} \sum_{\forall Ci \in Li} FarthestNeighborDist(C_i, C_j)\,, \quad (10)$$

$$InterClusterDist(L_i) = \frac{2}{|L_i| \cdot (|L_i| - 1)} \times \sum_{\forall Ci, Cj \in Li, Ci \neq Cj} NearestNeighborDist(C_i, C_j) \quad (11)$$

and

$$NearestNeighborDist(C_i, C_j) = \underset{k1 \in C1, k2 \in C2}{Min} 1 - Sim(K_1, K_2)\,. \quad (12)$$

Herein, an intercluster's distance is defined by the minimum distance between all possible pairs of keywords of two clusters, while an intracluster's distance is determined by the maximum distance between all possible pairs of keywords in a cluster. Figure 5 illustrates an example of the changes of intercluster, intracluster, and interlevel distances over clustering iterations for 180 query terms from the Web image search engine *PCHome* (`http://image.pchome.com.tw`). Partition occurs when an interlevel's distance is large enough, e.g., the peaks in Fig. 5.

6 Performance evaluation

In this section, we will examine the performance of the two proposed models P_1 and P_2, their performance for various datasets, and the performance of the ConceptGeneration algorithm described in Sect. 5.3. We will also give an example to show how relevance feedback affects the annotation accuracy.

6.1 Dataset

We tested our system on the Corel library using 5000 images to verify the efficiency of the proposed probabilistic model. The 5000 images were manually partitioned into five collections (*animal*, *natural scene, building*, *weather*, and *transportation*) and labeled manually by experts, in advance, with 20 140 keywords, including 1432 distinct keywords. On average, each image had 4.03 keywords

Table 1. Statistical information about the dataset

Collection Number of	Animal	Natural Scene	Building	Weather	Transportation
Images	1700	1300	500	500	1000
Keywords	6669	5221	1994	2008	4054
Distinct Keywords	556	406	267	160	386
Keywords Per Image	3.92	4.02	3.99	4.02	4.05

