2007

This is to certify that the
dissertation entitled

Automatic Image Annotation

presented by

Feng Kang

has been accepted towards fulfillment
of the requirements for the

Ph.D. degree in Computer Science

Major Professor's Signature

08/16/2007

Date

DATE DUE	DATE DUE	DATE DUE
FEB 2 1 2009		
10/1/10 0 9 27 10		

AUTOMATIC IMAGE ANNOTATION

By

Feng Kang

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science

2007

ABSTRACT

AUTOMATIC IMAGE ANNOTATION

By

Feng Kang

Automatic image annotation is to annotate an image with a set of textual words. The key in this process is to learn a statistical model which correlates the image features with the annotation words. To construct the statistical model, we start with a set of training images, each of which has a set of accompanying annotation words. Typically, images are first segmented into multiple homogeneous regions. Image features such as color and texture are then extracted to represent each image region. The image regions in the whole collection can also be grouped into clusters and thus each image region could be converted into its corresponding cluster id, called a blob. In this way, we obtain a discrete representation of the images. The correlation between annotation words and image features, either discrete or continuous, is constructed with a statistical model. Finally, given a new test image, the same set of image features are extracted, and words are predicated according to the relationship between image features and annotation words established by the learned statistical model.

In this thesis, we explore the automatic image annotation task through a series of statistical models. One model based on the discrete feature representation is the translation model, which constructs the correspondence between blobs and annotation words through a set of translation probabilities. Due to the fact that common words co-occur with many more blobs than rare words, the original translation model overestimates the common words and degrades the overall performance. We thus propose two enhanced translation models to improve the original translation model by incorporating different prior information of the desired translation probabilities into the model. One prior ensures that each word is associated with similar number of blobs, which is measured by the average of

the translation probabilities from different blobs to the word. Another prior considers the translation model from two directions: forward translation model, which translates from blobs to words; backward translation model, which translates from words to blobs. The prior specifies that the translation probabilities from these two kinds of models should be consistent with each other. Our empirical results demonstrate the improved performance of the two enhanced translation models over the original translation model.

However, there are still two problems with the translation models. First, they do not consider the correlation between annotation words when making the prediction. Secondly, they are based on the discrete representation, which potentially loses information encoded in the continuous features. However, the correlation information is difficult to explore since the possible number of correlated words is exponential. We propose a Correlated Label Propagation (CLP) framework to explore the correlation between annotation labels. Based on the property of the submodular function, this framework could be solved by a very efficient greedy algorithm and thus be applicable to a large set of labels. In addition, the continuous image features could be incorporated into the CLP framework. Our results show that the CLP framework outperforms the translation models and also can boost the performance to a higher level after the continuous features are incorporated.

In summary, this dissertation shows that 1) The performance of the original translation model can be improved by incorporating different priors; 2) Effectively exploring the correlation information between labels can improve the overall performance; 3) Similarity measurement is very important in label propagation and similarity measurement based on the continuous image features can achieve better performance.

ACKNOWLEDGMENTS

There are a lot of people I would like to thank for their help in this writing of thesis.

Firstly, I would like to thank my advisor, Dr. Rong Jin. This thesis is impossible without his guidance and support. Not only does he guide the writing of the thesis, he also corrects lots of parts in great detail. Also, thanks for his support and help during my Ph.D study so that I could explore this interesting field.

I would also like to thank the other committee members: Dr. Joyce Chai, Dr. Pangning Tan, and Dr. Selin Aviyente. They provide lots of helpful suggestions and discussions.

I would also like to thank other fellow students, Yi Liu, Liu Yang, Feilong Chen, Ming Wu, Tong Wei, Shaoling Qu, Chen Zhang, Matthew Gerber, Zahar Prasov, Haibing Chen, for their help during the Ph.D study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Image retrieval started as a text-based image retrieval system in the late 1970's. By manually annotating images with descriptive texts, existing text-based Database Management Systems were utilized to perform image retrieval. However, image retrieval based on manual annotations has two major drawbacks:

- It is expensive to hire large numbers of people to annotate large-sized image collections.
- Given that human perception of image content is highly subjective, different people might give different interpretations for the same pictures, depending on their knowledge, experience, mood and other circumstances.

To overcome the problems of image retrieval based on manual annotation, people have proposed content-based image retrieval, in which images are indexed and represented by their visual features, such as color, texture, shape etc. Content-based image retrieval relies on the low-level visual feature representation of images, thus avoiding the imprecise human annotation. A comprehensive survey on this subject can be found in[42]. While content-based image retrieval solves the problems related to text-based image retrieval, it introduces several new problems:

- While images could be retrieved based on their features such as color and texture, it is usually more natural and desirable for users to search image databases using textual queries. This is because textual queries allow users to express their information needs at the semantic level instead of the level of preliminary image features.

- Compared to content-based image retrieval, textual queries usually provide more accurate descriptions of users' information needs. For example, consider that a user is looking for images of tigers. Suppose he uses an image query, which consists of a photo of a tiger on the grass. Based on the match of image features, many of the retrieved images will be pictures of the grass without any tigers. This is because it is unclear to the system what the user is searching for, the grass or a tiger. In contrast, textual queries such as 'photos of tigers' are able to convey the information need of the user more clearly.

To search images based on textual queries, it's preferable that the annotation words for a large set of image collections can be automatically generated based on a model built from a small number of manually annotated training images. To search for images, users can simply pose textual queries and the relevant images are retrieved based on the matching between the textual queries and the automatically generated annotations for images. Unlike traditional text-based image retrieval, the automatic annotation by a computer program avoids the expense and subjectivity of the manual annotation.

Figure 1.1 illustrates the basic steps for automatic image annotation: an image is first segmented into multiple homogeneous regions; then image features, such as color, texture and shape, are extracted to represent each image region; finally, annotation words are predicted based on the visual features of segmented regions using a statistical model that is learned from the training data.

1.1 Methods in Automatic Image Annotation

One important issue with automatic image annotation is how to represent image regions. One type of approach represents every image region with the extracted continuous features,

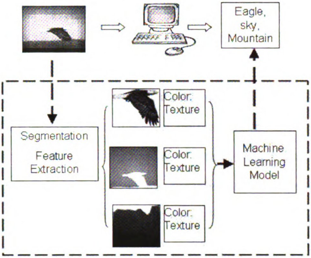


Figure 1.1. Procedure for automatic image annotation

such as color, texture, and shape. Another type of approach is to represent the image regions with the discrete features from a visual vocabulary. To obtain the visual vocabulary, we first cluster image regions of the whole collection into a small number of groups and represent each region by its cluster id, i.e. which cluster the region belongs to. These cluster ids are called *blobs* and they are the terms in the visual vocabulary. Then we can convert the image regions in each image to its corresponding cluster id and obtain the discrete representation. Compared to the representation using extracted features, the quantized representation can dramatically simplify the computation in automatic image annotation, but at the price of potentially losing information that is encoded in extracted continuous features.

The key to automatic image annotation is to learn annotation models that can automatically predict annotation words given extracted image features. Most automatic annotation methods applied machine learning techniques to first learn the correlation between image features and textual words from the examples of annotated images and then apply the learned correlation to predict words for unseen images. Based on what kinds of image representations the models are using, we group the annotation methods into two categories: the models based on discrete features and the models based on continuous features.

The models based on discrete features include: the machine translation model[41], the co-occurrence model[39], latent space approaches[37, 38],and relevance language models[24]. The co-occurrence model[39] collects the co-occurrence counts between words and image features and uses them to predict annotated words for images. Duygulu et al.[41] improved the co-occurrence model by utilizing machine translation models. Another way of capturing co-occurrence information is to introduce latent variables that link image features with words. Methods in this category include latent semantic analysis (LSA)[37] and probabilistic latent semantic analysis (PLSA)[38]

The models based on continuous features include: classification approaches[9], the continuous relevance language model[24], the hierarchical aspect model[28], the Gaussian Mixture Model (GMM), the Latent Dirichlet Allocator (LDA), and the correspondence LDA[5]. The classification approaches for automatic image annotation treat each annotated word as an independent class and create a different image classification model for every word. Work such as linguistic indexing of pictures[33], image annotation using SVM and Bayes point machine[9] fall into this category. Continuous relevance language models[32] is an application of relevance model[31] to the continuous features. Hierarchical aspect model[28], Gaussian Mixture Model (GMM), Latent Dirichlet Allocator (LDA), and correspondence LDA[5] are methods that introduce the correspondence between annotation words through latent variables. We will discuss these methods further in Chapter 3.

1.2 Issues in Automatic Image Annotation

In this section, we discuss some common issues in the image annotation task.

1.2.1 Skewed Distribution of Annotation Words

When we annotate the images, we simply put the annotation words as a description of the images and do not identify the corresponding image regions. Thus the computer is not sure which image region corresponds to which specific word. In addition, some of the common scenes appear frequently in many images such as 'water', 'sky' while the major subjects may differ across images. As a result, the annotation words follow a very skewed distribution. A few words happen very frequently and most of the words happen rarely. This is particulary a problem for the translation model, since the translation probability is based on the co-occurrence statistics between annotation words and blobs. The translation probabilities for the common words are thus much higher and the rare words have fewer opportunities to appear in the annotation set. We thus propose two enhanced translation models to alleviate this problem.

- *Regularized Translation Model.* One reason that the common words are overestimated is because the common words are associated with more blobs than rare words. One strategy to alleviate this problem is to force each word to be associated with a similar number of blobs. The number of blobs associated with each word can be measured by the average of the translation probabilities to the word. This measurement is treated as a prior to be incorporated into the original translation model to obtain better set of translation probabilities.

- *Symmetric Regularized Translation Model.* Typically, the translation model translates blobs to words. This model assumes that each of the blobs can be translated into one of the words. If the common words have higher probabilities associated with the blobs, rare words will have a smaller probability of appearing in the annotation set. Another view of the translation model is to translate from words to blobs, which assumes that each of the words could be translated into one of the blobs. In this way, the rare words could be associated with blobs with higher probability. Our prior is to enforce the consistency of these two sets of translation probabilities.

1.2.2 Correlation between Annotation Words

We know that many of the annotation words do not exist alone. They are correlated with each other. For example, the word 'grass' may appear more often with the word 'tiger' than with the word 'ship'. A 'whale' is usually related to the word 'water' but may never occur with 'grass'. How to effectively explore the correlation between annotation words is very challenging given the large size of annotation labels in our task. While this problem could be explicitly handled by specifying which set of labels are correlated together, it becomes a combinatorial problem and is difficult to scale to the large number of annotation words.

We investigate two approaches to this problem.

- *Multi-label Maximum Entropy Model.* This model takes the pairwise label constraints into consideration by specifying which pair of labels are correlated together and how strong the correlation is. However, due to computational issues, this model cannot be applied to our annotation task.

- *Correlated Label Propagation(CLP) Framework.* We propose a correlated label propagation framework to take into account the correlation information. Label propagation is not only based on the similarity between test images and training images. We also consider correlation between labels. We show that this framework has an optimal solution based on the properties of the submodular function. In addition, this solution can be obtained through an efficient greedy algorithm and thus be able to scale to a large set of labels. Furthermore, this framework can incorporate continuous features, which are missing in models based on the discrete features, such as the translation model.

1.3 Outline of the Thesis

In Chapter 2, we will provide some background information about image preprocessing, which mainly focuses on the feature representations and image segmentation. In Chapter 3, we discuss the two categories of statistical models built for automatic image annotation based on whether they employ discrete or continuous representation of the images. Chapter 4 focuses on the the translation models. We will discuss in detail its over-estimated common words problem. We then propose two enhanced translation models to alleviate this problem and demonstrate the improved performance. Chapter 5 presents the methods that consider correlations between labels. The Correlated Label Propagation(CLP) framework is proposed and demonstrates the improved performance over discrete features. We also show that the CLP framework can incorporate continuous features to achieve better performance than the discrete features. Chapter 6 summarizes this thesis.

CHAPTER 2

Images Preprocessing

2.1 Feature Representation

Visual features of images can be classified into two categories[42]: the general image features and the domain-specific features. The general image features include color, texture, shape etc. The domain-specific features are application dependent. For example, to detect and recognize human faces, special image features are required to describe the characteristics of human faces. In this survey, we focus on the general image features, particularly the features that are critical to automatic image annotation[41].

2.1.1 Color

Color is one of the most widely used visual features in image retrieval. Compared to other image features, color is relatively robust to the background complication and independent of image size and orientation[42]. It is typically represented in a color space. Each color space involves two basic components: the primaries and the matching function. The primaries of a color space describe the basic and independent components of colors. The matching function of a color space determines the weight of the primaries in matching a source color. Suppose the primaries areP_1, P_2, P_3 and the matching functions are f_1, f_2, f_3. Then any color could be expressed as: $f_1P_1 + f_2P_2 + f_3P_3$. One typical example of color space is RGB Color Space, in which the single wavelength primaries are used (645.16 nm for R, 526.32 nm for G and 444.44 nm for B). Another popular color space is CIE L*u*v, which is a more uniform space than RGB and obtained by a projective transformation of CIE XYZ [16], which yields the CIE u, v space:

$$(u, v) = \left(\frac{4X}{X + 15Y + 3Z}, \frac{9Y}{X + 15Y + 3Z} \right) \tag{2.1}$$

Color information of an image is usually represented by its statistics. There are two commonly used statistics for representing color information:

- Color histogram.

