Deriving Semantic Terms for Images by Mining the Web

Zhiguo Gong
Faculty of Science and Technology
University of Macau
Macao, P.R.China
fstzgg@umac.mo

Qian Liu
Faculty of Science and Technology
University of Macau
Macao, P.R.China
ma46620@umac.mo

Jingzhi Guo
Faculty of Science and Technology
University of Macau
Macao, P.R.China
jzguo@umac.mo

ABSTRACT

In this paper, we provide a novel image annotation model by mining the Web. In our approach, the concepts or words appearing in the associated text are extracted and filtered as the semantic annotations for the corresponding Web images. In order to alleviate the influence caused by the noise images, for each semantic concept, we improve Web image-word relationships using Mixture Gaussian Distribution Model. By doing so, the concepts or words relevant to any image are re-weighed by both considering their relevance to the image in term of text and in term of visual feature. In fact, all the words associated to an image are not semantically independent. We use co-occurrences between two words to describe their semantic relevance. Thus, we further use a method, called Word Promotion, to co-enhance the weights of all the words associated to a given image based on their co-occurrences. Our experiments are conducted in several ways and the results show that our annotation method can achieve a satisfactory performance.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval Models; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Object Recognition*

General Terms

Algorithms, Design, Experimentation

Keywords

Image Annotation, Web Mining, Mixture Gaussian Model, Word Promotion

1. INTRODUCTION

Today, though advanced computer techniques provide a strong support for digital image analyzing, there is still a gap between image semantics which are interpreted by human beings, and visual features of images which are automatically extracted from images by computers. With the exponential growth of digital images produced nowadays, it is an urgent and strong need to bridge this gap to support image-based applications, such as image retrievals, image understanding, trade mark managements, and digital libraries.

Traditionally, correlations between image visual features and semantic concepts can be created with some learning algorithms. The learning model used is critical for the performance of the annotation work. The existing approaches have provided several models for the learning of correlations between semantic terms and visual features, including LDA model[3], cross-media relevance model[7], 2D HMM[14], translation model[2], co-occurrence model[9] and continuous-space relevance model[8]. And one of the common points of those learning models is to correlate the evidence about the probability of visual features and manually annotated terms. Then, those evidences are used to annotate new images.

Several limitations exist in the past works, including: (1) the predefined term set is limited, without any evolution; (2) the work to label the sample images is labor-intensive, and may be biased with the knowledge of the experts in the specific domain; and (3) the sample images selected may not be enough. Those limitations cause many problems in the traditional image annotations. As a matter of the fact, with the explosive increase of Web information, Web images are becoming one of the most indispensable information representation types on the Web. More importantly, those Web images are associated with rich text descriptions. Therefore, it is intuitive to look for semantics of the Web images from those corresponding associate texts. And the resource can overcome the limitations of the traditional image annotation because (1) term sets can be evolved with the development of the Web; and (2) semantics are based on the interpretations of many different experts and their interpretations can be complementary with each other to remedy the bias of sole domain experts.

Though work[5, 13, 6, 15] are trying to provide some unsupervised learning algorithms, those works still show some limitations as follows: (1)They are designed for refining the results of Google Image Search in some specific area, so they can not be used for universal image annotation problems. (2)Each of them is based on some special image segmentation techniques, which are hard or impossible to be used in other application areas.

In this paper we provide a novel approach: Web-based

ICEC '09, August 12-15, 2009, Taipei, Taiwan

image annotation which is to automatically annotate images by mining the Web. Our system contains two basic stages. In the first stage, semantics and visual features are extracted from the associated texts and the corresponding Web images respectively. Thus, the term-feature correlations are created. However, though huge in amount, Web information may contain large percent of noise terms from the associate texts as well as noise images. Therefore, in the second stage, the key work is trying to adjust the correlation of the Web images and semantic terms by Mixture Gaussian Model and Word Promotion Method respectively.