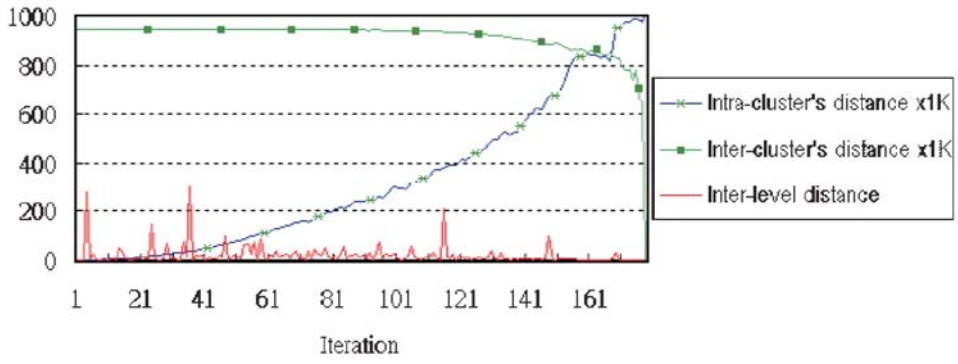


Fig. 5. Example of the interlevel distances for 180 image query terms

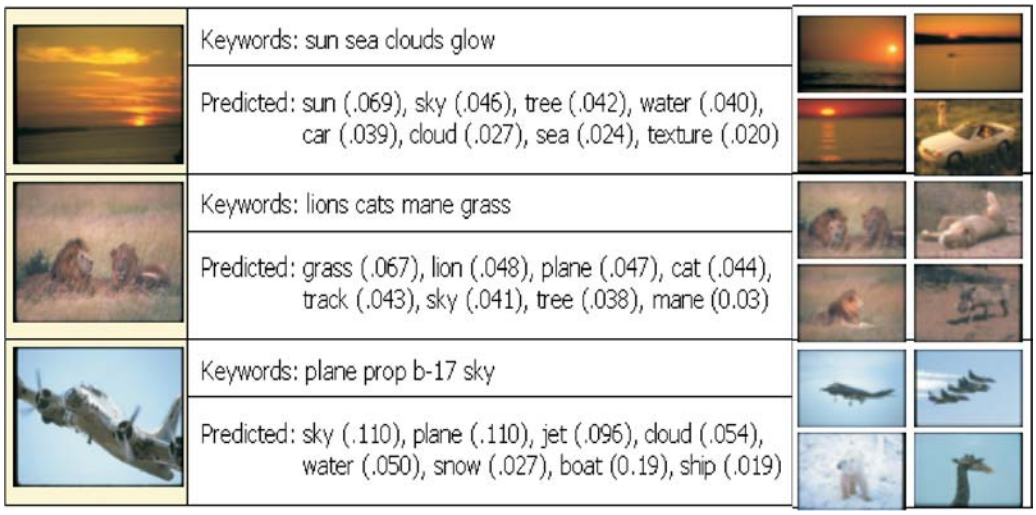


Fig. 6. Some annotation results showing the original images (sunset, lion, and plane) to be labeled, the Corel keywords, the predicted keywords (using model P_1) in rank order based on their scores, and the top-ranked 4 similar images

to describe it. Table 1 shows the statistical information about the five collections, while some sample images in the collections are illustrated in Fig. 6.

6.2 Experiment on image annotation with different models

The goal of this experiment was to examine the performance of the proposed model in recall and precision and then explore its performance with insufficiently labeled keywords. We used the Corel library as our corpus for clustering 1432 distinct keywords. For each collection, 100 images were randomly selected as new (or test) images; the others were treated as labeled (or database) images. Initially, if image I_i was labeled by keyword K_j, its weight β_{ij} was set at 1 by default. Table 2 shows the experimental results of applying the basic probabilistic model P_1 to labeling the images in the five collections. We represented the annotation performance in the form of "recall/precision" for each top-k keywords, where k was varied from 1 to 8. We did not optimize the weights of the visual features for computing image similarity. The proposed probabilistic model can achieve 54.6% recall [i.e., $(0.45 \times 1700 + 0.54 \times 1300 + 0.41 \times 500 + 0.58 \times 500 + 0.77 \times 1000)/5000 \times 100\%$] and 55.1% precision on average when the "top 4" set is selected for labeling. This result indicates that the proposed model P_1 is effective for annotating images when there are sufficient images with good keywords in the collection. We choose the "top 4" set because, on average, each image is expected to be assigned four keywords in the database. When the "top 8" set is selected, it can achieve 68.6% recall with 34.2% precision on average.

To further assess the impact of insufficiently labeled keywords on the basic probabilistic model, we restricted the number of the similar images to a new image to 10 (maximum), while 30% of the labeled keywords in the database were ignored for labeling the new images. We selected the *natural scene* collection as the dataset in this case. Table 3 shows that model P_2 outperforms P_1 when there are insufficiently labeled keywords in the database. Although model P_2 cannot fully meet the performance P_1 achieved with sufficiently labeled data, as shown in Table 2, it does not depend heavily on the quality of the labeled dataset.

The sample results in Fig. 6 show that the proposed approach is dependent on the occurrences and co-occurrences of the keywords associated with the images similar to the given images. Two factors significantly affect performance. The first is the accuracy of the re-

Table 2. Performance of image annotation for five collections (recall/precision) using model P_1

Category Top-K	Animal	Natural Scene	Building	Weather	Transportation
Top 1	0.15/0.60	0.20/0.86	0.13/0.55	0.23/0.81	0.25/1.00
Top 2	0.25/0.50	0.32/0.64	0.21/0.47	0.35/0.63	0.43/0.90
Top 4	0.45/0.45	0.54/0.54	0.41/0.42	0.58/0.56	0.77/0.80
Top 8	0.60/0.30	0.68/0.34	0.59/0.27	0.72/0.35	0.87/0.45

Table 3. Performance comparison between models P_1 and P_2

Model Top-K	P_1	P_2
Top 1	0.12/0.57	0.18/0.71
Top 2	0.22/0.51	0.32/0.64
Top 4	0.38/0.39	0.50/0.50
Top 8	0.57/0.21	0.64/0.32

trieved similar images. Dissimilar images may contribute noisy keywords if they are very similar to the images to be labeled, such as the predicted keyword *car* for the *sunset* image and *snow* for the *plane* image. This problem can be alleviated by improving pattern recognition or relevance feedback technologies. In Sect. 6.4, we will discuss how the system uses a relevance feedback method to retrieve more similar images.

