2013 IEEE International Symposium on Multimedia

A Two-Phase Generation Model For Automatic Image Annotation

Liang Xie, Peng Pan*, Yansheng Lu, Shixun Wang, Tong Zhu, Haijiao Xu, Deng Chen
School of Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, China, 430074
whutxl@hotmail.com, panpeng@mail.hust.edu.cn, lys@mail.hust.edu.cn, wsxun@hust.edu.cn,
zhutong@hust.edu.cn, guesskkk@hust.edu.cn, chendeng8899@hust.edu.cn

Abstract—**Automatic image annotation is an important task for multimedia retrieval. By allocating relevant words to un-annotated images, these images can be retrieved in response to textual queries. There are many researches on the problem of image annotation and most of them construct models based on joint probability or posterior probabilities of words. In this paper we estimate the probabilities that words generate the images, and propose a two-phase generation model for the generation procedure. Each word first generates its related words, then these words generate an un-annotated image, and the relation between the words and the un-annotated image is obtained by the probability of the two-phase generation. The textual words usually contain more semantic information than visual content of images, thus the probabilities that words generate images is more reliable than the probability that images generate words. As a result, our model estimates the more reliable probability than other probabilistic methods for image annotation. The other advantage of our model is the relation of words is taken into consideration. The experimental results on Corel 5K and MIR Flickr demonstrate that our model performs better than other previous methods. And two-phase generation which considering word's relation for annotation is better than one-phase generation which only consider the relation between words and images. Moreover, the methods which estimate the generative probability obtain better performance than SVM which estimates the posterior probability.**

Keywords-image annotation; generation model; probability estimation

I. Introduction

During the last decades, with the rapid development of computer science and information technology, multimedia data such as image becomes easy to deliver and access. For most web pages such as news, encyclopedia, etc, images can be able to make the information better to be recognized by people. Large digital image collections are favored by cheap digital recording and storage devices. Images can also be shared on some popular websites such as Flickr. Nowadays, the information world is full of multimedia especially images. However, efficient retrieval for images is required to make the people easy to access their interested images. Traditional image retrieval mainly focus on keyword based retrieval and content based retrieval. In content based image retrieval images are retrieved according to their visual features such as colors, textures and shapes [1]. The advantage of content based image retrieval is that images are not need to be manually labeled. But the disadvantage is that content based retrieval cannot support text query, which means users need an image or other visual query for retrieval. People are usually accustomed to use textual words to retrieve multimedia resource. In addition, visual features can only measure the content similarity of images, content based retrieval cannot be able to measure the semantic similarity which is more close to the understanding of human. Keyword based retrieval has the advantage that users can use some textual words to retrieve images and images contain the same semantic information to the query will be retrieved. But traditional keyword based retrieval system such as Google image, usually requires images to be manually labeled, or images are associated with some textual words, and manually labeling is costly and labor-intensive.

Automatic image annotation takes advantage of existing annotated image dataset to automatically label un-annotated images with semantically related words. They can solve the problem of the costly manually labeling while still retaining the advantage of semantic search, thus they integrate the advantages of content based retrieval and keyword based retrieval. Automatic image annotation has got a lot research in recent years. In the early research, the co-occurrence of images and words are studied and the Co-occurrence Model is proposed to solve the problem of image annotation [2]. Then machine learning methods have got much attention, [3] firstly proposes Translation Model for image annotation, it uses a vocabulary of blobs to describe images and assume that image annotation can be viewed as the task of translating from a vocabulary of blobs to a vocabulary of words. Based on the blob representation for images, several probabilistic generative models are proposed and they obtain good results for image annotation. Recently, most of state-of-the-art image annotation methods are discriminative methods which directly link the visual features of images to words. [4] proposes a baseline method which views image annotation as

* Corresponding Author

978-0-7695-5140-1/13 $26.00 © 2013 IEEE
DOI 10.1109/ISM.2013.33

the nearest neighbor propagation, it uses color and texture features to search the nearest neighbor of the un-annotated image from annotated images in training data. [5] proposes Tagprop which uses 15 visual features including color histograms, Sift, Gist and Hue. Since generative methods and discriminative methods are related to our work, we will discuss them in detail in section II.