  In image retrieval, the color histogram is the most commonly used color feature representation[42]. It is easily computed and in many cases very effective for image retrieval. However, one disadvantage of the color histogram is that it is usually a sparse representation with zero values for most of its entries. As a result, it can be sensitive to small distortions to images when applied to image retrieval[42].

- Color layout.

  The color histogram is usually viewed as a global feature given that it is computed based on the color distribution of the entire image. Despite that it has shown a certain degree of discriminative power in image retrieval, one disadvantage of any global feature is that it is unable to provide an accurate description of details in images and thus tends to give many false positives when the size of the image collections is large. In contrast, color layout methods take into account the local distribution of color features. One example of color a layout method divides images into multiple blocks and extracts image features for each block. In[34, 35], the author proposed an approach, named 'single blob with neighbors' or SBN, in which an image is divided into multiple sub-images and each sub-image is represented by the mean values of RGB components of the sub-image itself and its four neighboring blobs (i.e., up, down, left, right). This approach can be further improved by organizing sub-images into more complicated structures such as the quad-tree.

2.1.2 Texture

According to[42], textures are the homogeneous visual patterns that do not result from the presence of only a single color or intensity. Some of the commonly used texture representations include:

- Co-occurrence matrix.

  This method represents image texture using the spatial correlation of gray levels of different pixels [43]. It first constructs a co-occurrence matrix based on the orientation and distance between image pixels and then extracts meaningful statistics from the co-occurrence matrix as the texture representation.

- Tamura psychological perspectives

  Tamura explores the texture representation from psychological perspectives[16]. It identifies six visual properties that are related to textures: coarseness, contrast, directionality, line-likeness, regularity, and roughness. Tamura representations of texture differ from the co-occurrence representation in that their texture properties are visually meaningful whereas many of the texture properties extracted from the co-occurrence matrix representation are not.

- Spectral transform

  This method applies the wavelet transform or the Fourier transform to obtain texture representations in the frequency domain. One disadvantage of the Fourier transform is that the computation of the transform requires the pixel information from the entire image. As a result, the Fourier transform is unable to capture the local structure of images, which is important in representing images. To overcome this problem, one strategy is to use the Gabor filters[18, 47], in which a Gaussian kernel is put on top of the Fourier transform so that the transform is performed within a neighborhood determined by the standard deviation of the Gaussian kernel. Another strategy is to use the wavelet transform. In order to capture the local structures of images, the wavelet transform uses ‘wavelet’ as its basis functions, which are usually generated from a mother function by translation and contraction.

2.1.3 Shape

Shape is a very important visual feature in human perception. A good shape representation should be invariant to translation, rotation and scaling. Most shape representations can be classified into two categories: region based methods, and boundary based methods.

Boundary based methods represent the shape of a region by its outer boundary. They can be further classified into three sub-categories[50]:

- Global shape descriptors include area, circularity, eccentricity, and axis orientation. These shape features can only distinguish shapes with large dissimilarity.
- Shape signatures utilize the local feature of a shape, including the complex coordinates, the curvature and the angular features. They are usually sensitive to noise and therefore are not robust. In addition, they require intensive computation during similarity calculation, due to the hard normalization of rotation invariance.
- Spectral descriptors apply spectral transformation to the shape signatures to acquire the shape representation in the frequency domain. The most successful shape representation is the Fourier Descriptor[42], which applies the Fourier transform to the shape signatures.

In the region-based representations, all pixels within a shape region are taken into account to obtain the shape representation. The most successful representation is the moment descriptor [42], which is invariant to both translation and scaling.

2.2 Image Segmentation

The goal of image segmentation is to group pixels of similar properties into clusters. According to [16], methods for image segmentation can be classified into three categories:

- Image segmentation based on clustering.
- Image segmentation based on fitting.
- Image segmentation based on probabilistic methods.

In addition, recently there have been a number of studies on image segmentation at the semantic level[2], which will also be reviewed below.

2.2.1 Image segmentation as clustering

Image segmentation can be viewed as a clustering problem in that pixels of similar visual properties are clustered into a small number of groups. For any clustering algorithm, there are two key components:

- Distance measurement between any pair of data points or clusters.

  Commonly used distance measurements are:

  1. The single link, i.e., the shortest distance between any pair of data points in two clusters.
  2. The complete link, i.e., the maximum distance between any two points in two clusters.
  3. The group average link, i.e., the average distance between any pair of data points in the two cluster.

- The number of clusters. Usually, this is a very difficult problem to handle. Dendrogram[16], a hierarchical representation structure of clusters, could be used to see whether the clusters are good or not and help user make choice of clusters. Other commonly used approaches include the information criterion (e.g., AIC and BIC)[19].

The commonly used clustering algorithms for image segmentation are:

- Image segmentation by the K-means clustering algorithm

  The K-means algorithm[16] is one of the most commonly used clustering algorithms. It minimizes the within-cluster distance, which is calculated as the sum of distances of each data point in a cluster to its center. The cluster memberships of each data point are calculated through an iteration of the following two steps[16]:

  1. Given the current estimation of cluster centers, each data point is assigned to the cluster whose center is closest to the location of the data point.
  2. Given the cluster memberships of data points, the new center of a cluster is re computed as the average of the data points assigned to the cluster.

- Image segmentation by graph-theoretic clustering

  A clustering problem can be viewed as a problem of graph partitioning[16], in which each data point corresponds to a vertex in a graph, and the weight for each edge that connects two vertices is equivalent to the similarity of the corresponding two data points. Identifying clusters of similar data points is equal to dividing the corresponding graph into multiple disjoint sets with only edges of small weights removed. Central to the graph partitioning approaches is the similarity measure for any pair of data points. Different image features could be used to define the similarity measurement, such as the similarity in intensity, color, texture and motion.

  One of the popular algorithms in this category is the Normalized Cut[44, 45]. It divides a graph into two disjoint sub-graphs such that the ratio of the graph cut to the total affinity (i.e., similarity) within each sub-graph is minimized. In particular, to divide a weighted graph V into the two disjoint subsets A and B, based on the Normalized Cut algorithm, it is formulated into the following optimization problem:

$$\arg\min_{A,B} \frac{cut(A,B)}{asso(A,V)} + \frac{cut(A,B)}{asso(\mathbf{B},V)} \tag{2.2}$$

$cut(A,B)$ measures the similarity between the components A and B, which is defined as the sum of weights of all the edges in V that have one end in component A and the other end in component B. $asso(A,V)$ and $asso(B,V)$ measure the similarity of data points within component A and B. They are the sum of weights of all edges that have both ends in A and B, respectively. Given that the above problem is a combinatorial optimization problem and is NP-hard, usually approximate solutions are provided[44, 45].

2.2.2 Image Segmentation as Fitting

The goal of fitting is to determine possible structures observed in an image. An image can be viewed as a set of tokens, which can be pixels or edge points. The fitting-based image segmentation approaches group these tokens together to form regular shapes such as lines and circles. There are two major approaches in this category:

- Hough transform

  The Hough transform clusters pixels together based on their underlying structures[16]. It first identifies and stores all possible structures for each pixel, and then searches for structures that are commonly shared by many pixels. The Hough transform has been used to identify lines, curves, surfaces etc. It is advantageous in that it does not require computing analytical solutions of certain equations. However, the Hough transform is usually sensitive to noise[16], which can lead to the identification of phantoms.

- Curve fitting based on generative models.

A fitting approach based on a generative model assumes a probabilistic model that describes the probabilistic relationship between the observed pixels and the underlying curves. Usually, pixels are assumed to be generated from certain curves such as lines under Gaussian noises. Parameters governing the underlying curves are estimated through certain criteria, such as the least square criterion[16].

2.2.3 Probabilistic Methods for Image Segmentation

The Expectation Maximization (EM) algorithm[36] has been demonstrated to be an effective approach for missing data problems. Image segmentation can be viewed as a missing data problem, in which each image segment is assumed to be generated from a mixtures of probabilistic models and the missing information is the description of each probabilistic model. The EM algorithm computes the segmentation in the alternation of the E-step and the M-step:

- E-step: the EM algorithm estimates the segment membership for each pixel;
- M-step: the optimal parameters of mixture model are estimated based on the segment memberships that are estimated in the E-step.

These two steps are iterated until the EM algorithm converges to its local optimum.

In addition to image segmentation, the EM algorithm can also be applied to fit lines. Similar to applying the EM algorithm to image segmentation, in the E-step, we estimate the likelihood for each pixel to be in a line. Usually, this likelihood is proportional to its distance to the line and is cut by a threshold value. In the M-step, a maximum likelihood approach is used to re-estimate the parameters of the line. To determine when the iterative procedure should be stopped, we need to test convergence of the algorithm, which often is based on the change of size of the line and also the sum of distances from the points to the line [16].

Despite of the success of the EM algorithm for image segmentation, a general problem with the EM algorithm is its local minimum, which means that the choice of starting point will have a great impact on the quality of final solutions. One solution is to first apply the Hough transform to obtain the initial solution for the EM algorithm, or we can start the EM algorithm with different initial solutions, and search for the best solution[16].

2.2.4 Segmentation by Integrating Semantic Information

According to previous studies [2], image segmentations based on low-level features usually will not result in desirable object recognition. This is because the same object may exhibit very different distributions of image features in different images or even in the same image. For example, a penguin has white and black halves, and it is hard to acquire meaningful segmentation results just based on the low-level features. In [2], a solution is proposed by associating each segmentation region with a meaningful word. Then, two neighboring regions are merged when they have been assigned the same annotation words. By doing this, we are able to avoid creating too many image segments.

2.2.5 Evaluation

We have discussed different types of image features and image segmentation methods. One important yet unresolved issue is how to measure the 'goodness' of feature extraction and image segmentation. That is, the user wants to know how good a set of image features are or how good the segmentation is. Evaluation based on human judgment can only be done on small scales. For a large image collection, automatic mechanisms are required to measure the quality of image feature extraction and segmentation.

One automatic approach evaluates the performance of different approaches based on the annotations of images. Word annotations for images are automatically generated based on the selected image features and segmentation methods. These automatically generated annotations are then compared against the ground truth and the resulting annotation accuracy is used as a proxy to the performance of image feature extraction and segmentation. Based on this method, [2] compares the importance of different image features. The empirical study showed that the L*a*b color space is more effective than the RGB color space. For image segmentation algorithms, it showed that the Normalized Cut algorithm performs significantly better than the Blobworld segmenter. The mean-shift algorithm[11, 12] for image segmentation is somewhere between these two algorithms [2].

[14] compared commonly used image features including size, position, color and texture with a complicated set of features from visual content descriptions in MPEG-7. Surprisingly, the empirical study showed that the commonly used image features achieve better performance than the complicated ones.

Overall, previous studies show that no single set of image features can perform well for all different image collections and thus it is likely that the choice of image features depends on the characteristics of specific applications and datasets.

CHAPTER 3

Statistical Models for Image Annotation

Image annotation describes the content of images with a set of textual words. This set of annotation words can be obtained by manual annotation from humans. However, manual annotation is expensive and requires extensive labor work. Also, human annotation is very subjective. Even for the same picture, different people might use different annotation words. Thus automatic image annotation is preferred. In this chapter, we discuss statistical models for automatic image annotation. This type of annotation is divided into two categories. The first one annotates the images based on quantized image regions. The second one annotates the images based on the raw image features.

3.1 Image Annotation Based on Quantized Image Regions

One type of approaches toward automatic image annotation first clusters image regions into a small number of groups, which are called 'blobs' in [41]. The approaches then learn the correlation between annotated words and blobs. Automatic clustering methods, such as the K-means algorithm, can be applied to cluster image regions and get their cluster ids. One advantage of this discrete representation is that it simplifies the image representation dramatically, thus significantly reducing the computational cost of automatic image annotation.

Before discussing formal models for automatic image annotation based on the discrete representation, we will first formalize this problem. Let the collection of annotated images be denoted by T, and the size of the collection be denoted by $|T|$. Each annotated image $J_i \in J$ is divided into multiple regions. All the image regions are clustered into a small number of blobs $b_{i,1}, b_{i,2}, ..., b_{i,m}$. Then each image can be represented by a vector of blobs and annotation words, i.e., $J_i = \{\overrightarrow{b_i}, \overrightarrow{w_i}\} = \{b_{i,1}, b_{i,2}, ..., b_{i,m}; w_{i,1}, w_{i,2}, ..., w_{i,n}\}$. Here, m and n are the number of blobs and annotation words, respectively; $b_{i,j}$ is the number of j-th blob that appears in the i-th image; $w_{i,j}$ is a binary variable, which is 1 when the j-th word appears in the i-th image and zero otherwise. The key to automatic image annotation is to learn the statistical correlation between the blob representation and the word representation of images.