The remainder of this paper is organized as follows. Section 2 shows detail discussion on semantic extraction and visuals feature extraction techniques of Web images. Section 3 describes our solution to control influences caused by noise using Mixture Gaussian Model and Word Promotion Method. In Section4 our evaluations are presented. And Section 5 gives the conclusion.

2. FEATURE EXTRACTIONS FOR WEB IMAGES

Unlike training images in the traditional model, Web images can naturally bring text and visual features together. For any further processing, effective extractions of both text and visual features are critical.

2.1 Sematic Feature Extraction of Web Images

We firstly partition the associated text of a Web image into three parts as TM, LT, and BT, where TM includes the text elements of **title** and **meta**, LT is the text attached to image command **img** or image links, often highlighting the semantics of Web images, and BT is the text for presentation in **body** element, often providing some descriptions for the embedded images.

We further segment BT into a sequence of text blocks, say $B_0, B_1, ..., B_N$, according to the tag tree structure of BT. With all those $B_0, B_1, ..., B_N$, the recall of Web image retrieval is increased, with a small precision decreasing. Here in this paper, we would like to use Web images as the training images for the annotation model. Thus, the precision is more important. On the other hand, for any topic, large amount of relevant Web images exists in the Web. For those reasons, in this paper we only use B_0 as the text blocks from BT for the Web image sematic extraction.

Let I and T be the set of Web images and the set of words respectively in the Web collection. From above discussions, in order to derive the semantics of a Web image i_j, text blocks from its associated text, such as **TM**, **i-alt**, **i-loc**, **i-name** and B_0, are used as the sources for representing i_j, but with different weights. To simplify the annotations, for Web image i_j, we denote **i-alt**,**i-loc**,**i-name**,B_0 and **TM** as $T_1(i_j)$, $T_2(i_j)$, $T_3(i_j)$, $T_4(i_j)$ and $T_5(i_j)$, respectively. Then, for term $w_k \in T$ and Web image $i_j \in I$, the weight of term w_k associated to Web image i_j is defined as

$$w(w_k, i_j) = \sum_{l=1}^{5} \omega_l * tf(T_l(i_j), w_k) * idf(T_l(i_j), w_k) \quad (1)$$

where $T(i_j) = \bigcup_{l=1}^{5} T_l(i_j)$, ω_l is the weight of T_l, $tf(T_l(i_j), w_k)$ is the term frequency of w_k in text block $T_l(i_j)$, and $idf(T_l(i_j), w_k)$ is the inverted term frequency of w_k in all the text blocks of type T_l. Finally,the conditional probability of term t_k, give Web image i_j is defined as

$$p_T(w_k|i_j) = \frac{w(w_k, i_j)}{\sum_{w_k \in T(i_j)} w(w_k, i_j)} \quad (2)$$

Similarly, the conditional probability of a Web image $i_j \in I$ given a concept $w_k \in T$ is defined as

$$p_T(i_j|w_k) = \frac{w(w_k, i_j)}{\sum_{i_j \in I} w(w_k, i_j)} \quad (3)$$

Both probability $p_T(w_k|i_j)$ and $p_T(i_j|w_k)$ are based on the associated texts of Web images. They are popularly used in Web image retrievals. However, those models do not take into account of the influences of image visual features. In other words, for any concept w_k, many Web images may be relevant in term of equation 3. Some of them are much similar in visual feature while others show diverse in look. We suppose similar images co-enforce with each other to relevant to a concept, and outlier images may represent some noise to the concept. Therefore, we need to enhance the influences of those images with more similar images and reduce the influences of outlier images. To do so, visual features of Web images are need to be extracted.

2.2 Visual Feature Extraction of Web images

In human visual perceptions, texture, color and shape are considered as basic primitives of an image, and they are regarded as the images' features in digital image processing. Those features can be based on whole images or based on the units of images. But, as we discussed above, the technologies to segment images into regions is still an open problem. Therefore, in this paper, the global color feature and global texture feature are utilized.