In the probabilistic perspective, given an image I and a word w, generative methods usually model the relation of the image and word as the joint probability $p(w,I)$, while discriminative methods model the relation as the posterior probability $p(w|I)$. In fact, for a given image, they are equivalent. In this paper we assume that the relation of words and images can be modeled as generation probability $p(I|w)$, the probability how the word can generate the image. Words usually demonstrate more semantic information than images, the generation from words to images is more reliable than the generation from images to words, thus $p(I|w)$ may be better than $p(w|I)$ and $p(w,I)$ for modeling the relation of images and words. We propose the two-phase generation model (TPGM) which views the image annotation as a two-phase generation procedure. At first each word generates its semantically related words. Then these words are used to generate the un-annotated image. Finally we choose the words with the highest generation probability for the un-annotated image. While estimating our model, we incorporate the classical discriminative method support vector machine (SVM) to our model, thus our model retains the advantage of discriminative methods and it may be more reliable for annotation. Our model outperforms other generative and discriminative methods on the Corel 5K. And we also compare our model to one-phase generation and discriminative SVM to show the advantages of our model on Corel 5K and MIR Flickr. Experimental results confirm that TPGM can improve the effect of image annotation while it is not more complex than general generative and discriminative methods.

This paper is organized as follows. We discuss related work in section II. In section III we describe our two-phase generation model and its estimation, and use it for annotation. Section IV shows experimental results of our model and compares it to other method. Finally, the last section concludes with a discussion of future work in image annotation.

II. Related work

Automatic image annotation has been studied for some years, and many methods have been proposed for image annotation. Most research mainly focus on two types of methods: generative methods and discriminative methods.

Researches on generative methods usually design specific generation models for the joint probability $p(w,I)$ of word w and image I. The joint probability is generally generated by latent variables that encode the hidden states of the world, and the latent variables may be image documents in the training data, semantic topics, etc. Cross-media relevance model (CMRM) uses images in the training data as latent variables to generate the joint probability [6]. CMRM assumes blobs of the image and words are conditional independent over the latent variables, and the probabilities that latent variables (images in the training data) generate blobs and words are based on their occurrences in training images. [7] proposes Continuous-space Relevance Model (CRM) which also adopts images in the training data as latent variables. It assumes that latent variables generate the words by multinomial distribution, and the distribution that generates blobs of images is estimated by a non-parametric kernel-based density estimation. [8] proposes Multiple Bernoulli Relevance Model (MBRM) which is based on CRM. The main difference between MBRM and CRM is that MBRM assumes words are generated by multiple Bernoulli distribution and not the multinomial distribution, and MBRM is shown to be better than CRM on annotation performance. Topic model which has widely used in text analysis is also proven to be suitable for image annotation. Topic model is a generation model which learns semantic topics as latent variables. [9] extends the Latent Dirichlet Allocation (LDA) and proposes a Correlation LDA which relates words and pictures. This model assumes that a Dirichlet distribution can be used to generate a mixture of latent variables. These latent variables are then used to generate words and image regions. EM is used to estimate this model. [10] uses PLSA for image annotation and proposes Asymmetric PLSA. At first normal PLSA is used to learn the latent topics from textual words, then folding-in method is used to relate the latent topics to images. It should be noted that the generative models mentioned before are different to our two-phase generation model. In normal generation models images and words are generated by latent variables and they finally obtained the joint probability $p(w,I)$. In our model we use words to generate images and we finally obtained the generation probability $p(I|w)$ which is completely different to $p(w,I)$.

Unlike generative models which use latent variables for joint probability, discriminative methods directly relate images with words. [11] proposes supervised multiclass labeling (SML), it estimates posterior probability $p(w|I)$ by assuming the distribution of each annotation is a Gaussian mixture, then an extension of EM is used for estimation. Although SML estimates posterior probability, it use the Gaussian mixture which is a generative model, it is not the complete discriminative method. Nearest neighbor method is seen as different from general discriminative model in previous work, but nearest neighbor methods also directly link the images to words and they are widely used for discriminative tasks such as classification, so we think the nearest neighbor method is also a special type of discriminative method. [4] uses nearest neighbor propagation for image annotation. It extracts color and texture features from images and views the image annotation as a kind of k-nearest neighbor classification. The nearest neighbor method outperforms most of the generative models on image annotation, and it is also suitable for large scale data. [12]