To construct this kind of models, the first task is to quantize the image regions. One method is to perform a clustering procedure on the segmented image regions. The resulting clusters are the quantized image regions and their cluster ids are treated as the discrete representations of the images. That is, if one image region in the image belongs to one cluster, the image region is converted to its cluster id. The translation model for automatic image annotation[41] and the cross media relevance model[24] use this kind of quantized representation. Another way to quantize the image regions is to construct discrete features directly, such as the color histogram. This kind of representation is used in models such as Latent Semantic Analysis and Probabilistic Latent Semantic Analysis[37, 38].

3.1.1 Translation Model for Automatic Image Annotation

The translation model was originally developed for language translation[7], e.g. translating from French text to their English equivalent. [41] views the process of annotating images as a process of translating information from a 'visual language' to textual words. The lexicon of the visual language is the blobs. Compared to other models for automatic image annotation, such as the relevance models[24], statistical machine translation models for automatic image annotation have the advantage that words are annotated to each image region instead of to the whole image.

Based on the IBM model 1 of translation [41], given an annotated image $J_i = \{\overrightarrow{b_i}, \overrightarrow{w_i}\} = \{b_{i,1}, b_{i,2}, ..., b_{i,m}; w_{i,1}, w_{i,2}, ..., w_{i,n}\}$, the probability of annotating image blobs $b_{i,1}, b_{i,2}, ..., b_{i,m}$ with words $w_{i,1}, w_{i,2}, ..., w_{i,n}$, i.e.,$p(\overrightarrow{w_i}, \overrightarrow{b_i})$, can be expressed as follows:

$$p(\overrightarrow{w_i}|\overrightarrow{b_i}) = \prod_{j=1}^{n} p(w_{i,j}|\overrightarrow{b_i}) = \prod_{j=1}^{n}\sum_{k=1}^{m} t_{j,k}b_{i,k} \tag{3.1}$$

where $t_{j,k}$ stands for the probability of translating the k-th blob into the j-th word and is subject to the constraint $\sum_j t_{j,k} = 1$, namely each blob has to be translated into one of the annotated words. The key to a translation model for image annotation is the set of translation probabilities $t_{j,k}$. These probabilities can be obtained by maximizing the likelihood of annotated training images T, i.e.:

$$l(T) = \prod_{i=1}^{|T|} p(\overrightarrow{w_i}|\overrightarrow{b_i}) = \prod_{i=1}^{|T|}\prod_{j=1}^{n}\sum_{k=1}^{m} t_{j,k}b_{i,k} \tag{3.2}$$

The Expectation-Maximization (EM) algorithm[41] is applied to find the optimal solution for Equation3.2. It iteratively updates the translation probabilities using the following equation:

$$t_{j,k}^{new} = \frac{1}{Z_k}\sum_i \frac{w_{i,j}b_{j,k}t_{j,k}^{old}}{\sum_l b_{i,l}t_{j,l}^{old}} \tag{3.3}$$

where $t_{j,k}^{old}$ and $t_{j,k}^{new}$ are the translation probabilities learned in the previous and current iteration, respectively. Z_k is a normalization factor that ensures $\sum_j t_{j,k} = 1$. According to Equation 3.3, a common word may have large translation probabilities for many different blobs since it appears in many different annotations and therefore its term frequency $w_{i,j}$ is non-zero for a large number of annotated examples. This could lead to overestimated translation probabilities for common words. In Chapter 4, we will discuss some methods to alleviate this problem, including a regularized translation model [26] and a symmetric translation model [25].

3.1.2 Cross Media Relevance Model

The relevance language model was originally designed for ad-hoc information retrieval[31] and cross-language information retrieval[30] to determine the appropriate language model, namely the probability of observing a word w in the documents relevant to a specific information need $p(w|R)$. The key in estimating the query language model is to determine which documents are likely to be relevant to a given query. Assuming query words are sampled independently from a multinomial distribution, the probability for a word to be relevant to an information need could be approximated as:

$$P(w|R) \approx \sum_{M} P(M)P(w|M)\prod_{i=1}^{k} P(q_i|M) \tag{3.4}$$

Here M is one of the document language models that are estimated from a collection of documents, w is a query word, and R stands for the relevance language model for a given query.

The relevance language model can be extended to image annotation and image retrieval tasks, called 'Cross Media Relevance Model', or CMRM for short. Each annotated image in the training set is assumed to be a candidate model M in Equation 3.4. Given an image $I = b_1, b_2, ..., b_n$ to be annotated, following the relevance language model, probability $p(w_k = 1|I)$ can be estimated as:

$$p(w_k = 1|J) \propto p(w_k = 1, I) = \sum_{i=1}^{|T|} p(w_k = 1, I, J_i) \approx \frac{1}{|T|}\sum_{i=1}^{|T|} p(w_k = 1|J_i)\prod_{i=1}^{m} p(b_j|J_i) \tag{3.5}$$

where $p(J_i)$ is set to be a uniform distribution. Both $p(w_k = 1|J_i)$ and $p(b_j|J_i)$ are assumed to be multinomial distributions and are computed using the Jelinek-Mercer smoothing approach[10].

$$\begin{aligned} p(w_k = 1) &= (1 - \alpha_J)\frac{\#(w_k, T)}{|J_i|} + \alpha_J\frac{\#(w_k, T)}{|T|} \\ p(b_j|J_i) &= (1 - \beta_J)\frac{\#(b_j, J_i)}{|J_i|} + \beta_J\frac{\#(b_j, T)}{|T|} \end{aligned} \tag{3.6}$$

$\#(w_k, J_i)$ is the frequency of word w_k in the annotated image J_i, and $\#(w_k, T)$ is the number of words in the collection. $\#(b_i, J_i)$ is the frequency of blob b_i in the annotated image J_i, and $\#(b_i, T)$ is the number of blob b_i in the collection. α_J and β_J are smoothing parameters and are determined by the held out data set. The essential idea of the relevance language model for automatic image annotation is to propagate the words for the annotated images to a test image based on their similarity to the training images.

Another usage of the relevance language model is to apply it to image retrieval for textual queries. To bridge the gap between textual queries and image features, the above relevance language model will be first applied to generate a 'visual' language model for blobs based on the set of annotated images and the query words. Then, the estimated 'visual' language model is used to determine the relevance of images in the collection. More detailed description can be found in [24].

One obvious difference between the relevance language model and the translation model is that the translation model assigns annotation words to different image regions while the relevance language model only acquires annotation words for the entire images. Thus, unlike the translation model for automatic image annotation, which requires appropriate alignment between image regions and annotation words, the relevance language model avoids explicitly modeling the correlation between image blobs and annotation words, which makes it easy to implement and robust in practice.

3.1.3 Latent Semantic Analysis for Image Annotation

Latent Semantic Analysis(LSA) originates from textual retrieval and document analysis[13]. It maps a high dimensional representation of a document, which often is the term frequency vector of the document, into a low dimensional space, which is also called the latent semantic space[13]. On one hand, latent semantic analysis can be viewed as a dimension reduction method, and therefore is effective for alleviating the sparse data problem that commonly exists in text-related applications. On the other hand, latent semantic analysis is able to represent the document information beyond the lexical level. By aggregating related words into concepts based on word co-occurrence patterns, LSA is able to capture certain semantic correlations among words such as synonyms.

LSA is based on the Singular Value Decomposition(SVD). Suppose X is term document matrix of dimensionality $t \times d$, where t is the number of documents and d is the size of vocabulary. $X_{i,j}$ is the frequency of word j in the document i. X can be decomposed into three matrices using SVD as

$$X = T_0 S_0 D_0^{'} \tag{3.7}$$

Here T_0 and D_0 are orthonormal matrices and S_0 is a diagonal matrix. If we keep the first k largest singular values in S_0 and set the others to be 0, we obtain another matrix $\hat{X} = T_0 \hat{S}_0 D_0^{'}$ that minimizes the quantity $|X - \hat{X}|^2$, where $\hat{X}$ is a matrix of rank k. This operation also removes the corresponding columns of T_0 and D_0 respectively and get the matrices T and S. The columns in the matrices T and S form the base vectors for the latent space of documents and terms respectively. We can then map each document and each term into this space and compute their similarities.

There is a strong analogy between textual documents and images. In textual documents, the word sense ambiguity exits in two forms: the Polysemy, where a word corresponds to multiple different meanings, and Synonym, where multiple words correspond to the same meaning. Similarly, in image domain, the same distribution of image features can be related to different objects under different contexts. Meanwhile, the same object can show different visual properties that lead to different distributions in the space of image features. Because of this analogy, the latent semantic analysis, which has demonstrated to be a powerful tool in document analysis, can be applied to automatic image annotation[37]. In [37], color histograms are used to represent images. Each entry in the histogram of every segment is viewed as a different 'visual' word, totally there are 648 different 'visual' terms in the 'visual' vocabulary. Then, the LSA is applied to reduce the representation of images to the latent space, which has fewer dimensions. The annotation of test images is computed as the average annotations of training images that are weighted by their similarity to the test image in latent space.

3.1.4 Probabilistic Latent Semantic Analysis for Image Annotation

As pointed out in [22], LSA has difficulties in capturing the concept of polysemy, which refers to the case when a word has multiple senses. The author proposed the Probabilistic Latent Semantic Analysis (PLSA) for co-occurrence data, which introduces unobservable variables, i.e., latent topics or latent classes, to capture the correlation among words based on co-occurrence statistics between documents and words [22].

Let $p(d_i)$ denote the prior probability for document d_i, $p(z_k|d_i)$ denote the probability for document d_i to be in latent class z_k, $p(w_j|z_k)$ denote the probability of observing word w_j in latent class z_k. Then, the joint probability $p(d_i, w_j)$ can be written as:

$$\begin{aligned} p(d_i, w_j) &= p(d_i)p(w_j|d_i) \\ p(w_j|d_i) &= \textstyle\sum_{k=1}^{K} p(w_j|z_k)p(z_k|d_i) \end{aligned} \tag{3.8}$$

Using the equation in (3.8), given a document collection $d_1, d_2, ..., d_N$, where N is the number of documents, its log-likelihood could be written as:

$$\begin{aligned} L &= \textstyle\sum_{i=1}^{N} \sum_{j=1}^{M} n\left(d_i, w_j\right) \log p(d_i, w_j) \\ &= \textstyle\sum_{i=1}^{N} n\left(d_i\right) \left[\log p(d_i) + \sum_{j=1}^{M} \frac{n(d_i, w_j)}{n(d_i)} \log \sum_{k=1}^{K} \left(p(w_j|z_k)p(z_k|d_i)\right)\right] \end{aligned} \tag{3.9}$$

, where $n(d_i, w_j)$ is the term frequency of word w_j in document d_i. $n(d_i) = \sum_j n(d_i, w_j)$ is the length of document d_i. The EM algorithm is used to learn the values for parameters $p(z_k|d_i)$ and $p(w_j|z_k)$ [22].

[38] extends the application of PLSA from textual collection to automatic image annotation. Given an image q, the probability to annotate the image with word w_j could be written as:

$$p(w_j|q) = \sum_{k=1}^{K} p(w_j|z_k)p(z_k|q) \tag{3.10}$$

[38] also considers that each word encodes more information than image features, and so words should play more important role in deciding the structure of the latent space than image features. Based on this intuition, a variation of PLSA model, named PLSA-word, is proposed, in which words and image feature are trained separately in the latent space. More specifically, the learning procedure is divided into two steps:

1. Based on Equation (3.9), a PLSA model is trained only on the annotation word sets to obtain the probability distribution of latent class z given an annotated image d, i.e., $P(z|d)$, and the probability distribution of words w given the latent class z, i.e., $P(w|z)$.

2. Fix $P(z|d)$, train another PLSA model to obtain the probability distribution of visual feature v given latent class z, i.e., $P(v|z)$.

To annotate a test image, the following two steps are performed.

1. Based on the visual features v of the test image and the distribution $P(v|z)$, the likelihood for the test image to be in latent class z is computed as $P(z|d)$.

2. Using the word distribution of each latent class, i.e., $P(w|z)$, the probability of annotating the test image with a word w is computed as:

$$p(w|d) = \sum_k p(w|z_k)p(z_k|d) \tag{3.11}$$

3.2 Image Annotation Based on Continuous Features of Image Regions

Previously, we reviewed statistical models for image annotation based on the discrete representation of image regions. By grouping different image segments into a small number of clusters, each image segment is mapped to a discrete id. This discretization procedure simplifies the image representation and hence reduces the computational cost. However, one disadvantage of the discretization procedure is that it loses valuable information encoded in the image features. There have been many studies on automatic image annotation that directly uses the raw image features extracted from image regions.