Color is the basic and most straight-forward characteristic of the image and most extensively used in visual content-based image retrieval. And for color feature, a suitable color space is a crucial factor for the similarity and there are several color spaces, such as HSL, CMY and RGB. In our solution, HSL color space is most suitable because it is tractable and perceptually uniform and easy and possible to transform from popular RGB color space to HSL color space. Further, when extracting color feature, there are several feature forms, including color histogram, color coherence vector[10], color correlogram, color moments and color set[12]. And in Web-based image annotation, the form of color feature is quantized color histogram because it is easy to compute, invariant to rotation and translation and sensitive to noisy interference such as illumination changes and quantization errors. Therefore, in HSL color space, hue is quantized into 18 levels, and lightness and saturation are quantized into 3 levels respectively. And there are 162(18*3*3) colors in HSL color space. In addition, grey color is also quantized into 4 levels. At length, the color histogram is 166-dimension in this work.

Texture is another crucial characteristic to identify the images' content. And it is an innate property of all surfaces and refers to visual patterns of homogeneity. Thus, it is discriminable and popular with image segmentation, image recognition and visual content-based image retrieval. There are several methods to represent and extract the texture, such as Tamura feature and wavelet transform. In our solution, Daubechies wavelet transform is chosen because of its better performance in time and frequency domain. In the process

of wavelet transform, each image is decomposed into four frequencies at each level, diagonal coefficients(HH), vertical coefficients(HL), approximation coefficients(LL), horizontal coefficients(LH). For those frequencies, there are two main methods for continuing wavelet transform, including tree-structure wavelet transform (TWT) and pyramid-structure wavelet transform (PWT). In TWT, the decomposition of high frequency, HH, is unstable, because each frequency is decomposed again. In PWT, the information in HL, LH, HH, is lost, because only LL frequency is decomposed again. Therefore, the composite method is: except HH frequency, each frequency is decomposed again. For each frequency at each level, the mean and variance are used as the components of texture feature vector, which can be described as follow:

$$\overrightarrow{fvt} = \{\frac{\mu_{11}}{\delta_{\mu_{11}}}, \frac{\sigma_{11}}{\delta_{\sigma_{11}}}, \cdots, \frac{\mu_{ij}}{\delta_{\mu_{ij}}}, \frac{\sigma_{ij}}{\delta_{\sigma_{ij}}}, \cdots, \frac{\mu_{NM}}{\delta_{\mu_{NM}}}, \frac{\sigma_{NM}}{\delta_{\sigma_{NM}}}\} \quad (4)$$

In (4), N is the level of the transform and M is the number of frequency of each level and here. In this work, M is set to 4, denoting the number of one approximation frequency and three detail frequencies. μ_{ij} and σ_{ij} is respectively the mean and variance of the frequency j in level i. And $\delta_{\sigma_{ij}}$ and $\delta_{\mu_{ij}}$ are standard deviations of σ_{ij} and μ_{ij} respectively in the entire database. For each image unit, 4-level wavelet transform is performed. Therefore, texture feature is represented as a vector of 320(4*(1+3+9+27)*2)-dimensions.

Up to now, through the techniques of information extraction and digital image processing, term-feature correlation is obtained and can be used as the basic of image annotation. Unfortunately, the correlation contains some noise because there is no standard for the Web images and further it is difficult to use suitable method to extract the semantic. Therefore, the correlation must be improved in order to raise the annotation performance.

3. IMAGE ANNOTATION MODEL

To annotate any image i_x, we need to find some Web images which are close to i_x in visual features. Then, i_x can be annotated using the annotations of those selected Web images. Therefore, effective correlations between visual features and text features of Web images are crucial to the performance of our annotation system.