The second factor is the occurrences and co-occurrences of appropriate keywords. Similar images may contribute useful keywords if they are statistically significant. For example, in the *sunset* images the keywords *sun*, *sky*, *cloud*, and *sea* often appear. Although some keywords like *sky* and *water* are not selected by the Corel library, they are still suitable for the description of the *sunset* image in this case. The proposed approach can help annotate a number of keywords that are not easily collected by human experts. In addition, in the example of the *plane* image we can observe that it is very difficult for our system to judge if the blue color stands for *sky* or *water*.

6.3 Experiment on image annotation with various datasets

Since finding keywords based on similar images is affected by the number of manually labeled images and the number of similar images, we further examine which factor is more important in image annotation. The goal of this experiment was to examine the performance of the proposed model in recall and precision under different circumstances.

First, we randomly selected 300 images from the five collections as testing images. For each test image, we reduced the number of labeled images (except the test one) by 20 to 80%. Annotation performance based on model P_1 in recall and precision was reported for each top-K generated keywords, where K was varied from 1 to 10. The experimental results in Figs. 7 and 8 show that uniformly reducing the number of keywords makes the data-sparseness problem more serious. Consequently, many useful keywords that do

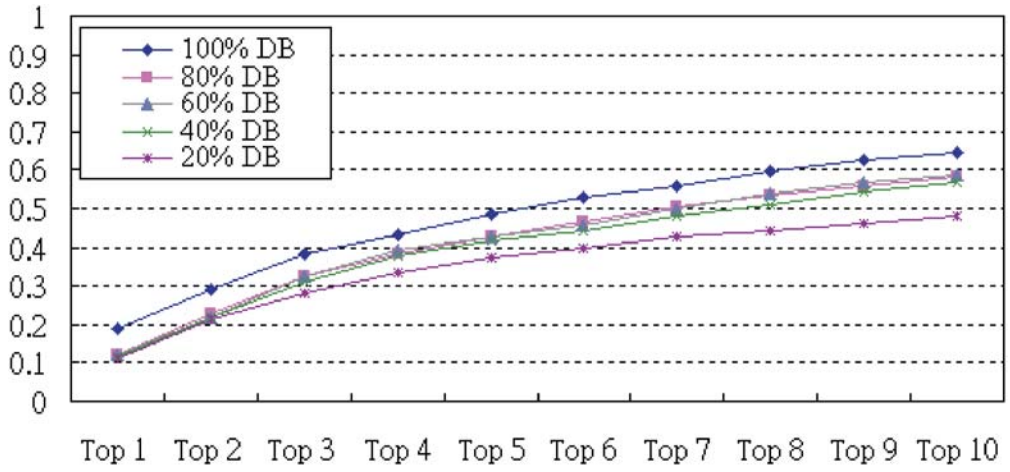


Fig. 7. Recall on different sizes of the labeled datasets, where X% of labeled keywords in the database remain, i.e., $(100 - X)\%$ are ignored for labeling 300 randomly selected images