3.2.1 Hierarchical Model for Image Annotation

The semantic meaning of different words could differ significantly in their generality. For example, the word 'animal' is more general than the word for a specific animal such as 'tiger'. To account for different generality of different words, we can organize words into a hierarchical structure such that general words are put on the top of the structure and the words with specific meaning are put close to the leave nodes of the structure. For example, word 'animal' will be put as one of the parent nodes for the word 'tiger'. In document analysis, there have been a few studies on the hierarchical models[21, 23], which extracts the hierarchical relations between documents and abstract organization of keywords.

Similar to textual words, the patterns of image features could be organized into a hierarchy with each textual word corresponding to a different visual pattern. For example regions for 'sky' appear more commonly than regions for 'tiger'. Thus, the goal is to organize both the textual words and image regions into a hierarchical structure. The general blobs and words are put at the top levels of the hierarchy that are close to the root, and the blobs and words with specific meaning are put on nodes close to the leaves. Annotated images are put to the leaf nodes of the hierarchy, which are considered as clusters. The hierarchical structure defines a path along which each annotated image generates its image blobs and annotation words[3, 28]. Refer to paper [3, 28] for an excellent figure to illustrates this procedure.

Because of the uncertainty in determining the cluster membership for an annotated image, to compute the probability to generate the observations D (i.e., words, and image regions) of an annotated image d, we need to sum over the uncertainty in the distribution of assigning annotated image d to different clusters, i.e.,

$$p(D|d) = \sum_{c} p(c) \prod_{i \in D} \sum_{l} p(i|l,c)p(l|c,d) \tag{3.12}$$

,where $p(l|c,d)$ is the probability of going through level l given the document d and cluster c. $p(i|l,c)$ is the probability to generate the observation i given the level l and cluster c.

The EM algorithm could be used to train the parameters in the likelihood function. The probability of generating observations of words $p(i|l,c)$ is computed based on counting. A Gaussian distribution is used to compute the probability of generating observations of image regions. More information about the application of the EM algorithm to the hierarchical structures could be found in [21, 23].

Given the learned hierarchical structure, the probability of annotating an image B with a word w, i.e. $p(w|B)$, is written as:

$$\begin{aligned} p(w|B) &= \textstyle\sum_c p(w|c,B)p(c|B) \propto \sum_c p(w|c,B)p(B|c)p(c) \\ &= \textstyle\sum_c p(w|c,B)\prod_{b\in B} p(b|c)p(c) \\ &= \textstyle\sum_c \sum_l p(w|c,l)p(l|c,B)\prod_{b\in B}\left(\sum_l (p(b|l,c)p(l|c))\right)p(c) \end{aligned} \tag{3.13}$$

However, the above model is not a true generative model, since the probability of generating words and image segments rely on the specific document. In [1], three different hierarchical models have been developed. They are:

- Model I-0.

  This model is defined as:

$$p(D|d) = \sum_c p(c) \prod_{w\in W} \left[\sum_l p(w|l,c)p(l|d)\right]^{\frac{N_w}{N_{w,d}}} \prod_{b\in B} \left[\sum_l p(b|l,c)p(l|d)\right]^{\frac{N_b}{N_{b,d}}} \tag{3.14}$$

- Model I-1.

  In this Model, the probability of generating level $p(l|d)$ in Model I-0 is replaced with $p(l|c,d)$, which also depends on the cluster the document belongs to. The generation probability of observations is changed to:

$$p(D|d) = \sum_c p(c) \prod_{w\in W} \left[\sum_l p(w|l,c)p(l|c,d)\right]^{\frac{N_w}{N_{w,d}}} \prod_{b\in B} \left[\sum_l p(b|l,c)p(l|c,d)\right]^{\frac{N_b}{N_{b,d}}} \tag{3.15}$$

- Model I-2.

Model I-2 further modifies the probability of generating level l, and makes it independent of the documents and only dependent on the cluster. The probability of observation is thus changed to:

$$p(D|d) = \sum_{c} p(c) \prod_{w \in W} \left[\sum_{l} p(w|l,c)p(l|c) \right]^{\frac{N_w}{N_{w,d}}} \prod_{b \in B} \left[\sum_{l} p(b|l,c)p(l|c) \right]^{\frac{N_b}{N_{b,d}}} \tag{3.16}$$

3.2.2 Latent Dirichlet Allocation for Image Annotation

Multi-type data problem is a general problem in which each object/pattern is represented by different types of data. Annotated images can be viewed as multi-type data, in which there are two types of data, the image regions and annotation words. These two types of data should correlate with each other since they describe the same images. In [5], three models are introduced to solve the multi-type data problem. In these models, image features are assumed to follow a multivariate Gaussian distribution with diagonal covariance matrix. The words are assumed to follow a multinomial distribution.

- Model 1: Gaussian multinomial mixture model

  In this model, each latent variable is an indicator of cluster, both words and image regions are considered as different types of data that are generated from the same latent variable. Refer to [5] for a figure to illustrate the generation procedure of this model. The joint probability for latent class z, annotated words $\overrightarrow{w} = \{w_1, w_2, ..., w_M\}$, and image regions $\overrightarrow{r} = \{r_1, r_2, ..., r_N\}$, i.e.,$p(z, \overrightarrow{r}, \overrightarrow{w})$, could be formally represented as:

  $$p(z, \overrightarrow{r}, \overrightarrow{w}) = p(z|\lambda) \prod_{n=1}^{N} p(r_n|z, \mu, \sigma) \prod_{m=1}^{M} p(w_m|z, \beta) \tag{3.17}$$

  where μ and σ govern the Gaussian distributions that determine the generation probabilities of image regions, and β governs the multinomial distributions that determine the generation probabilities of annotation words.

To predict words for a given image, we first compute the probability that the image belongs to a latent class z based on the image region features and then words are generated from the latent class z. Finally, the words from different clusters are mixed together to obtain the annotation. This is formally described as:

$$p(w|r) = \sum_{z} p(z|r)p(w|z) \tag{3.18}$$

- Model 2: Gaussian Multinomial Latent Dirichlet Allocation

In GMM model, the generation of the words and image features is conditioned on the same latent variable. A more flexible model is Latent Dirichlet Allocation (LDA). LDA was first developed for document clustering[6]. Each document is considered to consist of several topics and observations in one document are generated from these different topics. For image collection, we could also view an image as consisting of several different topics. Each of the topics will generate the corresponding words and image regions belonging to the topic. The overall observation is a mixture from these different topics. Refer to [6] for this graphical model.

Let latent variables z and v model the different topics from which words and image regions are generated. A variable θ is used to indicate that the two types of topics for words and image regions are correlated. Also, it models the concept that a single document could consist of multiple topics, which are sampled based on the variable θ. The formal representation of this generation procedure is expressed as:

$$p(r, w, \theta, z, v) = p(\theta|\alpha) \left(\prod_{n=1}^{N} p(z_n|\theta)p(r_n|z_n, \mu, \sigma) \right) \left(\prod_{m=1}^{M} p(v_m|\theta)p(w_m|v_m, \beta) \right) \tag{3.19}$$

- Model 3. Correspondence LDA.

While the GMM model is too restrictive and LDA is too flexible, a model is proposed to position between the two kinds of models. Due to the flexible and correlated relation between image regions and words, correspondence LDA is introduced to model the correspondence between specific words and image segmentations. That is, the generation of words is based on both the latent variables to generate the words as in LDA and the latent variables to generate the image segmentations. Refer to [6] for a graphical model to illustrate this procedure.

The formal representation of the model is

$$p(r,w,\theta,z,y) = p(\theta|\alpha)\left(\prod_{n=1}^{N} p(z_n|\theta)p(r_n|z_n,\mu,\sigma)\right)\left(\prod_{m=1}^{M} p(y_m|N)p(w_m|y_m,z,\beta)\right) \tag{3.20}$$

- Further comparison of the different models.

  We could make a further discussion of the three models based on latent variables: Gaussian Multinomial Mixture Model (GMM), Probabilistic Latent Semantic Analysis (PLSA), and Latent Dirichlet Allocation (LDA). From the following discussion, we could see that these three models differ mainly in how they sample the topics. The topics are modeled by latent variables.

  In GMM, the probability of sampling a topic is fixed by the latent variable. In PLSA, the sampling of the topic depends on each of the documents. In LDA, the flexibility of sampling of the latent variable is between GMM and PLSA. The topic probability depends on the parameters sampled out from a Dirichlet distribution but not on the document. From this perspective, LDA is more restrictive than PLSA. However, the capability to change the parameters of topic sampling based on Dirichlet distribution also provides more flexibility than GMM.

3.2.3 Continuous Relevance Model

Previously, we discussed the application of the discrete relevance model to image annotation. The related words to a test image could be predicted by aggregating annotation words of labeled instances in the training set weighted by the visual similarity between test image

and the labeled image. The continuous relevance model is discussed in[32]. The difference is to use a different method to generate the image segments and words.

The conditional probability of generating words for given images is approximated by the joint probability of observing the image regions and words:

$$p(r_A, w_B) = \sum_{J \in T} p_T(J) \prod_{b=1}^{n_B} p_v(w_b|J) \prod_{a=1}^{n_A} \int_{R^k} p_R(r_a|g_a) p_G(g_a|J) dg_a \tag{3.21}$$

$p_R(r_q|g_a)$ is a global probability distribution responsible for mapping generator vectors $g \in R^k$ to actual image regions $r \in R$. In [32], it is assumed that for every image there is only one corresponding generator. So a particularly simple form for the distribution P_R is assumed to be:

$$P_R(r|g) = \begin{cases} \frac{1}{N_g} & if G(r) = g \\ 0 & otherwise \end{cases}$$

,where N_g is the number of all regions in R.

Gaussian distribution $p_G(g|J) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\sqrt{2^k \pi^k \det(\Sigma)}} \exp\left(g - G(r_i)^T \Sigma_{-1}(g - G(r_i)\right)$ is used to generate the image features from a model J. $G(r_i)$ is the feature vector of every region of image J. In this model, it is assumed that the feature vector g in given test image could be generated from every region r_i in the image J following a Gaussian distribution with r_i as the mean and a diagonal matrix as the covariance matrix.

The word annotation probability is estimated based on multinomial distribution with Dirichlet smoothing. The expression for this Bayesian estimation of posterior of word probability is

$$p_v(v|J) = \frac{\mu p_v + N_{v,J}}{\mu + \sum_{v'} N_{v',J}} \tag{3.22}$$

In [32], the model probability $p(J)$ is assumed to be uniform.

3.2.4 Classification Method

The annotation procedure can be viewed as a classification problem. A classifier is trained for each word and this classifier is then used to determine whether the word should appear in the annotation set of a given image. These trained classifiers are then applied to both labeled and unlabeled images to get a vector, which includes all the concepts predicted from classifiers for each of the concepts. The values of the components of the vector are confidence values obtained from the classifier[9]. The application of the multiple binary classifiers for image annotation task could be viewed as an ensemble of binary classifiers with a method of One-Per-Class(OPC). This is a simple ensemble strategy compared to the pair-wise combination and Error Correct Output Coding(ECOC)[29]. However, the way OPC used in [9] is not exactly the traditional usage of OPC. In the traditional usage of OPC, the class label with the maximum score is used to annotate the image. In [9], the confidence to annotate the image with one concept is computed by adding the scores of different ensembles for this concept. The framework is called Content-Based Soft Annotation (CBSA). [9]uses Support Vector Machine(SVM) and Bayesian Point Machine as the classifiers.

Support Vector Machine tries to find a hyperplane that separates the training data with maximum margin. The points closest to the hyperplane are called support vectors[8]. The Bayesian Point Machine approximates the Bayesian average of statistical inference with a unique classifier called Bayesian Point[20]. Bayesian inference gives a Bayesian optimal solution for a classification task. However, it is often impossible to get a unique classifier that has the same result as Bayesian inference. An approximation is made by finding a hypothesis in a fixed smaller space. The hypothesis constructed is called Bayes Point, which is believed to well approximate the Bayesian inference. This Bayes Point Machine could be computed as the center-of-mass in the version space. [9] shows that the classifier with Bayesian Point Machine performs better than the classifier of Support Vector Machine. Also, for the words with prediction score higher than 0.5, the results are very likely related to the semantics of the pictures.