3.1 Overview of the Annotation System

The working principle of our system is illustrated in Fig. 1. In the figure, sm_j is the distance between the unlabeled image i_x and Web image i_j, which is calculated by Euclidean distance between visual features of the two images. $p_T(w_k|i_j)$ represents the association degree of Web image i_j to concept w_k, and M and N are the number of keywords and Web images respectively in the Web collection. What is more important is how to determine the relevance degree of a concept $w_k \in T$ given an unlabeled image i_x. It is intuitive that we can summarize those Web images which are similar to i_x for the semantics of the unlabeled image i_x.

Let i_j be a Web image, then the associated text for i_j is defined as term-weight vector $v(i_j)$

$$(w_1 : p_T(w_1|i_j), w_2 : p_T(w_2|i_j), ..., w_M : p_T(w_M|i_j)). \quad (5)$$

where $w_k(k = 1toM)$, are terms in T, and $p_T(w_k|i_j)$ are their corresponding relevant degrees as defined in equation 2. Then, for any given image i_x,let $i_1, i_2, ..., i_J$ are the first J Web images close to i_x in the Web collection. The associated text of i_x is summarized from those Web images as

$$sv(i_x) = \sum_{j=1}^{J} \frac{\mathrm{sm}_j}{\sum_{l=1}^{J} \mathrm{sm}_l} * v(i_j). \quad (6)$$

From this equation, it is clear that the effectiveness of such a model is dominated by two points: (1)the correctness of the associated term vectors of the Web images, and (2)enough number of Web images which are close to image i_x in the collection.

However, though the Web provides a huge and comprehensive source for images associated with texts, much noise exists either in letter of Web images or in letter of words in the collection. In other words, for any word w_k, it may associated with many irrelevant Web images. And at the same time, give a Web images, many irrelevant words may appear in its associated text. Therefore, it is necessary to enhance the influences of the correct data and alleviate the influences of the noise data before using the correlation to annotate unlabeled images. In this study, we use two techniques for this sake. For any given word w_k, we cluster all the relevant Web images on the basis of their visual features, then use Gaussian Model to give a visual feature distribution of images for w_k. Then we re-calculate the relevance degrees of Web images to w_k by both taking into account of text-based association degrees and visual distributions. By this way, the relevances of similar web images to a given word w_k can be enforced, the influences of noise images can be reduced. On the other hand, for a given image i_x, we use co-occurrence relationships between terms to improve their relevance degrees to the image.

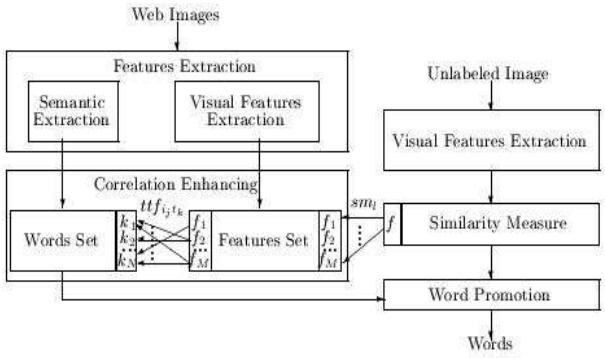

Figure 1: The Principle of the System

3.2 Mixture Gaussian Model

For any given concept or word w_k, associated Web images may give different visual spectra of w_k. Therefore, we can cluster those images and each cluster can be taken as one spectrum of w_k. In this paper, Mixture Gaussian Model[11] is used to describe images' distribution.

For each word $w_k \in T$, let $I(w_k) = \{i_j | p_T(i_j|w_k) > \delta_k\}$ be the set of Web images with high probabilities for w_k. Then, we suppose all Web images in $I(w_k)$ are relevant to concept w_k. It seems that we can take $I(w_k)$ as the training set for image annotation modeling. However, the obvious weakness of such training set, comparing with traditional training set of manually annotated images, lies in that many noise images may be included in $I(w_k)$, and the image qualities are

not consistent. But, we suppose, out of the huge amount of the Web images, qualified positive images are always enough to support our modeling. Thus, the problem is how to reduce the noise images or reduce the influences of them in the modeling. In our approach, two techniques are used, including to reduce noise images with clustering techniques and to reduce the influences of false images with using negative training set. Let $NI(w_k)$ be the set of negative Web images for word w_k. Actually the images are randomly selected from the set $\{i_j | i_j \in I \wedge P_T(i_j|w_k) = 0\}$. Then, the modeling technique **Mixture Gaussian Model** which is based on these two training sets is adopted in this paper.