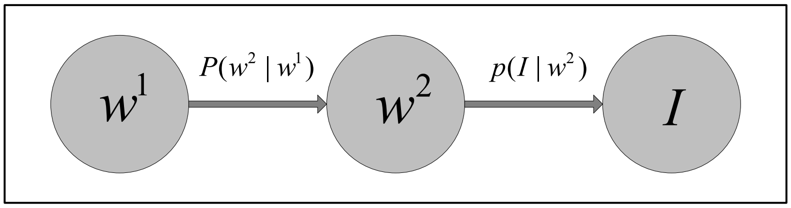


Figure 1. The graphical illustration for TPGM, where w^1 and w^2 are the words from dictionary $\mathbf{W}$, and I is the un-annotated image to be generated

uses content based image retrieval to search nearest neighbors for large scale image annotation. Nearest neighbor methods also can be solved in probabilistic framework. [5] proposes Tagprop which is also based on nearest neighbor model. Tagprop uses posterior to describe the annotation probability and obtained a likelihood function from training data, and then weights of distance from different features can be learned by optimizing the likelihood function. Other discriminative methods are also used for image annotation. [13] treats image annotation as a regression problem and use group sparsity to selected features for the annotation task. Kernel learning which is an effective discriminative method for image analysis, is used in [14] for image annotation.

III. TWO PHASE GENERATION MODEL

A. Description of the Model

Suppose there is an un-annotated image I, and there is a dictionary $\mathbf{W} = \{w_1, \ldots, w_k, \ldots, w_K\}$ of K words to annotate this image. The general methods either estimate the posterior probability $p(w_k \mid I)(k = 1, \ldots, K)$ or the joint probability $p(w_k, I)$ and they choose the words with the higher probability. In fact the posterior probability and joint probability are equivalent for annotation, because given an image I to be annotated, the prior of the image is fixed. However, in our work we adopt the generation probability $p(I \mid w_k)$ to describe the probability of annotation. Textual words usually show more semantic information than visual features of the image, thus the probability that the words generate the images is more reliable than the probability that the images generate the words, this means estimating $p(I \mid w_k)$ is better for the annotation than estimating $p(w_k \mid I)$. We also consider the relation of each word. Some words are semantically correlated, and they are likely to occur in the same image. For example, sun is likely to co-occur with sky, and ship is likely to be in the same images with sea. Thus we propose the two-phase generation model (TPGM) for the image annotation which can better describe the annotation probability of each word as well as describe the relation of them.

The core idea of the TPGM is that image is generated by the words according to the two phase generation. Firstly each original word will generate some words (they can contain the original word) which are semantically correlated to the original word. Then these semantically correlated words will generate the target image. The procedure of TPGM is:

1. Given a word w_i^1, generate the correlated word w_j^2 by probability $p(w_j^2 \mid w_i^1)$, $w_i^1, w_j^2 \in \mathbf{W}$
2. Using word w_j^2 to generate the image I by probability $p(I \mid w_j^2)$

Both w_i^1 and w_j^2 are one of the words in dictionary $\mathbf{W}$, we use the superscript 1 and 2 to distinguish them for the different roles of them in the two phases, w_i^1 and w_i^2 are the same words but their roles in the procedure of TPGM are different. Then the final generation probability of this model is:

$$p(I \mid w_i^1) = \sum_{j=1}^{K} p(I \mid w_j^2) p(w_j^2 \mid w_i^1), w_i^1 \in \mathbf{W} \qquad (1)$$

The first phase generation probability $p(I \mid w_j^2)$ describes the probability that w_j^2 generated I directly, it can also be used as the probability for annotation, and we refer the simple model which uses the first phase probability as one-phase generation model (OPGM). $p(w_j^2 \mid w_i^1)$ can be seen as the weight for each generation probability, it can improve the annotation results by utilizing the semantic correlation of the words. Thus our two-phase generation can better describe the annotation probability, because not only it is based on the more reliable generation from textual words, but also it utilizes the words correlation which may improve the annotation results. The graphic illustration of TPGM is shown in Figure 1. And we will find that TPGM is similar to pLSA [16], but they are different because the latent variables are replaced by annotation words in TPGM, thus it is not needed to learn latent variables in TPGM.