3.2.5 Bootstrapping and Co-training Approach for Image Annotation

All of the previous models have certain limitations. They require a large number of annotated images for effective learning, which are often difficult to obtain. Furthermore, most existing techniques for automatic image annotation require semantically meaningful segmentation of images, which is again difficult to accomplish given the current state-of-art segmentation techniques. [15] proposes using bootstrapping and co-training methods to annotate large image collections. The basic idea is to start with a small set of annotated examples, and successively annotate and learn from a large set of unlabeled examples. To incorporate more instances from the unlabeled sets for the training purpose, two statistically independent classifiers are used to co-annotate the images and the quality of the final result is determined by the consistency of annotations. This procedure is called co-training. The consistent labels are considered as good examples and added to the training set. The inconsistent labels are considered as bad examples and are used for the user's manual annotation. More precisely, the co-training approach [15] is carried out as follows. Two SVM classifiers are used to determine the words for images. Each SVM uses a different subset of image features extracted from automatically segmented image regions. One set contains the color histogram and another includes texture and shape features. These SVMs trained on different kinds of features for different segmented regions are then applied to annotate images. If the annotated words could agree with each other, the words will be used for annotating the image. If there is some conflict between different classifiers, some disambiguation rules are applied to pick the correct one. If the conflict cannot be resolved, the image is prompted for the user to annotate manually. At the same time, the training set is enlarged and the classifiers are updated.

3.2.6 Linguistic Indexing of Pictures by a Statistical Modeling Approach

Most of the previous methods require semantically meaningful segmentation to correlate the segmented regions with words. However, given the current segmentation techniques, semantically meaningful regions are difficult to obtain. A number of papers propose approaches that do not need to get the meaningful segmentation but still effectively model the structures of the images[33].

In [33], image features are modeled as 2 dimensional Multi-resolution Hidden Markov Model (2D MHMM). Images are divided into blocks and represented in multi-resolution. The number of blocks in both rows and columns are reduced successively by half to get lower and lower resolutions. At each resolution level, features are generated by a 2D HMM, which takes into account the spatial relationship between blocks at the specific resolution by specifying the diagonally upper blocks as the previous blocks. For every state in the 2D MHMM, the feature vectors are assumed to follow a multivariate Gaussian distribution. To model the relationship across different resolutions, a one dimensional Markov chain is used to connect the related parent-children blocks at different resolutions and thus a 2D MHMM is formed to represent the information of the image. The 2D MHMM model mainly captures the spatial relationship of the feature vectors, for both inter-scale and intra-scale statistical dependence. The number of states along horizontal and vertical directions is unknown and thus different combinations are tried. Refer to [33] for a figure to illustrate this idea.

A dictionary of concepts is defined as the possible set of annotation words. For each of the concepts in the dictionary, a set of images is collected and also a short description of the concept is given. It is easier to collect a set of images related to a concept with a short description than provide a set of annotations for each of the image. Given this set of images, a 2D MHMM is trained for each of the concepts.

Given a test image, the similarity score between the image and images in the training set is computed as the likelihood of generating this test image with the 2D MHMM for the concept of the category. The top k categories with the highest likelihood are picked out and the word descriptions of the categories are treated as the candidates for annotating the test image. The paper introduces a method to pick appropriate words by ranking the statistical significance of the words in the candidate description. The statistical significance is computed by comparing the occurrence of the word in the predicted k categories with randomly selected k categories. The top ranked words are selected as the annotation of the image.

CHAPTER 4

Enhanced Translation Model for Automatic Image Annotation

Feng Kang, Rong Jin and Joyce Y. Chai. Regularizing translation models for better automatic image annotation. In *CIKM '04: Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management*, pages 350-359, 2004.

Feng Kang and Rong Jin. Symmetric statistical translation models for automatic image annotation. In *Proceedings of the fifth SIAM International Conference on Data Mining*, pages 616-620, 2005.

CHAPTER 4

Enhanced Translation Model for Automatic Image Annotation

The translation model in section 3.1.1 represents images by a visual vocabulary that is constructed from the clustering results of image regions. The image annotation is viewed as a translation from a visual language to a textual language. The key challenge in training the translation model arises from the fact that the alignment between image regions and their annotation words is not provided. According to [7], one strategy to avoid the alignment problem is to utilize the co-occurrence statistics between image regions and annotation words. If an image blob co-occurs more frequently with word 'A' than with other words, it will be more likely for the image blob to be associated with 'A'.

However, the term frequency of annotated words follows Zipf's law, namely that a small number of words appear very often in image annotations and most words are used only by a few images. Figure 4.1 plots the percentage of times that each word is used by image annotations from a subset of images from the COREL dataset[14]. As a result of the skewed distribution, a common word can 'accidentally' co-occur with a blob that is associated with a rare word. For example, word 'grass' is used for annotations much more frequently than word 'tulip'. Meanwhile, for most images where tulip appears, it is always surrounded by grass. As a result, the type of blobs for tulip co-occurs with word 'grass' as frequently as word 'tulip'. The problem with the co-occurrence statistics in automatic image annotation

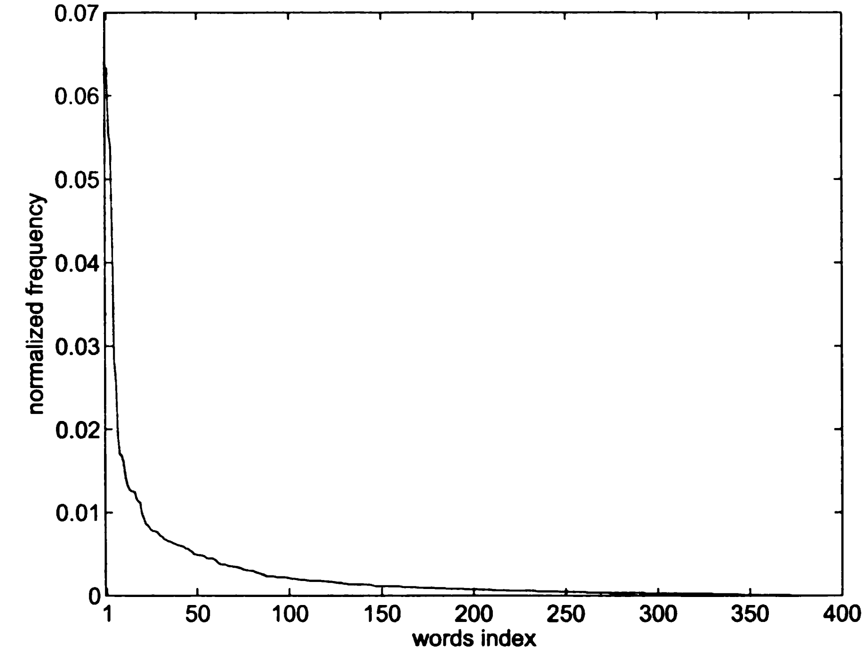


Figure 4.1. The distribution of term frequency for annotated words in a subset of COREL Data

is further complicated by the noise in clustering the massive number of image regions into a relatively small number of blobs. Because each image region is represented by a set of fixed features such as colors, textures and shapes, regions for different annotated words can have similar distributions over the space of image features and therefore are grouped into the same cluster, or the same image blob. As a result, a blob for a rare word can co-occur more frequently with a common word than the rare word. Using the previous example, consider that image regions for tulips are grouped together with image regions for other flowers. If most flowers are surrounded by the grass, word 'grass' will co-occur more frequently with the blob for flowers than any single flower name. It is the inaccurate co-occurrence statistics that allow common annotated words to be associated with many irrelevant image blobs and thus degrade the quality of auto-annotations generated by the machine translation models. We propose two categories of modified translation models, namely the regularized translation model, and the symmetric translation model, that alleviate the problem caused by the skewed distribution. The basic idea of the enhanced models is to raise the number of blobs that are associated with uncommon words. The regularized translation model accomplishes this goal through the introduction of a special Dirichlet prior and the symmetric translation model considers both the translation from blobs to words and the translation from words to blobs. In this chapter, we will first present the two frameworks and then show our empirical study with both modified translation models. Chapter 5 addresses their comparison with other models.

4.1 Regularized Translation Model

In this section, we will first present the framework of the regularized translation model for automatic image annotation, followed by the description of an efficient EM algorithm for finding the optimal solution[26].

4.1.1 Framework for the Regularized Translation Model

The basic idea of the regularized translation model is to impose our prior knowledge of the desired translation model in the selection of translation models. If each blob represents a different type of objects, we would expect that roughly equal number of blobs are associated with each word, or at least each word is associated with a certain number of blobs. In the framework of Bayesian Learning, this prior preference of translation models can be introduced into the original statistical translation model through an appropriate prior.

In order to form such a prior, the first task is to find an appropriate measurement that indicates the number of blobs that are associated with each word. We use the normalized sum of translation probabilities for each word, i.e.,$\beta_j = \frac{\sum_k t_{j,k}}{m}$ where m is the total number of blobs. Since the measurement β_j is proportional to the sum of translation probabilities for the $j-th$ word, it does provide a good indication of how many blobs that are associated with the word j. Meanwhile, β_j can also be interpreted as the probability for the $j-th$ word to be associated with any blobs. Particularly, β_j satisfies the axioms for probability, namely

1. $0 \leq \beta_j \leq 1$
2. $\sum_j \beta_j = 1$

Because of the probability interpretation for β_j, we can introduce a prior distribution for β_j that will indirectly influence the results for translation probabilities $t_{j,k}$.

The second task for forming a prior is to choose an appropriate distribution for β_j. Since the desired translation model is to have almost equal number of blobs to be associated with each word, a Dirichlet prior can be used for $\vec{\beta}$,

$$P(\vec{\beta}) \sim Dirichlet(\vec{\beta}, \alpha) = \frac{1}{\text{ß}(\alpha)} \prod_{j=1}^{n} \beta_j^{\alpha-1} \tag{4.1}$$

where $\alpha > 0$ is the hyper-parameter that determines the shape of the Dirichlet distribution and n is the total number of words. ß is the normalization constant and follows multinomial beta distribution as:

$$\text{ß}(\alpha) = \frac{(\Gamma(\alpha))^n}{\Gamma(n\alpha)}$$

Γ is Gamma function defined as $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$. Note that Dirichlet distribution is the conjugate prior of the multinomial distribution.

In our model, we set α_i as a constant. One property of Dirichlet distribution is that it reaches the maximum point when β_j is a constant. Furthermore, the larger the α is, the narrower the distribution will be. Figure 4.2 illustrates the Dirichlet distribution with two β component and different values of the parameter α.

By adding the above prior to the original translation model, the posterior probability for the training images is then modified into the following form:

$$l_{reg}(T) = P(\vec{\beta}) \prod_{i=1}^{T} p(\vec{w_i}|\vec{b_i}) \propto \prod_{l=1}^{n} \left(\sum_{s=1}^{m} t_{l,s} \right)^{\alpha} \prod_{i=1}^{T} \prod_{j=1}^{n} \sum_{k=1}^{m} (t_{j,k} b_{i,k}) \tag{4.2}$$

With same notation as the previous translation model, m is the number of blobs, n is the number of words, and T is the number of images in the training set.

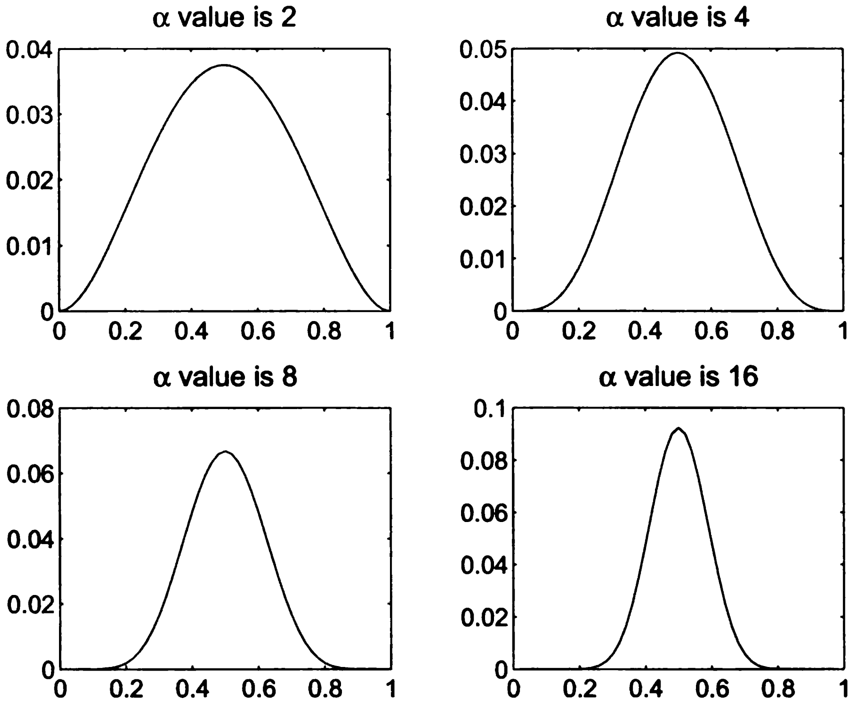


Figure 4.2. Dirichlet distribution for different values of α

The optimal translation probabilities are obtained by maximizing the objective function in Equation 4.2. Compared to the original translation model in Equation 3.2, the objective function in Equation 4.2 requires that not only should the optimal translation probabilities explain well the correspondence between image blobs and annotated words but also be consistent with the prior preference on translation models, namely that different words are associated with a similar number of image blobs. Therefore, the resulting translation model from Equation 4.2 will be more desirable than the model obtained from Equation 3.2. For late reference, we call this modified translation model 'regularized translation model', or *RTM*.