To construct the Mixture Gaussian Model, both $I(w_k)$ and $NI(w_k)$ are clustered respectively on the basis of the visual similarities. Then the positive probability and negative probability are constructed using Mixture Gaussian Model according to the clusters. Then, the relevance of a image i to w_k is defined as the ratio of the positive density function over the negative density function as

$$R_F(i, w_k) = \frac{P1(i|w_k)}{P2(i|nonw_k)} \tag{7}$$

where $P1(i|w_k)$ and $P2(i|nonw_k)$ are the positive component based on $I(w_k)$, and negative component based on $NI(w_k)$ respectively. And they are defined using Mixture Gaussian Model as

$$EXP_l = exp^{-\frac{1}{2}(i-\mu_{l,j})^T \Sigma_{l,j}^{-1}(i-\mu_{l,j})} \tag{8}$$

$$P1(i|w_k) = \sum_{j=1}^{m_1} \omega_{1,j} \frac{1}{\sqrt{(2\pi)^N |\Sigma_{1,j}|}} EXP_1 \tag{9}$$

$$P2(i|non - w_k) = \sum_{j=1}^{m_2} \omega_{2,j} \frac{1}{\sqrt{(2\pi)^N |\Sigma_{2,j}|}} EXP_2 \tag{10}$$

where i is represented as the visual feature, N is the dimension of the image feature, m_1 is the number of positive components, $\omega_{1,j}$,$\mu_{1,j}$, and $\Sigma_{1,j}$ represent the weight, the mean vector, and the covariant matrix of the j-th positive component, respectively, on the condition $\sum_{j=1}^{m_1} \omega_{1,j} = 1$. Parameters for $P2(non - w_k|i)$ have similar explanations. Work [15] is trying to refine the model by iteratively using text-base classifier (similar to model P_T)and Mixture Gaussian Model(above). But their experiments shows more iterations can not improve the performance of the model.

Like in many traditional works, for any image i (a new image which may not be in the Web image collection I), it seems that $R_F(i, w_k)$ can be used as the annotation model. In other words, we may use ($w_i : R_F(i, w_1)$,$w_k : R_F(i, w_2)$,..., $w_M : R_F(i, w_M)$) as the annotation vectors for image i. With some experiment, however, the performance is much lower. With a close study, we find the problem is due to the lower quality of the training set. Therefore, in our system, $R_F(i, w_k)$ is only used to refine and modify the relevance between existing Web images and words. We modify $p_T(w_k|i_j)$ as

$$R_{TF}(i_j \rightarrow w_k) = p_T(w_k|i_j)(1 + \alpha \bullet \frac{R_F(i_j, w_k)}{Max_{\{l\}}\{R_F(i_l, w_k)\}}) \tag{11}$$

where α is constant used to maximize the overall performance with using $R_{TF}(i_j \rightarrow w_k)$.

The principle behind this model is based on our following observation: (1) The relevance between a Web image i_j and a term w_k is mainly dominated by the associated text of the Web image, which is described with $p_T(w_k|i_j)$; (2)$p_T(w_k|i_j)$ does not take into account of the co-enforcements among similar images related to the same word.

Therefore, in our consideration, $R_F(i_j, w_k)$ is used to enhance a Web image's relevance to a word if there exist more similar images relevant to the same words. Reversely, if a Web images that has less similar Web images in supporting the same word, its relevance to the word will be decreased relatively. With $R_{TF}(i_j \rightarrow w_k)$, the associated text for Web image i_j can be modified into $v_F(i_j)$ as

$$(w_1 : R_{TF}(i_j \rightarrow w_1), ..., w_M : R_{TF}(i_j \rightarrow w_M)). \tag{12}$$

Comparing with $v(i_j)$, $v_F(i_j)$ has refined the weights of terms using feature re-enforcements among Web images.