B. Estimation for TPGM

In order to estimate the final generation probability $p(I \mid w_i^1)$, we need to estimate two types of probability: $p(I \mid w_j^2)$ and $p(w_j^2 \mid w_i^1)$. Suppose there are N annotated images in the training data, then we need to learn the two types of probability from the training data. According to the procedure of TPGM, we find $p(I \mid w_j^2)$ and $p(w_j^2 \mid w_i^1)$ can be learned separately. Although we can estimate the generation probability $p(I \mid w_i^1)$ by the Bayesian approach, we consider that separately estimating

$p(I \mid w_j^2)$ and $p(w_j^2 \mid w_i^1)$ are better, because it is easier to be solved and some sophisticated probabilistic methods such as probabilistic SVM can be incorporated into our model.

We first estimate $p(I \mid w_j^2)$, the traditional method is using a parametric distribution such as Gaussian distribution to describe $p(I \mid w_j^2)$, and then the parameters of the distribution can be learned from the training data. However, due to the semantic gap, the relation of image I and word w_j^2 is complex to model. Existing distributions such as Gaussian may not be able to describe such a complex relation. On the other hand, discriminative method has the ability to map the image to the word at a relatively high precision. Thus we can convert the problem of estimating $p(I \mid w_j^2)$ to estimating the posterior $p(w_j^2 \mid I)$ by using the following equation:

$$p(I \mid w_j^2) = \frac{p(w_j^2 \mid I)p(I)}{p(w_j^2)}, w_j^2 \in \mathbf{W}. \qquad (2)$$

When annotating an image, $p(I)$ for each word is unchanged, thus it can be ignored, $p(w_j^2)$ is the prior of each word, we use the following equation to compute it:

$$p(w_j^2) = \frac{N(w_j^2) + \mu}{N + \mu}, w_j^2 \in \mathbf{W}. \qquad (3)$$

Where μ and μ' is the smoothing parameter, $N(w_j^2)$ is the number of w_j^2 occurs in the training images, N is the total number of images, the equation is obtained by using Bayes estimation with the beta prior and w_j^2 follows the Bernoulli distribution [15].

For posterior probability $p(w_j^2 \mid I)$, many probabilistic discriminative methods can be used to estimate it. We adopt the support vector machine (SVM) to estimate $p(w_j^2 \mid I)$. SVM is an effective method; it has good performance in discriminative tasks such as classification and is scalable to large-scale data. However, traditional SVM cannot output probabilities, the SVM scores should be calibrated to probabilities. So Libsvm which is a prevalent SVM tools [17], is adopted as the implementation of SVM, it can also output the probabilities and this make Libsvm can be perfectly incorporated into our model. We first train binary SVM for each word in $\mathbf{W}$ from the training images, each binary SVM is trained by using the one-versus-all scheme where images annotated by this word are positive samples and images un-annotated by this word are negative samples. Then each binary SVM estimates each posterior probability $p(w_j^2 \mid I)$ for the new un-annotated imag I. If we estimated $p(w_j^2 \mid I)$ and $p(w_j^2)$, by using equation (2) ,we can obtained the first phase generation probability $p(I \mid w_j^2)$.

After estimating the first phase generation probability $p(I \mid w_j^2)$. Then we estimate the conditional probability $p(w_j^2 \mid w_i^1)$. $p(w_j^2 \mid w_i^1)$ reflects the probability that how w_j^2 can be generated by w_i^1, and we use the co-occurrence of the two words in the training images to describe $p(w_j^2 \mid w_i^1)$. The co-occurrence measures the relation of words, words which are more likely to co-occur in an image means they are more likely to be correlated in semantic. For example, plane and sky is semantically correlated and they are likely to appear in an image, beach and sea are also likely to co-occur in an image. The conditional probability is obtained by the following equation:

$$p(w_j^2 \mid w_i^1) = \frac{N(w_j^2, w_i^1)}{\sum_{w_j^2 \in \mathbf{W}} N(w_j^2, w_i^1)}, w_i^1, w_j^2 \in \mathbf{W}. \qquad (4)$$

Where $N(w_j^2, w_i^1)$ denotes the number of annotated training images where w_i^1 and w_j^2 both appear. And equation (4) normalizes the co-occurrence number $N(w_j^2, w_i^1)$. This equation also measures the relation of each word and itself, it is obviously that words will generate themselves with the highest probability, which will make sure that the role of word w_i^1 from the final generation probability is more important than its related words in the two-phase generation procedure.