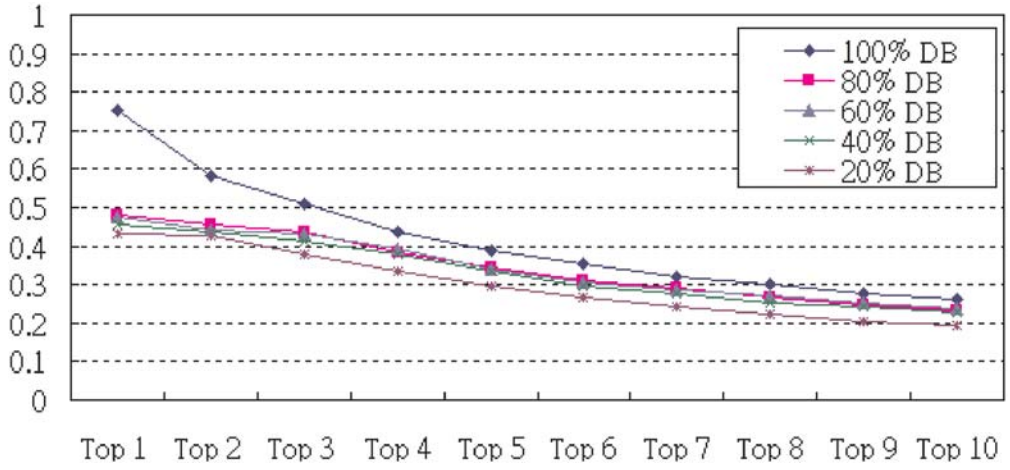


Fig. 8. Precision on different sizes of the labeled datasets, where X% of labeled keywords in the database remain, i.e., $(100 - X)\%$ are ignored for labeling 300 randomly selected images

not reach statistical significance will not be ranked high.

Next, for each testing image from the 300 randomly selected images, we retrieved the top-ranked 500 similar images and then reduced the number of the similar images by 10 to 40%. The experimental results, shown in Figs. 9 and 10, show that the performance drops rapidly, even if only 10% (or 50) of the top-ranked similar images are removed. The main reason is that each image in the collections has only a few similar images. Consider the *animal* collection, which contains 17 types of animals including lions, birds, dogs, cats, horses, butterflies, elephants, fishes, etc. Although each type has 100 images, not all of them are visually similar. Ignoring top-ranked similar images will reduce the probability of occurrences and co-occurrences of effective keywords.

Compared with the previous experiment, the image-similarity problem seems to be more important than the data-sparseness problem. In the proposed model, as shown in Figs. 2 and 3, retrieved similar images in the image level mainly determine whether effective keywords have statistical significance. If semantically relevant images can be discriminated from semantically irrelevant ones in the image level, the data-sparseness problem in the keyword level will be alleviated. Based on the experimental results, it is believed that pattern recognition techniques are still the major challenge in the field of automatic image annotation.

6.4 Experiment on concept generation

Since the proposed model P_2 requires automatic generation of concepts, which consist of sets of semantically relevant keywords, in this experiment we compare how closely clusters generated by our approach match a set of categories previously assigned to a set of keywords by human judges. The image directory of the Web image search engine *Want2* (`http://www.want2.com.tw`) served as

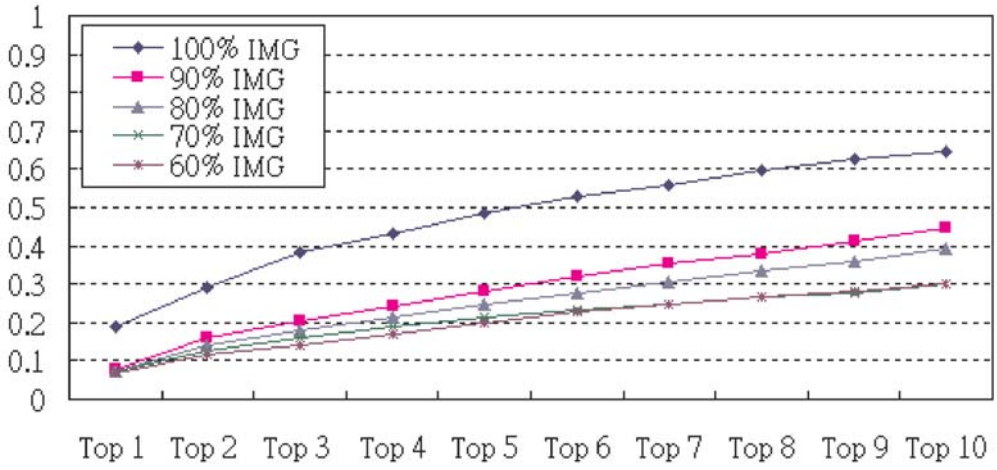


Fig. 9. Recall on different sizes of retrieved similar images, where X% of retrieved similar images in the database remain, i.e., $(100 - X)\%$ are ignored for labeling 300 randomly selected images