To annotate any image i_x,we select Web images, say, i_1, i_2, ..., i_J, such that $||i_j - i_x|| < \varepsilon$. An annotated term vector $sv_F(i_x)$ is computed as

$$sv_F(i_x) = \sum_{l=1}^{J} \frac{\mathrm{sm}_l}{\sum_{j=1}^{J} \mathrm{sm}_j} * v_F(i_j). \tag{13}$$

Generally, $sv_F(i_x)$ is a long list of weighted words. To annotate image i_x, users often prefer one or limited words which can stand for the semantics of i_x. For this sake, the first several words ranked with their weights are selected as the potential annotations for this image.

3.3 Word Promotion Method

Up to now, we didn't take into account of the semantic relevance of the words in annotating image i_x. As matter of the facts, many words in $sv_F(i_x)$ are closely related in semantic. Thus, we need to re-weight the words according to their semantic relevances. To reach this objective, we propose *Word Promotion Method* with respect to the semantic relevances among terms.

We define term semantic relevances using the concepts *Support* and *Confidence* between any two words. For any two words w_i and w_j, *Support* and *Confidence* between them are defined respectively as

$$Sup(w_i, w_j) = \frac{||D(w_i) \bigcap D(w_j)||}{||D||} \tag{14}$$

$$Conf(w_i, w_j) = \frac{||D(w_i) \bigcap D(w_j)||}{||D(w_i)||} \tag{15}$$

where $D(w_i)$ and $D(w_j)$ are the Web documents containing word w_i and w_j respectively, D is the collection of the Web documents. Therefore, $Conf(w_i, w_j)$ indicates the degree of w_i's supporting w_j in semantics, and the value of $Sup(w_i, w_j)$ indicates the stability degree of the confidence. In this paper, we only remain the *Confidence* values with $Sup(w_i, w_j) > 0.001$ and $Conf(w_i, w_j) > 0.1$.

Now, we use $Conf(w_i, w_j)$ to re-calculate the weights of terms in $sv_F(i_x)$ and get the promoted term-vector $\widetilde{sv_F}(i_x)$. For each word w_k, the new weight with respect to i_x is recalculated as

$$\widetilde{sv_F}(i_x)|_{w_k} = \sum_j Conf(w_j, w_k) \bullet sv_F(i_x)|_{w_j} \tag{16}$$

The weight of w_k in $\widetilde{sv_F}(i_x, w_k)$ is not only based on the relevance of w_k to the image i_x, but also the semantic supports from other terms associated to the same image i_x. In

Table 1: MRR of Annotation Performance

	Baseline	MGM	WPM
MRR	0.310053	0.361333	0.393006

fact, $\widetilde{sv_F}(i_x)|_{w_k}$ gives the degree of term w_k as the label to stand for the overall semantics of image i_x.

4. PERFORMANCE EVALUATION

To evaluate the performance of our image annotation system, the crawler of our system has gathered as many as 50000 Web pages from the domain dot.com, and dot.edu. And 12000 Web images are filtered out from the collection after noise images, such as icons, banners, logos and any image with size less than 5k, removed. And there are more than 50000 words within their associated texts. In our experiments, 200 images are chosen for testing and several evaluation methods are conducted.

In [4], Cheng and Chien measure their annotation system using two basic objective functions, precision values and recall values with respect to word positions in the list. Comparing to their system, all terms in our Web collection can be selected to annotate an image. As one of the result, an image may either correctly be annotated with 'laptop' or 'notebook'. We take both situations as correct annotations. Therefore, we manually annotate all testing images with multiple labels and to see if those correct labels can be ranked earlier in the list and if all the possible correct labels can be in the list.