According to the above estimation for first phase generation probability $p(I \mid w_j^2)$ and the conditional probability $p(w_j^2 \mid w_i^1)$, it is obvious to know that the second phase generation probability $p(I \mid w_i^1)$ is influenced by these two factors. If the word whose most semantically correlated words are likely to generate the image I, then this word is also likely to generate image I, this means it is likely to annotate image I.

C. TPGM for annotation

Once we estimate the second phase generation probability $p(I \mid w_i^1)$ for all words in $\mathbf{W}$, then we can choose the words with the highest generation probability. In this paper we choose five words for each image, in fact the number of words for each image is not fixed for all cases. We use five words to make our experiment the same to the previous works. The generation procedure of TPGM is two-phase, and the annotation based on TPGM is also two-phase, the annotation procedure is:

TABLE I. The comparison of annotation performance for TPGM, OPGM, discriminative SVM on two data set:COre 5K and MIR FLickr

Dataset	Corel 5K			MIR FLickr		
method	**SVM**	**OPGM**	**TPGM**	**SVM**	**OPGM**	**TPGM**
Precision	0.32	0.32	**0.34**	0.49	0.39	0.44
Recall	0.38	0.39	**0.51**	0.40	**0.54**	0.50
F1	0.349	0.353	**0.408**	0.441	0.453	**0.468**
N+	146	146	**185**	**38**	**38**	**38**

1. For a new image I , using SVM and equation (2) to estimate the first phase generation probability $p(I \mid w_j^2)$;
2. Estimating the second phase generation probability $p(I \mid w_i^1)$ by equation (1).
3. Sort all $p(I \mid w_i^1), w_i^1 \in \mathbf{W}$, and choose M words whose generation probability is higher than the other words.

From the annotation procedure and our TPGM model described in previous section, we can find at the first phase, words are only related to the image, but in the second phase each word is also related to other words. Not only words which annotate the image are closely related to this image, but also their semantically related words should be related to this image. Thus TPGM for annotation has the advantage in considering the relation of words, TPGM can performs better than one phase annotation.

IV. Exeprimental Results

In this section we will discuss details of the dataset used for experiments. And then we show experimental results of our model as well as other image annotation methods. Finally we show some examples to illustrate the annotation result of our model.

A. Datasets and features

We use two dataset: Corel 5K and MIR Flickr [18] for the experiments on image annotation. Core 5K was first used in [3]. Since then, it has become an important benchmark for automatic image annotation. It consists of 5000 images from 50 Corel Stock Photo cds, and each cd includes 100 images on the same topic. Each image in the data set is assigned 1-5 keywords. Overall there are 371 words in the dataset. A fixed set of 499 images are used as testing set, and the rest is used for training. There are 260 words which both appearing in the testing and training set, thus following [5], we only consider these 260 words for annotation.

Images in Corel 5K usually share the same words if they are similar in visual content especially colors, which means if the color features of the un-annotated image are similar to the annotated images in training set, then words in the annotated images are likely to be the annotation words for the un-annotated image. This is not accord with the real world. Thus we also use another data set which is more close to the real world. The MIR Flickr data set contains 25000 images collected by downloading images from Flickr over a period of 15 months. The collection contains images under the Creative Common license that scored highest according to Flickr's "interestingness" score. These images were annotated for 24 concept words, including object categories but also more general scene elements such as sky, water or indoor. For 14 of the 24 concept words a second, stricter, annotation was made: for each concept a subset of positive images was selected where the concept is salient in the image. In total we therefore have 38 category words which is smaller than Corel 5K. The mean categories per images in this data set are near 5. We use 12500 images for training and the other 12500 images for testing which is the same to [19].

For both Corel 5K and MIR Flickr, we use 12 visual features from images. 11 features of them are 6 RGB, HSV, LAB color histograms, 4 SIFT histograms and GIST descriptor, the details of them are described in [5] and they can be downloaded from [20]. Besides, we also use HOG histograms [23]. We first extract HOG descriptors on 16×16 overlapping patches with a spacing of 2 pixels, and then use k-means of clustering the subset of HOG descriptors to form a visual vocabulary of 1000 visual words. Finally descriptors in each image are quantized into a histogram with 1000 visual words. When using SVM to train and predict the posterior probability of images $p(w_j^2 \mid I)$, we use histogram intersection kernel [20] for all histograms and use RBF kernel for GIST. All 12 kernels are combined with equal weights to form a combined kernel for SVM.