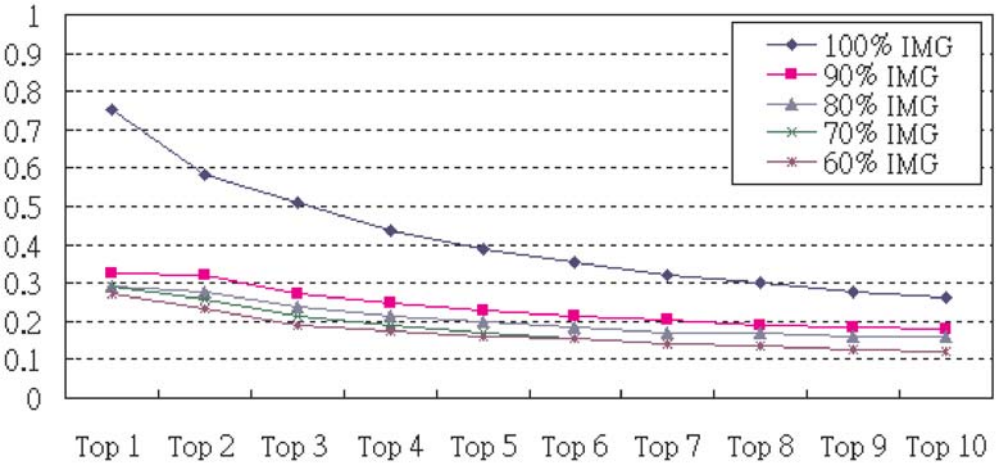


Fig. 10. Precision on different sizes of retrieved similar images, where X% of retrieved similar images in the database remain, i.e., $(100 - X)\%$ are ignored for labeling 300 randomly selected images

our benchmark, in which 95 313 quality Web images from Mainland China were manually classified into 12 main classes such as entertainment, art, computer, nature, consumption, and animation. There were 2712 subclasses in total. We randomly selected 90 subclasses from the 12 main classes and 1000 keywords from the selected subclasses as our dataset.

The F-measure [10], a combination of recall and precision in a single efficiency measure, was adopted as the performance metric. Figure 11 shows the experimental results where the number of generated clusters is varied from 20 to 300 and term vocabulary size N is set to 2000. The F-measure of the generated clusters could achieve 0.61 when the number of clusters was around 160. This explains why the complete-link method prefers to identify compact clusters in which keywords are very relevant to each other. Increasing the number of clusters (or reducing the threshold δ used in the ConceptGeneration algorithm) will produce smaller clusters with closely related keywords that improve precision; however, it might reduce recall performance. Table 4 illustrates two generated clusters corresponding to *Want2*'s categories *Leisure/ Sports/ Tennis/ Stars* and *Leisure/ Travel/ Europe/ France*.

We further compare four different methods of computing the similarity between clusters, including the single-linkage, complete-linkage, group-average-linkage, and centroid methods, under different term vocabulary sizes.