4.1.2 An EM Algorithm for the Regularized Translation Model

With the regularized translation model in Equation 4.2, the next important question is how to efficiently obtain the optimal translation probabilities $t_{j,k}$ that maximize the function in

Equation 4.2. The difficulty with optimizing Equation 4.2 lies in two aspects:

1. It has a large number of parameters. The number of parameters in translation models is $m \cdot n$, i.e., the number of blobs times the number of unique words. For the experiment conducted in this study, the number of parameters is close to 20,000.
2. It is a constrained optimization problem. The optimal solution to Equation 4.2 should satisfy the axioms of probability, namely $0 \leq t_{j,k} \leq 1 \forall j, k$ and $\sum_j t_{j,k} = 1 \forall k$.

In the following, we present an EM algorithm for efficiently optimizing the objective function in Equation 4.2.

First, instead of optimizing the likelihood of training data in Equation 4.2, we can optimize the log-likelihood of training data, i.e.,

$$\Phi = log\left(l_{reg}(T)\right) = \alpha \sum_{l=1}^{n} log\left(\frac{\sum_{s=1}^{m} t_{l,s}}{m}\right) + \sum_{i=1}^{|T|}\sum_{j=1}^{n} w_{i,j} log\left(\sum_{k=1}^{m} t_{j,k} b_{i,k}\right) \tag{4.3}$$

Then, following the idea of the EM algorithm, we update the optimal solution iteratively. Particularly, at each iteration, we need to find a set of translation probabilities $t_{j,k}^{new}$ better than the old ones $t_{j,k}^{old}$ that are computed for the previous iteration. To this end, we can examine the difference in the log-likelihood between two consecutive iterations, i.e.,

$$\Phi - \Phi' = \alpha \sum_{l=1}^{n} log\left(\frac{\sum_{s=1}^{m} t_{l,s}^{new}}{\sum_{s=1}^{m} t_{l,s}^{old}}\right) + \sum_{i=1}^{|T|}\sum_{j=1}^{n} w_{i,j} log\left(\frac{\sum_{k=1}^{m} t_{j,k}^{new} b_{i,k}}{\sum_{k=1}^{m} t_{j,k}^{old} b_{i,k}}\right)$$

$$\geq \alpha \sum_{l=1}^{n}\sum_{s=1}^{m} \frac{t_{l,s}^{old}}{\sum_{s=1}^{m} t_{l,s}^{old}} log\left(\frac{t_{l,s}^{new}}{t_{l,s}^{old}}\right) + \sum_{i=1}^{|T|}\sum_{j=1}^{n} \frac{w_{i,j} t_{j,k}^{old} b_{i,k}}{\sum_{k=1}^{m} w_{i,j} t_{j,k}^{old} b_{i,k}} log\left(\frac{t_{j,k}^{new}}{t_{j,k}^{old}}\right)$$

The new translation probabilities $t_{j,k}^{new}$ are obtained by maximizing the above difference. Taking derivative and setting it to 0, the updating equation becomes:

$$t_{j,k}^{new} = \frac{1}{Z_k}\left(\alpha \frac{t_{j,k}^{old}}{\sum_l t_{j,l}^{old}} + \sum_i \frac{w_{i,j} t_{j,k}^{old} b_{i,k}}{\sum_{k=1}^{m} w_{i,j} t_{j,k}^{old} b_{i,k}}\right) \tag{4.4}$$

where Z_k is a normalization factor that ensures $\sum_j t_{j,k}^{new} = 1$.

Comparing the above updating equation to Equation 3.3, we can see that Equation 4.4 has an extra term $\alpha\frac{t_{j,k}^{old}}{\sum_l t_{j,l}^{old}}$ in the right hand side of the equation. For each word j, this extra term gives a larger share of α to $t_{j,k}$, the maximum translation probability for the $j-th$ word, than any other translation probabilities $t_{j,l}, l \neq k$ for the same word. As a result, the maximum translation probability for each word will benefit most from this extra term. Furthermore, the updating equation 4.4 is able to adjust the sum of translation probabilities for different words (i.e.,$\sum_k t_{j,k}$) to be close. This is because according to Equation 4.4, a word j with a small sum of translation probabilities, which is the denominator of $\alpha\frac{t_{j,k}^{old}}{\sum_l t_{j,l}^{old}}$, will get more promotion than a word that has a large sum of translation probabilities. Usually, the rare words co-occur with fewer blobs than the common words and the sum of translation probabilities related to rare words is thus smaller and gets more promotion.

Comparison to the normal usage of Dirichlet priors. Note that the Dirichlet prior introduced in this work is different from the Dirichlet priors used in many other studies [49]. For most previous studies of Bayesian learning, Dirichlet priors simply introduce constant pseudo counts into the estimation of probabilities. However, here the pseudo count introduced by the Dirichlet prior (i.e., $\alpha\frac{t_{j,k}^{old}}{\sum_l t_{j,l}^{old}}$) is no longer a constant. In fact, it is this non-constant pseudo count that leads to a more balanced distribution of the number of blobs associated with each word.

The global optimum for the EM algorithm. The objective function in Equation 4.3 is strictly convex. It can be easily understood by treating each term $\left(\sum_{s=1}^{m} t_{l,s}\right)^{\alpha}$ in the prior as α number of pseudo-annotated images that include all blobs in its picture and are annotated only by $l-th$ word. As a result, the regularized model is almost identical to Translation Model 1 except that the regularized model uses both the pseudo-annotated images and the annotated images from the training dataset. Since the translation model for any number of annotated images is strictly convex [7], the new objective function in Equation 4.3 will be strictly convex. Therefore, it does not have any local optimum and the EM algorithm presented in Equation 4.4 will guarantee to find the global optimal solution.

The choice of α. As already revealed in the previous discussion, constant α has a great impact on the resulting translation model. A larger value for α will introduce more pseudo-annotated images and therefore result in a more balanced distribution for the number of blobs that is associated with each word. In the experiment part, we provide a detailed study of how the value of α will influence the quality of auto-annotations.

4.2 Symmetric Translation Model

Most previous studies on translation models for automatic image annotation focus on the model that translates image regions/blobs into textual words, which is called forward translation model in our view. Apparently, we can apply the translation model in a reverse way, namely translating textual words into image blobs. We call this translation model a backward translation model. In this section, we propose a symmetric translation by combining these two kinds of translation models together[25].

4.2.1 Discrepancy between Forward and Backward Translation Models

Although both forward and backward translation models utilize the same set of co-occurrence statistics, they make different assumptions for words and image blobs:

1. The forward translation model assumes that each image blob is translated into a single word, while each word can be translated into multiple blobs. Due to the problem with accidental co-occurrence, common words are usually associated with many more image blobs than uncommon words. In particular, many rare words are even associated with no image blobs at all, which makes it impossible for these words to be used as annotations.

2. The backward translation model is based on the assumption that each word is translated into a single image blob. Thus, for each annotation word, a large translation probability is assigned to an image blob if it co-occurs *relatlvely* frequently with the

Table 4.1. An example of forward and backward translation models for two image blobs(i.e. b_1 and b_2) and two words(i.e.w_1 and w_2).

<table>
<tr><th></th><th>w_1</th><th>w_2</th></tr>
<tr><td>b_1</td><td>100</td><td>50</td></tr>
<tr><td>b_2</td><td>200</td><td>35</td></tr>
<tr><td colspan="3">Forward translation model $p(w|b)$</td></tr>
<tr><td></td><td>w_1</td><td>w_2</td></tr>
<tr><td>b_1</td><td>0.67</td><td>0.33</td></tr>
<tr><td>b_2</td><td>0.85</td><td>0.15</td></tr>
<tr><td colspan="3">Forward translation model $p(b|w)$</td></tr>
<tr><td></td><td>w_1</td><td>w_2</td></tr>
<tr><td>b_1</td><td>0.33</td><td>0.58</td></tr>
<tr><td>b_2</td><td>0.67</td><td>0.42</td></tr>
</table>

given word compared to other image blobs. As a result, an image blob can be assigned with a large translation probability for a word although the ***absolute*** number of co-occurrence between the word and the image blob is rather small.

In order to better illustrate the difference between these two types of translation models, consider a simple example of co-occurrence statistics that is shown in Table 4.1. On one hand, for the forward translation model, the translation probabilities for word 'w1' are dominative for both image blobs, and word 'w2' is not strongly associated with any of the two blobs. As a result, it is unlikely for word 'w2' to be used in any auto-annotations. On the other hand, for the backward translation model, we do find that image blob 'b1' is strongly associated with word 'w2'. But, the chance for image blob 'b2' to be associated with word 'w2' is also high (i.e., 42%). This large uncertainty can significantly corrupt the accuracy of auto-annotations. Thus, neither of the two translation models provides a satisfactory answer.

To utilize the two directions of translation models, we can adjust the translation probabilities in Table 4.1. On one hand, for the forward translation model, to balance the number of blobs associated with different words, we need to increase probability $p(w_2|b_1)$, and decrease probability $p(w_1|b_1)$. This will result in word 'w2' to be associated with blob 'b1'. On the other hand, for the backward translation model, the uncertainty as to which blob is associated with word 'w2' can be reduced using the information from the forward translation model. This is because, according to the forward translation model, it is more likely for blob ‘b2’ to be associated with word ‘w1’ than for blob ‘b1’. Thus, to balance the number of image blobs associated with each word, we need to reduce translation probability $p(b_2|w_2)$ and increase $p(b_1|w_2)$. The refinement of the forward translation model and the backward translation model will be carried out iteratively until they both converge to the desired distribution.

As indicated by the above analysis, the key is to explore the correlation between the forward translation model and the backward translation model under the assumption that the number of blobs associated with different words should be evenly distributed. Using the information collected from the backward translation model, we are able to alleviate the problem with the forward translation model in which rare words are associated with few image blobs; Using the information collected from the forward translation, we are able to alleviate the problem with the backward translation model in which large uncertainty exists for uncommon words as to which image blob the word is associated with. By combining these two translation models together, we will be able to avoid the difficulties of each individual translation model that are caused by their underlying assumptions.

The mathematical formula for the backward translation model is similar to the forward translation model. In the backward translation model, the translation probability from annotation words to image blobs is written as:

$$p(\vec{b_i}|\vec{w_i}) = \prod_{j=1}^{m} p(b_{i,j}|\vec{w_i}) \propto \prod_{j=1}^{m} \left(\sum_{k=1}^{m} u_{j,k} w_{i,k} \right)^{b_{i,j}}$$

where $u_{j,k}$ stands for the probability of translating the $k-th$ word into the $j-th$ blob subject to the constraint $\sum_j u_{j,k} = 1$, namely each word has to be translated into a single image blob. Similarly, EM is used to find the set of translation probabilities and the updating equation is written as:

$$u_{j,k}^{new} = \frac{1}{Z_k} \sum_i \frac{w_{i,j} b_{i,k} u_{j,k}^{old}}{\sum_{k'} w_{i,k'} u_{j,k'}^{old}} \qquad (4.5)$$

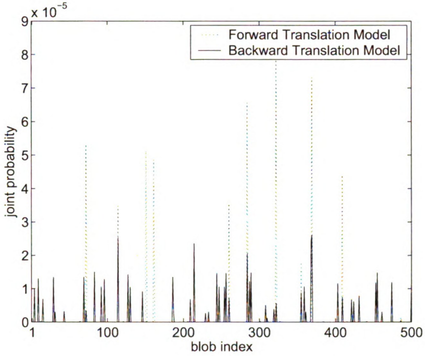


Figure 4.3. The joint probabilities $p(w, b)$ for word 'clouds' that are computed using the forward and backward translation models.

Z_k is a normalization factor that ensures $\sum_j u_{j,k}^{new} = 1$.

With this definition, we could illustrate the discrepancy between forward and backward models with the results trained using the COREL dataset(See the experiment part for the details of this dataset). Then, we compute the joint probability $p(w_j, b_k)$ for every word and every blob using both forward and backward translation models. More specifically, the joint probability based on the forward translation model is computed as:

$$p(w_j, b_k; f) = t_{j,k} p(b_k)$$

The joint probability for backward translation model is computed as:

$$p(w_j, b_k; b) = u_{k,j} p(w_j)$$

If the two translation models are consistent, we would expect the two joint distributions based on the two different translation models to be similar. However, as shown in Figure

4.3, the joint probabilities for word 'clouds' that are computed from two translation models are quite different. Furthermore, we also compute a relative KL divergence between the joint distributions based on the two translation models. It measures the relative difference between two distributions and is defined as:

$$Rel - KL = \frac{\sum_{j=1}^{n}\sum_{k=1}^{m} p(w_j, b_k; f) \log \frac{p(w_j, b_k; f)}{p(w_j, b_k; b)}}{\sum_{j=1}^{n}\sum_{k=1}^{m} p(w_j, b_k; f) \log p(w_j, b_k; f)}$$

The resulting value is 0.16286, which indicates that the overall difference between the two translation models is large.