B. Results for Automatic image annotation

In this section we evaluate the performance of our TPGM for the task of automatic image annotation. We use a fixed number of words to annotate the images. Each image in the Core 5K and MIR Flickr are both annotated with 5 words.

TABLE II. THE COMPARISON OF ANNOTATION PERFORMANCE FOR TPGM AND ORTHER PREVIOUS METHODS ON COREL 5K

method	CRM[7]	MBRM[8]	SML[11]	JEC[4]	GS[13]	MRFA[22]	Tagprop[5]	TPGM
Precision	0.16	0.24	0.23	0.27	0.30	0.31	0.33	**0.34**
Recall	0.19	0.25	0.29	0.32	0.33	0.33	0.42	**0.51**
F1	0.174	0.24.5	0.257	0.293	0.314	0.316	0.370	**0.408**
N+	107	122	137	139	146	172	160	**185**

To evaluate the annotation performance we use recall and precision calculated for every word in the testing set. The recall can be calculated by the following equation:

$$recall = \frac{correct(w)}{true(w)}, w \in \mathbf{W}. \tag{5}$$

Where $correct(w)$ is the number of images correctly annotated with word w, $true(w)$ is the number of images having this word in ground-truth annotation. The precision can be calculated by:

$$precision = \frac{correct(w)}{annot(w)}, w \in \mathbf{W}. \tag{6}$$

Where $annot(w)$ is the number of images automatically annotated with word w. Then recall and precision values are averaged over all testing words. To combine recall and precision in a single efficiency measure, we use the F1 score which is:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{7}$$

Moreover we use N+ to denote the number of words with non-zero recall value.

To firstly illustrate the advantages of the TPGM, we first compare TPGM to the one-phase generation model (OPGM) which estimates the $p(I \mid w^2)$ and the discriminative method which uses SVM to estimate posterior probability $p(w^2 \mid I)$. The result of performance on corel 5K and MIR Flickr is in table I, the μ in equation (3) is set to 300 for Core 5K and 3500 for MIR Flickr. From table I we can find that TPGM performs best on F1 score, and the OPGM performs slightly better than discriminative SVM. This means estimating generation probability $p(I \mid w^2)$ is better than estimating posterior probability $p(w^2 \mid I)$, and two-phase generation which considers the relation of words can improve the performance. Moreover, the recall and N+ of TPGM is significantly better than OPGM on Corel 5K, but on MIR Flickr their N+ are the same and the recall of TPGM is even lower than OPGM. This is because there are more words in Corel 5K than in MIR Flickr, thus better relation of words can be obtained from Corel 5K. And from table I we can find for both three methods all 38 words in MIR Flickr have non-zero recall. This is also the reason why the improvement of TPGM on MIR Flickr is not more significant than on Corel 5K, generally if there are more words with non-zero recall, the average recall on all words is higher. At last the F1 score which measure the overall performance of annotation shows TPGM is better than OPGM even in MIR Flickr which has small number of words.

TABLE III. THE PERFORMANCE OF TOP 80 WRODS FOR TPGM AND ORTHER PREVIOUS METHODS ON COREL 5K

method	SVM	OPGM	TPGM
Precision	0.493	0.494	**0.653**
Recall	0.640	**0.648**	0.500
F1	0.557	0.561	**0.566**
N+	75	**76**	**76**

TABLE IV. THE PERFORMANCE OF LEAST 80 WRODS FOR TPGM AND ORTHER PREVIOUS METHODS ON COREL 5K

method	SVM	OPGM	TPGM
Precision	**0.125**	**0.125**	0.093
Recall	0.131	0.131	**0.481**
F1	0.128	0.128	**0.156**
N+	11	11	**40**

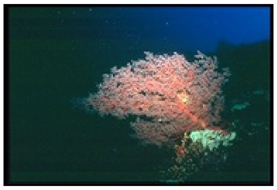
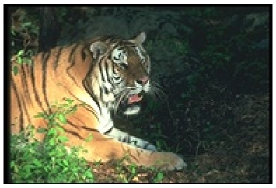
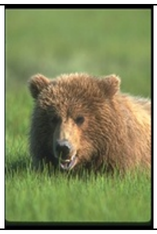
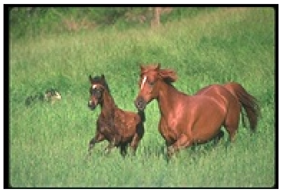
Image				
Original Annotation	sea, coral, fan, ocean, reefs	forest, cat, tiger, bengal	grass, bear, meadow, grizzly	field, horses, mare, mare, foals
Automatic Annotation	sea, coral, fan, ocean, reefs	forest, cat, tiger, Bengal, cougar	grass, bear, ground, meadow, grizzly	field, meadow, horses, mare, mare, foals