With our system, an unlabeled image can be assigned a long list of words ordered by their relevant weights. Therefore, the position of the first correct label is important. For this sake, Mean Reciprocal Rank (MRR) measure is used in our work. MRR function is defined as

$$MRR = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{n_k} \tag{17}$$

where N is the number of testing images used and n_k is the position of the first correct word in the word list when annotating. And Table 1 is the annotation performance. In Table 1, baseline value is obtained by using the original data to make the annotation, and MGM and WPM denote the improved annotation model by using **Mixture Gaussian Model** only,and both **Mixture Gaussian Model** and **Word Promotion Method** together, respectively. It is clear from Table 1 that the original MRR value is 0.310053 and both MGM and WPM show good improvements with MRR value 0.361333 and 0.393006 respectively.

Though MRR can show us how early we can get a correct word to annotate a image from the list of words, for most situation, we need to select several words together to describe the semantic of an image. For such situation, we need to know how easy it is for us to find more correct words in describing the semantic of an image. Therefore, in our prototype system, beside the evaluation methods in [4], F-measure is also used to determine the optimal number of words in the list to annotate an image on average. Let S be the set of all the correct words for an image, S_1 be the set of the words in the annotation list up to current position, then F-measure is defined in (20).

$$p = \frac{|\{w_j | w_j \in S_1 \wedge w_j \in S\}|}{|S_1|} \tag{18}$$

$$r = \frac{|\{w_j | w_j \in S_1 \wedge w_j \in S\}|}{|S|} \tag{19}$$

$$F_\delta = \frac{(\delta^2+1)pr}{\delta^2 p + r} \tag{20}$$

In our system, $\delta = 1$, that is, harmonic F_1 is used. And the performance is shown in Table 2. From Table 2, F_1 is improved more or less through those processing techniques. The biggest improvement occurs at the position 4. In other words, with Gaussian Mixture Model and Word Promotion Method, the system can provide a good performance if we use the first 4 words to annotate an image.

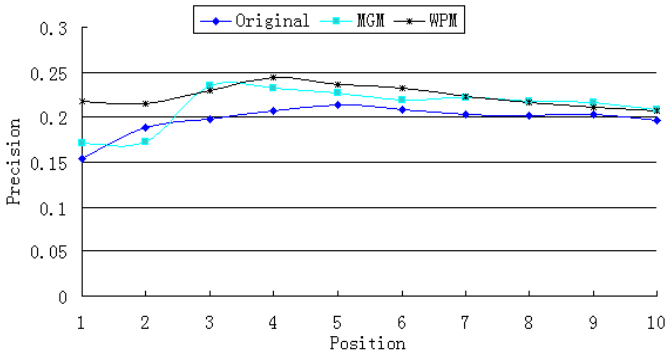

Figure 2: Position&Precision of Annotation Performance

In our prototype system, each image can be described with a set of words, for example, if an image is about "notebook", its description may include other semantically related words besides "notebook". Practically,we hope an annotation system can always correctly annotate an image. In other words, we may not be so interested in if the system can use all the possible correct words to annotate an image, rather we hope it can correctly annotate an image. Therefore, the recall is not so important, while what is more important is how many unlabeled images have been annotated well and whether the