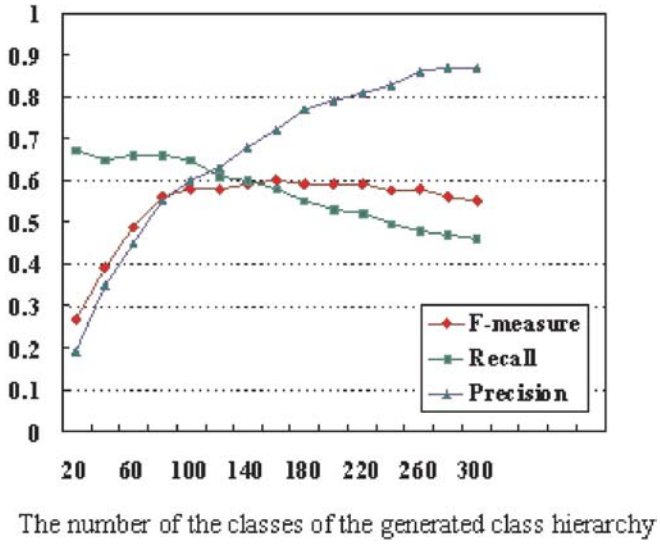


Fig. 11. The F-measure of the generated clusters

The resulting F-measures are shown in Table 5, where the number of clusters is fixed at 160. The experimental results show that the complete-linkage and group-average-linkage methods perform much better than the single-linkage and centroid ones in maximizing the F-measure. The main reason is that in the single-linkage and centroid methods it is very easy to generate large clusters with loosely relevant keywords in real data.

Table 4. Examples of two generated clusters

Cluster (Sports/Tennis) Precision: (1.000000), Recall: (0.785714) F-measure: (0.880000) Want2: (Leisure/Sports/Tennis/Stars)	Cluster (Travel/France) Precision: (0.705882), Recall: (0.705882) F-measure: (0.705882) Want2: (Leisure/Travel/Europe/France)
網球(Tennis) 卡普莉雅蒂(Capriati) 伊凡尼塞維奇(Ivanisevic) 艾芙特(Evert) 辛吉絲(Hingis) 阿格西(Agassi) 桑普拉斯(Sampras) 張德培(Chang,Michael) 莎芭提妮(Sabatini) 莎莉絲(Seles) 葛拉芙(Graf)	凡爾賽宮(Versailles) 巴黎歌劇院 (Opera House of Paris) 巴黎鐵塔(Paris Tower) 艾菲爾鐵塔(Eiffel Tower) 里昂(Lyon) 耶舒依登教堂(Jesvit Church) 香榭大道(Uiric De Varens) 納沃納廣場(Piazza Navona) 馬德蓮教堂(Sainte Marie Madeleine) 梵諦岡(Vatican) 凱旋門(Arc de Triomphe) 勝利女神(Goddess of Victory) 喬托鐘塔(Campanile di Giotto) 塞納河畔(Seine) 聖母之花大教堂(Santa Maria del Fiore) 聖母院(Notre Dame) 羅浮宮(Palais du Louvre)

Table 5. The F-measure under various term vocabulary sizes $|S|$ and similarity functions

Method \ $\|S\|$	50	100	500	1000
CL	0.3838	0.4610	0.5203	0.5408
SL	0.1321	0.1301	0.1654	0.2210
GA	0.3766	0.4563	0.4713	0.4604
CE	0.1207	0.1698	0.2419	0.2338

CL: complete-linkage SL: single-linkage
GA: group-average CE: centroid

6.5 Experiment on image annotation with relevance feedback

The proposed probabilistic model has been integrated with relevance feedback strategies [17]. Suppose a user plans to annotate the image in Fig. 12. First, the system returns a set of similar images stored in the databases. If the retrieved images are similar to the given one, the proposed models will be applied well. Otherwise, for some of the retrieved images, the user needs to mark them as *relevant* or *irrelevant* according to his subjective perception. The system will adjust the weights of different visual features based on the given relevant/positive or irrelevant/negative images. Thus a set of more similar images may be returned and used for generating effective keywords for the given image later. In other words, the system can be improved by better visual feature extraction for specific applications. Figures 13 and 14 show the results of image retrieval and image annotation using relevance feedback. After relevance feedback is performed, the retrieved images are more semantically relevant to the given image. As a result, some irrelevant keywords, such as *sand*, *beach*, *picnic*, and *food*, will be eliminated for both models.

The experimental results show that the proposed model is not restricted by specific pattern recognition technologies. With the development of these technologies, the proposed model could produce more accurate annotated keywords for given images.

7 Discussion

Although manual annotation of images provides high-quality results, it has several fundamental drawbacks.