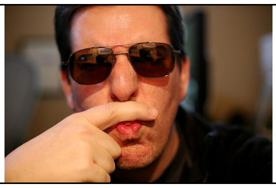
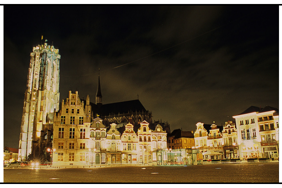
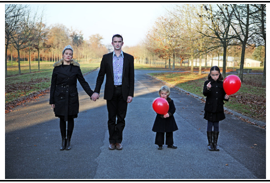
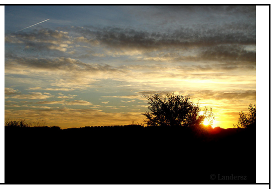
Image				
Original Annotation	indoor, male, people, portrait	clouds, night, sky, structures	female, male, people, plant life, portrait, sky, structures, tree	clouds, plant life, sky, sunset, transport, tree
Automatic Annotation	indoor, male, people, portrait	clouds, night, sky, structures	male, plant life, sky, structures, tree	clouds, sky, sunset, tree

Figure 2. Automatic annotations compared with the original manual annotations. The top of this figure shows the images in Core 5K, and the bottom of this figure shows the images in MIR FLickr

We can find even on MIR Flickr whose words are all likely to be annotated, TPGM performs better. To further show the characteristic of our model. We only evaluate the top 80 words of Corel 5K which appear more than the other words. Table III shows the results of the 80 words. We also evaluate the least 80 words of Core5K which appear least than the other words. Table IV shows the result of the least 80 words. From Table III and IV it can be find that the more a word appears in the data set, the more likely it will be annotated for images, the least words are not likely to be annotated. TPGM can solve this unbalance in some extent. For the top 80 words, most of them have non-zero recall by the three methods, and the improvement of TPGM is not significant. But for the least 80 words, TPGM leads more words to annotate images, in TPGM words with least appearance are more likely to appear in images than discriminative SVM and OPGM. This interprets why using the relation of words can improve the performance of annotation.

At last we compare TPGM to other previous methods on Corel 5K, the comparison on MIR Flickr is absence because little previous works focus on this data set for experiment. The result in Table II shows that our TPGM outperform other previous methods on the annotation task of Corel 5K. Especial the recall and N+ of TPGM is significant better than other methods, this may due to the analyzing of words relation in TPGM. If some words may hardly be annotated for images, but other words which are related to them may be easily learned by the system, then these words are more likely to be annotated according to their related words. MRFA use markov random field to model the relation of words, thus it also annotate more words than the rest methods. Furthermore, our model uses the classical discriminative method SVM for the generation procedure, and we do not improve SVM itself, so the precision of our model is not much higher than other discriminative annotation methods such as Tagprop.

C. *Illustrative results*

This section shows some illustrative examples of the annotations generated by our model. Figure 2 shows the automatic annotation examples of TPGM compared with original images. For images in Corel 5K, we can find that the original annotation lost some words and our model can predict the lost words. Our model annotates “ground” for image 3 and “meadow” for image 4 in the top of the figure, these words are missed in the original annotation but they also describe the content of the images. Our model is surely not perfect and may make some mistakes. We can find that our model make the incorrect annotation by recognizing the tiger as cougar in the top image 3. Our model also failed to annotate “female”, “people”, “portrait” in bottom image 3 and “plant life”, “transport” in bottom image 4. The “transport” in bottom image 4 is too dark to recognize, thus it may be reasonable for our model to miss it.

V. CONSLUSIONS AND FUTURE WORK

In this paper we proposed the two-phase generation model for automatic image annotation. Unlike previous methods, TPGM estimates the probability that words generate the images. And a two-phase generation procedure which considers the relation of words is used to estimate the generation probability. Experimental results show that TPGM will make more words in the dictionary to be annotated and performs better than one-phase generation model and general discriminative methods such as SVM on two datasets. TPGM also outperforms previous generative and discriminative methods on Corel 5K.