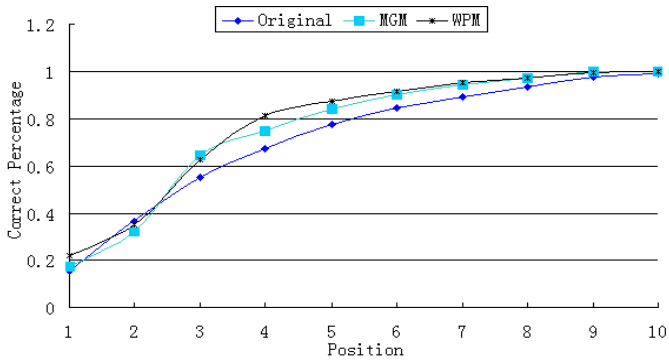

Figure 3: Position&Accuracy of Annotation Performance

Table 2: F1 of Annotation Performance

For Original Data						
Position	1	2	4	6	8	10
Recall	0.044306	0.115761	0.240581	0.360423	0.440349	0.508807
Precision	0.153781	0.214104	0.234494	0.234211	0.220688	0.207477
F1	0.068793	0.150273	0.237499	0.283923	0.294022	0.294759
After MGM						
Position	1	2	4	6	8	10
Recall	0.046152	0.109186	0.272814	0.371075	0.471612	0.535822
Precision	0.171623	0.195837	0.252974	0.240159	0.230353	0.216568
F1	0.072742	0.140204	0.262519	0.291597	0.309523	0.308462
After WPM						
Position	1	2	4	6	8	10
Recall	0.062725	0.122826	0.284057	0.386055	0.470273	0.505651
Precision	0.217502	0.209006	0.262957	0.245823	0.22876	0.223829
F1	0.09737	0.154725	0.2731	0.300378	0.307795	0.310301

precision of the annotation is high or not. According to the discussion, two objectives are used to evaluate the annotation performance in this respect, including precision with respect to word position in the list, and percentage of the images which have been annotated accurately until to the current position in the list. The performance of this evaluation method is in Fig. 2 and 3. From Fig. 2 and 3, both MGM and WPM produce significant improvements especially at position 4.

5. CONCLUSION

In this paper, we propose an automatic image annotation system based on Web images. In our solution, the semantics of Web images are extracted from their associated texts. Then the relationships between words and visual features are constructed accordingly.

In order to improve the performance, we use Mixture Gaussian Model to scale down the influences of noise images for a concept, and use Word Promotion Method to promote significant words in their annotation list for a image. With such techniques, the performance can be improved.

6. ACKNOWLEDGMENTS

This work was supported in part by the University Research Committee under Grant No. RG066/07-08S/ 09R/ GZG/FST and by the Science and Technology Development Found of Macao Government under Grant No. 044/2006/A.

7. REFERENCES

[1] *SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 28 - August 1, 2003, Toronto, Canada*. ACM, 2003.

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

[3] D. M. Blei and M. I. Jordan. Modeling annotated data. In *SIGIR* [1], pages 127–134.

[4] P.-J. Cheng and L.-F. Chien. Effective image annotation for searches using multilevel semantics. *Int. J. on Digital Libraries*, 4(4):258–271, 2004.

[5] H. Feng and T.-S. Chua. A bootstrapping approach to annotating large image collection. In N. Sebe, M. S. Lew, and C. Djeraba, editors, *Multimedia Information Retrieval*, pages 55–62. ACM, 2003.

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

[7] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *SIGIR* [1], pages 119–126.

[8] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *NIPS*. MIT Press, 2003.

[9] Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words, 1999.

[10] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *ACM Multimedia*, pages 65–73, 1996.

[11] S. Russel and P. Norvig. *Artificial Intelligence-A Modern Approach*. Prentice Hall, Inc., Upper Saddle River, New Jersey 07458, USA, 2003.

[12] J. R. Smith and S.-F. Chang. Visualseek: A fully automated content-based image query system. In *ACM Multimedia*, pages 87–98, 1996.

[13] X. Song, C.-Y. Lin, and M.-T. Sun. Autonomous visual model building based on image crawling through internet search engines. In *Multimedia Information Retrieval*, pages 315–322, 2004.

[14] J. Z. Wang and J. Li. Learning-based linguistic indexing of pictures with 2–d mhmms. In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 436–445, New York, NY, USA, 2002. ACM Press.

[15] K. Yanai and K. Barnard. Probabilistic web image gathering. In *MIR '05: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pages 57–64, New York, NY, USA, 2005. ACM Press.