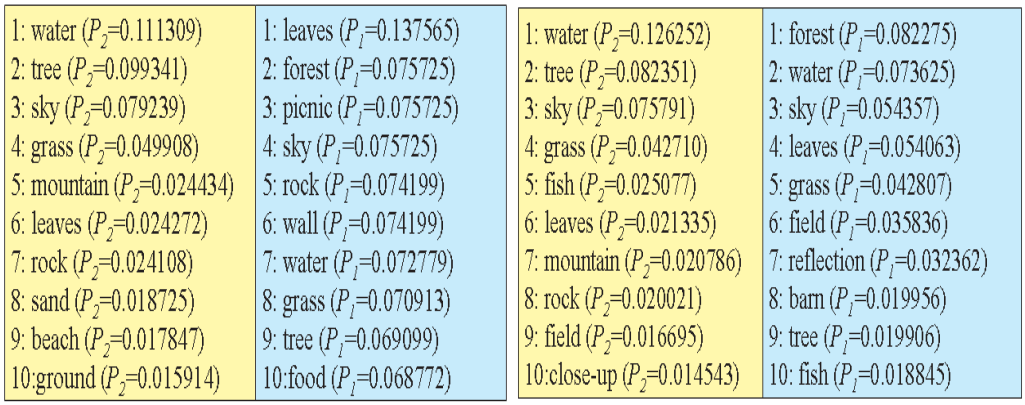
Manual annotation with keywords:

Forest, tree, water, sky, reflection, leave, grass

Fig. 12. Example of an image to be annotated

(a) Similar images before user's feedback (b) Similar images after user's feedback

Fig. 13. Examples of similar images using relevance feedback



(a) Annotation before user's feedback (b) Annotation after user's feedback

Fig. 14. Examples of image annotation with relevance feedback (P_1 & P_2)

First, it is extremely labor intensive, time consuming, and unscalable in the face of ever-growing image digital libraries. Second, it can easily lead to incomplete annotations, but this problem can be alleviated with our approach. However, the issue is more serious when numerous candidate keywords are considered or the number of images to be labeled and the candidate keywords increase. Consider the example in Fig. 6. The proposed approach can produce appropriate keywords not appearing in users' annotations such as *sky* and *water* in the *sunset* image. Those generated keywords are not easily collected by human experts.

Third, manual annotation may lead to inconsistent annotation results. Different annotators may emphasize different aspects of the same images or use different vocabulary to label the same images. Since the proposed approach can deal with large vocabularies, it offers a universal method for describing image content. In addition, if users' query terms that are logged in digital libraries can be collected, the proposed approach would provide a way to label images with them at the concept level. It would also prevent the problem of the same semantics being represented by different vocabulary, which causes unrecoverable mismatches in later retrieval processes. Finally, manual annotation usually requires auxiliary tools such as ontology [6, 21], which might not be available for certain domains. The proposed approach can automatically cluster semantically relevant keywords and may provide some suggestions for labeling images. For example, suppose a given image has many similar images pertaining to tennis games. If some of them contain players' names, the proposed approach may guess that the given image is also about a tennis game and suggest possible players' names for annotation, as illustrated in Table 4.

8 Conclusion

Much attention [20, 22] has been devoted to automatic image understanding based on accompanying texts, but not all images have such textual information. On the other hand, using low-level visual features to endow images with meaning is very difficult since the visual features of semantically irrelevant images may be similar.

In this work, we have proposed two probabilistic models, P_1 and P_2, to estimate the possibility of annotating keywords for given new images based on existing content-based image retrieval techniques and statistical analysis. Its annotation accuracy can therefore be improved with the advance of these techniques. With the models, the annotated keywords of new images are defined in terms of their similarity at different semantic levels. A prototype system based on the models has been implemented. Experimental results show that model P_1 is effective for annotating images when there are sufficient images with high-quality keywords in a collection. And model P_2 does not depend strongly on the quality of the labeled data. The obtained experimental results also show the possibility of the proposed automatic approach to annotating image in digital libraries.

Further work should be focused on determining the threshold used in the ConceptGeneration algorithm in an automatic way. The threshold will affect computational time and storage space.