Our model has remained some areas to be improved. For the generation where words generate their most related words, we use the co-occurrence to measure the relation. However the relation of words may be more complex and a more sophisticated method need to design for analyzing the semantic relation of words. Then for estimating the first generation probability that words generate images, we use a normal SVM and it can be replaced by some state-of-the-art discriminative methods. Multiple kernel learning [24] may be suitable for our model, it learns different weights for different kernels which may make our model performs better than directly combining kernels with equal weights.

REFERENCES

[1] Datta, Ritendra, et al. "Image retrieval: Ideas, influences, and trends of the new age." ACM Computing Surveys (CSUR) 40.2 (2008): 5.

[2] Mori, Yasuhide, Hironobu Takahashi, and Ryuichi Oka. "Image-to-word transformation based on dividing and vector quantizing images with words." First International Workshop on Multimedia Intelligent Storage and Retrieval Management. 1999.

[3] Duygulu, Pinar, et al. "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary." Computer Vision—ECCV 2002. Springer Berlin Heidelberg, 2006. 97-112.

[4] Makadia, Ameesh, Vladimir Pavlovic, and Sanjiv Kumar. "A new baseline for image annotation." Computer Vision–ECCV 2008. Springer Berlin Heidelberg, 2008. 316-329.

[5] Guillaumin, Matthieu, et al. "Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation." Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009.

[6] Jeon, Jiwoon, Victor Lavrenko, and Raghavan Manmatha. "Automatic image annotation and retrieval using cross-media relevance models." Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. ACM, 2003.

[7] Lavrenko, Victor, R. Manmatha, and Jiwoon Jeon. "A model for learning the semantics of pictures." Advances in neural information processing systems. 2003.

[8] Feng, S. L., Raghavan Manmatha, and Victor Lavrenko. "Multiple bernoulli relevance models for image and video annotation." Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. Vol. 2. IEEE, 2004.

[9] Blei, David M., and Michael I. Jordan. "Modeling annotated data." Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. ACM, 2003.

[10] Monay, Florent, and Daniel Gatica-Perez. "Modeling semantic aspects for cross-media image indexing." Pattern Analysis and Machine Intelligence, IEEE Transactions on 29.10 (2007): 1802-1817.

[11] Carneiro, Gustavo, et al. "Supervised learning of semantic classes for image annotation and retrieval." Pattern Analysis and Machine Intelligence, IEEE Transactions on 29.3 (2007): 394-410.

[12] Li, Xirong, et al. "Image annotation by large-scale content-based image retrieval." Proceedings of the 14th annual ACM international conference on Multimedia. ACM, 2006.

[13] Zhang, Shaoting, et al. "Automatic image annotation and retrieval using group sparsity." Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 42.3 (2012): 838-849.

[14] Yuan, Ying, et al. "Image annotation by composite kernel learning with group structure." Proceedings of the 19th ACM international conference on Multimedia. ACM, 2011.

[15] Bishop, Christopher M., and Nasser M. Nasrabadi. Pattern recognition and machine learning. Vol. 1. New York: springer, 2006.

[16] Hofmann, Thomas. "Unsupervised learning by probabilistic latent semantic analysis." Machine learning 42.1-2 (2001): 177-196.

[17] Chang, Chih-Chung, and Chih-Jen Lin. "LIBSVM: a library for support vector machines." ACM Transactions on Intelligent Systems and Technology (TIST) 2.3 (2011): 27.

[18] Huiskes, Mark J., and Michael S. Lew. "The MIR flickr retrieval evaluation." Proceedings of the 1st ACM international conference on Multimedia information retrieval. ACM, 2008.

[19] Guillaumin, Matthieu, Jakob Verbeek, and Cordelia Schmid. "Multimodal semi-supervised learning for image classification." Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010.

[20] INRIA features for image annotation and classification data sets, http://lear.inrialpes.fr/people/guillaumin/data.php

[21] Barla, Annalisa, Francesca Odone, and Alessandro Verri. "Histogram intersection kernel for image classification." Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on. Vol. 3. IEEE, 2003.

[22] Xiang, Yu, et al. "A revisit of generative model for automatic image annotation using markov random fields." Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009.

[23] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, 2005.

[24] Bach, Francis R., Gert RG Lanckriet, and Michael I. Jordan. "Multiple kernel learning, conic duality, and the SMO algorithm." *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004.