In the following section 4.2.2, we will introduce a soft regularization term to enforce the consistency between the forward and backward translation models.

4.2.2 Symmetric Translation Model through Regularization

As discussed in the above section, there is a certain discrepancy between the translation probabilities of forward and backward translation model. A *regularization-based symmetric translation model (RSTM)* is proposed to correct this discrepancy by utilizing the information across the two models. First, we introduce a symmetric KL divergence term that measures the discrepancy between the forward and backward models:

$$KL = \sum_j \sum_k p(w_j, b_k; f) \log(\frac{p(w_j, b_k; f)}{p(w_j, b_k; b)}) + \sum_j \sum_k p(w_j, b_k; b) \log(\frac{p(w_j, b_k; b)}{p(w_j, b_k; f)}) \tag{4.6}$$

According to the property of KL divergence, the above expression becomes zero iff $p(w_j, b_k; f) = p(w_j, b_k; b)$ for any $j \in [1...n]$ and $k \in [1...m]$. Then, we add the KL divergence term into the objective function as the regularization term to ensure the consistency between the forward and backward translation models:

$$\Omega_{RSTM} = \underbrace{\{\sum_{i=1}^{|T|} \sum_{\{w_{i,j}=1\}} \log(\sum_{k=1}^{m} t_{j,k} b_{i,k}) + \sum_{i=1}^{|T|} \sum_{\{b_{i,k}=1\}} \log(\sum_{j=1}^{n} u_{k,j} w_{i,j})\}}_{translation} \tag{4.7}$$

$$\underbrace{-\lambda\{\sum_j\sum_k p(w_j,b_k;f)\log(\frac{p(w_j,b_k;f)}{p(w_j,b_k;b)})+\sum_j\sum_k p(w_j,b_k;b)\log(\frac{p(w_j,b_k;b)}{p(w_j,b_k;f)})\}}_{regularization}$$

where λ is used to determine the degree of consistency between the two models. Efficiently finding the optimal solution to (4.7) is more complicated than the EM algorithm for the standard translation model. We will give the updating equation in the following and leave the derivation in the next section.

$$t^{new}_{j,k} = \frac{2C_{j,k}}{B_{j,k} + \sqrt{B^2_{j,k} + 4A_{j,k}C_{j,k}}} \tag{4.8}$$

$$\begin{aligned} A_{j,k} &= 2\lambda\frac{p(b_k)}{t^{old}_{j,k}} \\ B_{j,k} &= \lambda p(b_k)\log(t^{old}_{j,k}) - \lambda p(b_k)\log(u_{k,j}p(w_j)) + \lambda_k \\ C_{j,k} &= \sum_{i=1}^{|T|}\frac{t^{old}_{j,k}b_{i,k}}{\sum_{k'=1}^{m} t^{old}_{j,k'}b_{i,k'}} + \lambda p(w_j)u_{k,j} \end{aligned}$$

where λ_k is the normalization factor that ensures $\sum_j t^{new}_{j,k} = 1$.

$$u^{new}_{k,j} = \frac{2F_{k,j}}{E_{k,j} + \sqrt{E^2_{k,j} + 4D_{k,j}F_{k,j}}} \tag{4.9}$$

$$\begin{aligned} D_{k,j} &= 2\lambda\frac{p(w_j)}{u^{old}_{k,j}} \\ E_{k,j} &= \lambda p(w_j)\log(u^{old}_{k,j}) - \lambda p(w_j)\log(t_{j,k}p(b_k)) + \beta_j \\ F_{k,j} &= \sum_{i=1}^{|T|}\frac{u^{old}_{k,j}w_{i,j}}{\sum_{j'=1}^{n} u^{old}_{k,j'}w_{i,j'}} + \lambda p(b_k)t_{j,k} \end{aligned}$$

where β_j is the normalization factor that ensures $\sum_k u^{new}_{k,j} = 1$.

Note that the translation probability $u_{k,j}$ for the backward translation model is utilized in the estimation of $t_{j,k}$ through the computation of $B_{j,k}$ and $C_{j,k}$. Changes in the backward translation model will be reflected in the computation of translation probabilities for the forward translation model, and vice versa. Thus, through the regularization term, the two translation models are able to exchange information to enhance the estimation of their translation probabilities.

Choice of probabilities $p(w)$ *and* $p(b)$. One natural choice for $p(w)$ and $p(b)$ is to use the empirical values that are estimated from the training corpus. However, one problem is that, the empirical distribution for term frequency follows a skewed distribution (Zipf's law). As a result, if the empirical $p(w)$ is used directly, the regularization term will mainly focus on the consistency checking for common words. For rare words, since its empirical $p(w)$ is very small, its impact in the regularization term is almost ignorable. To put equal emphasis on both common words and uncommon words, we decide to use a uniform distribution for both $p(w)$ and $p(b)$, which turns out to have better performance in our empirical studies.

Choice of λ. Weight λ determines the degree of consistency that we want between the two translation models. A very large λ will enforce the two translation models to have almost identical prediction, which can significantly degrade the performance for automatic image annotation. A very small λ will simply ignore the correlation between the two translation models and return back to the individual translation models. To gain the best performance, we divide the training data into the set for training the model and a set for evaluation set to determine the value of λ. After deciding the value for λ, we will retrain the regularization-based symmetric translation model using all training data.

4.2.3 Derivation of the EM Algorithm for Regularization based Symmetric Translation Model

Suppose that the objective function is $l = l_1 + l_2$, and the parameter set is $\theta = \{t_{j,k}, u_{k,j}\}$.

Define $l_1 = \log(l(\theta)) - \log\left(l\left(\theta^{old}\right)\right)$, where

$$log(l(\theta)) = \sum_{i=1}^{|T|} \sum_{\{j|w_{i,j}=1\}} log\left(\sum_{k=1}^{m} t_{j,k} b_{i,k}\right) + \sum_{i=1}^{|T|} \sum_{\{k|b_{i,k}=1\}} log\left(\sum_{j=1}^{n} \mu_{k,j} w_{i,j}\right)$$

Applying Jensen's inequality, the bound for the log likelihood of l_1 can be written as:

$$\begin{aligned} l_1 \geq & \sum_{i=1}^{|T|} \sum_{\{j|w_{i,j}=1\}} \sum_{k=1}^{m} \frac{t^{old}_{j,k} b_{i,k}}{\sum_{k'=1}^{m} t^{old}_{j,k'} b_{i,k'}} log\left(t_{j,k} b_{i,k}\right) \\ & + \sum_{i=1}^{|T|} \sum_{\{k|b_{i,k}=1\}} \sum_{j=1}^{n} \frac{\mu^{old}_{k,j} w_{i,j}}{\sum_{j'=1}^{n} \mu^{old}_{k,j'} w_{i,j'}} log\left(\mu_{k,j} w_{i,j}\right) \end{aligned}$$

Define l_2 as the penalization term of KL-divergence.

$$\begin{aligned} l_2 & = -\lambda \left(\sum_j \sum_k t_{j,k} p(b_k) \log \left(\frac{t_{j,k} p(b_k)}{t_{k,j} p(w_j)} \right) + \sum_j \sum_k \mu_{k,j} p(w_j) \log \left(\frac{\mu_{k,j} p(w_j)}{t_{j,k} p(b_k)} \right) \right) \\ & = -\lambda \sum_j \sum_k t_{j,k} p(b_k) \left(\log \left(\frac{t_{j,k}}{t^{old}_{j,k}} \right) + \log \left(t^{old}_{j,k} \right) + \log \left(p\left(b_k\right) \right) - \log \left(\mu_{k,j} p\left(w_j\right) \right) \right) \\ & -\lambda \sum_j \sum_k \mu_{k,j} p\left(w_j\right) \left(\log \left(\frac{\mu_{k,j}}{\mu^{old}_{k,j}} \right) + \log \left(\mu^{old}_{k,j} \right) + \log \left(p\left(w_j\right) \right) - \log \left(t_{j,k} p\left(b_k\right) \right) \right) \end{aligned}$$

Suppose

$$\begin{aligned} \eta_{j,k} &= \log \left(t^{old}_{j,k} \right) + \log \left(p\left(b_k\right) \right) - \log \left(\mu_{k,j} p\left(w_j\right) \right) \\ \delta_{k,j} &= \log \left(\mu^{old}_{k,j} \right) + \log \left(p\left(w_j\right) \right) - \log \left(t_{j,k} p\left(b_k\right) \right) \end{aligned}$$

With inequality $\log(x) \leq x - 1$, the lower bound of l_2 could be written as

$$\begin{aligned} l_2 & \geq -\lambda \sum_j \sum_k t_{j,k} p(b_k) \left(\log \left(\frac{t_{j,k}}{t^{old}_{j,k}} - 1 \right) + \eta_{j,k} \right) \\ & -\lambda \sum_j \sum_k \mu_{k,j} p\left(w_j\right) \left(\log \left(\frac{\mu_{k,j}}{\mu^{old}_{k,j}} - 1 \right) + \delta_{k,j} \right) \end{aligned}$$

So the objective function we will optimize is l with constraints $\sum_j t_{j,k} = 1$ for each k and $\sum_k \mu_{k,j} =$ for each j.

After introducing the Lagrangian term, the function we will optimize is

$$\Phi = \sum_{i=1}^{|T|} \sum_{\{j|w_{i,j}=1\}} \sum_{k=1}^{m} \frac{t^{old}_{j,k} b_{i,k}}{\sum_{k'=1}^{m} t^{old}_{j,k'} b_{i,k'}} log\left(t_{j,k} b_{i,k}\right)$$

$$+\sum_{i=1}^{|T|} \sum_{\{k|b_{i,k}=1\}} \sum_{j=1}^{n} \frac{\mu_{k,j}^{old} w_{i,j}}{\sum_{j'=1}^{n} \mu_{k,j'}^{old} w_{i,j'}} log\left(\mu_{k,j} w_{i,j}\right)$$

$$-\lambda \sum_{j} \sum_{k} t_{j,k} p(b_k) \left(\log \left(\frac{t_{j,k}}{t_{j,k}^{old}} - 1 \right) + \eta_{j,k} \right)$$

$$-\lambda \sum_{j} \sum_{k} \mu_{k,j} p\left(w_j\right) \left(\log \left(\frac{\mu_{k,j}}{\mu_{k,j}^{old}} - 1 \right) + \delta_{k,j} \right)$$

$$+\lambda_k \left(1 - \sum_{j} t_{j,k} \right) + \beta_j \left(1 - \sum_{k} \mu_{k,j} \right)$$

Taking the derivative with respect to $t_{j,k}$ and setting it to 0, we can get the solution for $t_{j,k}$

$$\begin{aligned}
t_{j,k}^{new} &= \frac{2C_{j,k}}{B_{j,k} + \sqrt{B_{j,k}^2 + 4A_{j,k}C_{j,k}}} \\
A_{j,k} &= 2\lambda \frac{p(b_k)}{t_{j,k}^{old}} \\
B_{j,k} &= \lambda p(b_k) \log(t_{j,k}^{old}) - \lambda p(b_k) \log(u_{k,j} p(w_j)) + \lambda_k \\
C_{j,k} &= \sum_{i=1}^{|T|} \frac{t_{j,k}^{old} b_{i,k}}{\sum_{k'=1}^{m} t_{j,k'}^{old} b_{i,k'}} + \lambda p(w_j) u_{k,j}
\end{aligned}$$

With constraint $\sum_j t_{j,k} = 1$, we could treat λ_k as a variable and search the solution for this equation. After obtaining the value of λ_k, we could in turn get the solution for $t_{j,k}$. Similarly, we could get the updating equation for $\mu_{k,j}$.