References

1. Barnard K, Forsyth D (2001) Learning the semantics of words and pictures. In: Proc. international conference on computer vision 2, pp 408–415
2. Chang S-F, Chen W, Sundaram H (1998) Semantic visual templates: linking visual features to semantics. In: Proc. international conference on image processing, workshop on content based video search and retrieval, pp 531–535
3. Chang S-F, Smith JR, Beigi M, Benitez A (1997) Visual information retrieval from large distributed online repositories. Commun ACM 40(12):63–71
4. Cheng P-J, Chien L-F (2003) Auto-generation of topic hierarchies for Web images from users' perspectives. In: Proc. ACM CIKM international conference on information and knowledge management, pp 544–547
5. Cheng P-J, Chien L-F (2004) Effective image annotation for search using multi-level semantics. In: Proc. Asian Digital Libraries, pp 230–242

6. Dillon C, Caelli T (1998) Learning image annotation: the CITE system. J Comput Vis Res 1(2):
7. Flickner M, Sawhney HS, Ashley J, Huang Q, Dom B, Gorkani M, Hafner J, Lee D, Petkovic D, Steele D, Yanker P (1995) Query by image and video content: the QBIC system. IEEE Comput 28(9):23–32
8. Gevers T, Aldershoff F, Smeulders AWM (1999) Classification of images on Internet by visual and textual information. In: Proc. SPIE Internet Imaging 3964, pp 16–27
9. Gupta A, Jain R (1997) Visual information retrieval. Commun ACM 40(5):71–79
10. Larsen B, Aone C (1999) Fast and effective text mining using linear-time document clustering. In: Proc. ACM SIGKDD international conference on knowledge discovery and data mining, pp 16–22
11. Lu Y, Hu C-H, Zhu X-Q, Zhang H-J, Yang Q (2000) A unified framework for semantics and feature based relevant feedback in image retrieval systems. In: Proc. ACM international conference on multimedia, pp 31–37
12. Ma W, Manjunath B (1996) Texture features and learning similarity. In: Proc. IEEE conference on computer vision and pattern recognition, pp 425–430
13. Minka TP, Picard RW (1996) Interactive learning using a "society of models." In: Proc. IEEE conference on computer vision and pattern recognition
14. Paek S, Sable CL, Hatzivassiloglou V, Jaimes A, Schiffman BH, Chang S-F, McKeown KR (1999) Integration of visual and text-based approaches for the content labeling and classification of photographs. In: Proc. ACM SIGIR workshop on multimedia indexing and retrieval
15. Pentland AP, Picard RW, Sclaroff S (1996) Photobook: content-based manipulation of image databases. Int J Comput Vis 18(3):233–254
16. Picard RW, Minka TP (1995) Vision texture for annotation. J Multimedia Syst 3:3–14
17. Rui Y, Huang TS, Ortega M, Mehrotra S (1998) Relevance feedback: a power tool for interactive content-based image retrieval. IEEE Trans Circuits Syst Video Technol 8(5):644–655
18. Rui Y, Huang TS, Chang S-F (1999) Image retrieval: current techniques, promising directions and open issues. J Vis Commun Image Represent 10(1):39–62
19. Salton G, Buckley C (1988) Term weighting approaches in automatic text retrieval. Inf Process Manage 24:513–523
20. Shen HT, Ooi BC, Tan KL (2000) Giving meanings to WWW images. In: Proc. ACM international conference on multimedia, pp 39–47
21. Soo V-W, Lee C-Y, Li C-C, Chen S-L, Chen C-C (2003) Automated semantic annotation and retrieval based on sharable ontology and case-based learning techniques. In: Proc. ACM/IEEE-CS joint conference on digital libraries
22. Srihari RK (1995) Use of multimedia input in automated image annotation and content-based retrieval. Storage and Retrieval for Image and Video Databases (SPIE), pp 249–260
23. Vailaya A, Jain A, Zhang H-J (1998) On image classification: city images vs. landscapes. Pattern Recog 31(12)1921–1935
24. Wang JZ, Li J, Wiederhold G (2001) SIMPLIcity: Semantics-sensitive Integrated Matching for Picture LIbraries. IEEE Trans Pattern Anal Mach Intell 23(9):947–963