4.3 Experiments

In this section, we will investigate the two enhanced methods on the COREL data set used in [41]. The data set consists of 5000 annotated images, among which 4500 images are used for training and selection of parameters and the rest 500 images are used for testing. 374 different words are used for annotating both the training and testing images. The maximum number of annotation words for one image is 5 and the average number of annotation words for each image is 3.5. Similar to the previous studies on automatic image annotation, the quality of automatic image annotation is measured by the performance of retrieving auto-annotated images regarding to single-word queries. For each single-word query, *precision* and *recall* are computed using the retrieved lists that are based on the true annotations and the auto-annotations. Let I_j be a test image, t_j be its true annotation, and g_j be its auto-annotation. For a given query word w, precision and recall are defined respectively as:

$$\begin{aligned} precision(w) &= \frac{|\{I_j|w \in t_j \wedge w \in g_j\}|}{|\{I_j|w \in g_j\}|} \\ recall(w) &= \frac{|\{I_j|w \in t_j \wedge w \in g_j\}|}{|\{I_j|w \in t_j\}|} \end{aligned}$$

The $precision(w)$ measures the accuracy in annotating images with word w and the $recall(w)$ measures the completeness in annotating images with word w. After we obtain the precision and recall values for each word, the F measure of the word is defined as:

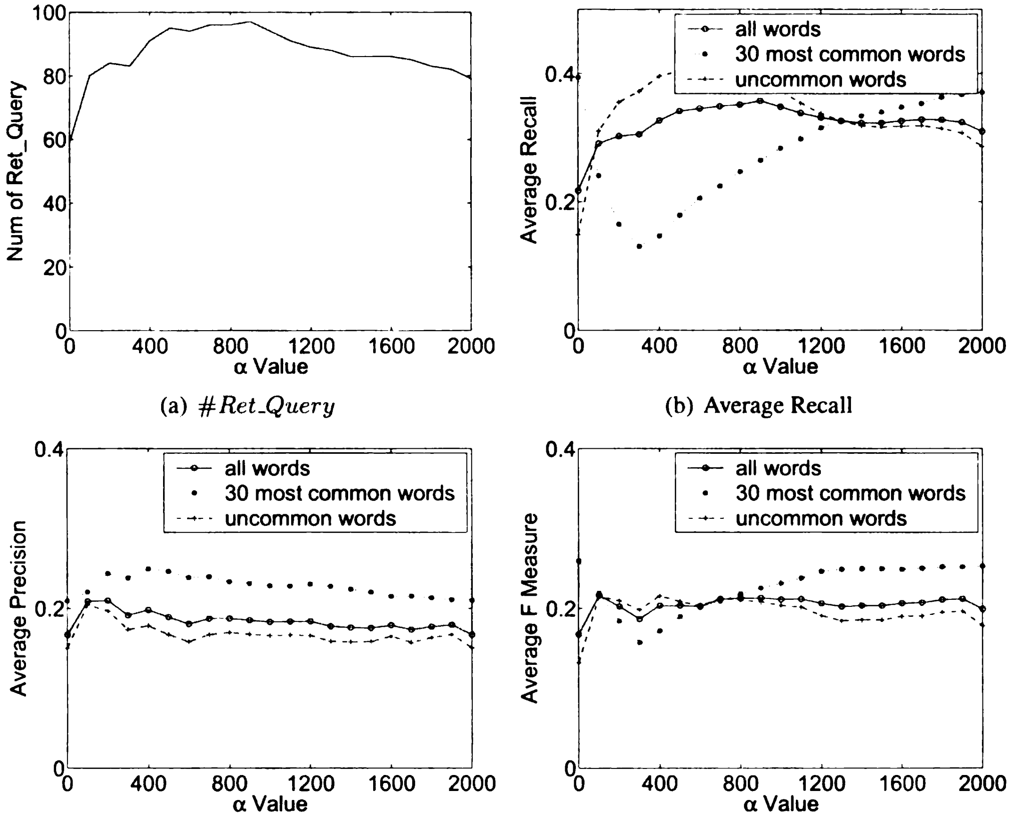
$$F(w) = \frac{2precision(w)recall(w)}{precision(w) + recall(w)}$$

The average of precision, recall, and F measure over different single-word queries are used to measure the overall quality of automatically generated annotations.

The forth metric, *#Ret_Query*, is the number of single-word queries for which at least one relevant image can be retrieved:

$$\#Ret\_Query = |\{w|precision(w) > 0 \wedge recall(w) > 0\}|$$

This metric compensates the metrics of average precision, average recall, and average F measure by providing information about how wide is the range of words that contribute



(a) *#Ret_Query*

(b) Average Recall

(c) Average Precision

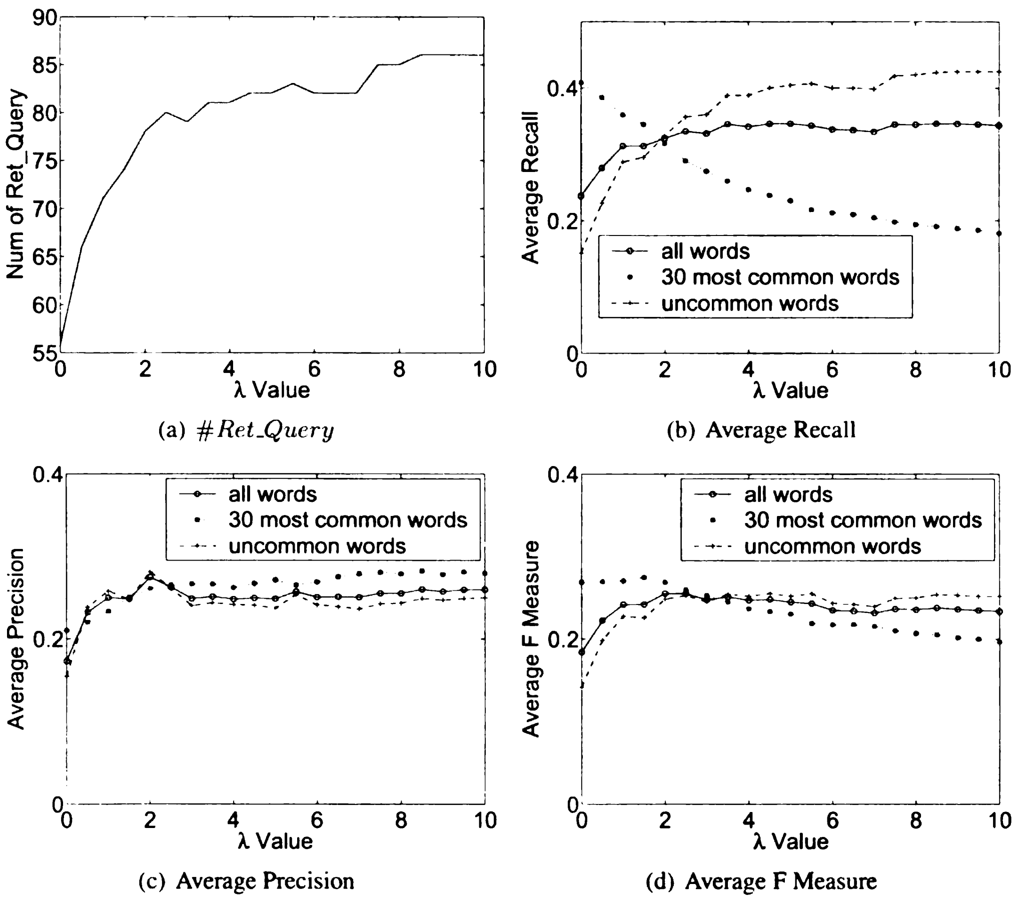
(d) Average F Measure

Figure 4.4. Performance measurement for different α values of Regularized Translation Model

to the average precision ,recall, and F measure. In this section, we divide the training set into two parts: 4000 images are used to train the model and 500 images are used as the evaluation set to determine the parameters of the model. When choosing the parameters for each of the models, we use 5 words as the size of the annotation set and evaluate the performance of the two individual models under different values of the parameters. After the value for the parameter is chosen, we use the whole training set to train each of the models again and the resulting performance is compared with other models in Chapter 5.

4.3.1 Impact of Prior on the Regularized Translation Model

As already pointed out, the constant α has a large impact on the performance of regularized translation model for automatic image annotation. In this experiment, we measure the change with respect to the four metrics for different values of α. Figure 4.4 plots the curve for $\#Ret\_Query$, average precision, average recall, and average F measurement when α is varied from 0 to 2000. We observe that the performances first improve and then the improvements become saturate when α is larger than 400. This is because, when α is a small value, the distribution of the number of blobs associated with words is rather skewed and thus increasing the value for α will have a great impact on balancing the distribution, which leads to significant improvement with respect to the four metrics. However, when α becomes large, the number of blobs associated with different words tends to be evenly distributed over different words. As a result, increasing the value of α will make little adjustment to the distribution of blob numbers and therefore little change will be made to the translation model. From Figure 4.4, both average precision and recall are increased substantially when α is increased from 0 to 200. This is contradictory to many studies in information retrieval, in which improvement in recall usually leads to degradation in precision. To have a better understanding of this phenomenon, we divide the words into two groups: a group of common words and a group of uncommon words. In Figure 4.4, in addition to the average precision ,recall and F measure, we also plot the curve for the average precision, average recall, and average F measure for both common words and uncommon words using the dotted lines and dashed lines, respectively. According to Figure 4.4, for both common words and uncommon words, the change in precision and recall follows the normal patterns, namely that an increase in recall is usually accompanied with decrease in precision. Furthermore, the trends in the change of precision and recall for these two groups of words are almost opposite to each other: increase in the recall of uncommon words is usually accompanied with a decrease in the recall of common words. Since the overall average value for precision and recall is the mean of precision and recall for these



(a) #*Ret_Query*

(b) Average Recall

(c) Average Precision

(d) Average F Measure

Figure 4.5. Performance measurement for different λ values for Regularized Symmetric Translation Model

two groups of words, the opposite trends in these two groups somehow compensate each other and lead to increase in both the average precision and recall. One disadvantage of using large values for α is that a larger α usually results in a slower convergence for the EM algorithm. Apparently, the performance of the regularized model saturates after α is greater than 400, $\alpha = 400$ has the best tradeoff between computational cost and predication accuracy. This value is used in our later experiment.

4.3.2 Effects of Exploring Correlation between Forward and Backward Translation Models

The importance of exploring the correlation between the forward and backward translation model can be illustrated by varying the value for λ. The results for the four metrics are plotted in Figure 4.5. When $\lambda = 0$, no correlation between the two translation models are taken into account. By increasing λ, more and more correlations are introduced between the forward and the backward model. We experiment the set of values that are multiplier of 5000. The multiplier ranges from 1 to 20 is used in our study. As indicated in Figure 4.5(Note: When plotting the figure, the λ values are divided by 10^4 to be fit in the paper), the performance improves at first and then saturates after $\lambda = 50,000$. We also plot the performance of the 30 most common words and the performance of the rest of words. We observe that average recall keeps decreasing for common words and keeps increasing for rare words, which indicates that with the increasing of λ, we obtain better and better performance on rare words. This could be used to determine the value of λ to have a good tradeoff between the performance of rare words and common words. In addition, the average precision first increases and then keeps relatively stable. This result further indicates that introducing more correlation between the forward and backward translation model can significantly enhance the quality of auto-annotations. In our later experiment, we choose $\lambda = 50,000$.

CHAPTER 5

Image Annotation through Label Correlation

Feng Kang, Rong Jin and Rahul Sukthankar. Correlated Label Propagation with Application to Multi-label Learning. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1719-1726, 2006.

CHAPTER 5

Image Annotation through Label Correlation

In Chapter 4, we discussed translation models for the automatic image annotation. Two enhanced translation models are proposed to address the over-estimated common word problem with the traditional translation model. While the proposed enhanced translation models improve the original model substantially, there are two major problems that prevent the performance from being further improved:

- The discrete representation of images loses information and result in high annotation errors. To construct the translation models for the automatic image annotation, we need first apply clustering algorithms to group image regions into clusters to be able to construct the visual vocabulary to represent the images. While this representation reduces the storage space compared with the continuous features, it loses information in the clustering procedure. For example, it might group regions of different flowers into the same cluster. As a result, the identities of different flowers become indistinguishable. However, this difference will be retained if we use continuous features.

- None of the previous models explores the correlation among different annotation words. Effectively exploring the semantic correlation among annotation words is important since the visual features are often insufficient for deciding the appropriate annotation words. For instance, the words 'ocean' and 'sky' are both strongly related

> to the blue color feature. Therefore it may be difficult to distinguish these two words based on the color features alone. However, if we are confident that an image should be annotated with 'grass', then it is more likely that a region of blue in the same image should be annotated as 'sky' rather than 'ocean'.

In this chapter, we first present the framework for multi-label learning that explores the correlation among different class labels and then apply the proposed framework to automatic image annotation.

A natural way to define a set of correlated labels is to explicitly specify which set of labels are correlated among the whole label sets. However, one big concern of this explicit representation is the scalability issue. Even for 20 different labels, the number of possible label combinations reaches $2^{20} = 1048576$. The combinatorial problem is usually computationally expensive and thus may only apply to a small number of labels. One simplification of this method is to only consider the pairwise correlation among any two class labels instead of all possible labels, which is adopted by several researchers[51, 46, 17]. However, due to the large number of labels in our annotation task, even this simplified approach may not be able to work efficiently, since the complexity is on the order of $O(n^2)$, where n is the number of labels.

In the following, we will first present one of the methods based on the explicit specification of label correlation, the Multi-label Maximum Entropy Model[51], which extends the single label Maximum Entropy Model[4] by specifying the pairwise constraints between any two class labels. The Multi-label Maximum Entropy Model[51] was originally applied to multi-label text classification. In this study, we apply the model to automatic image annotation.

We then present a framework of multi-label learning that addresses the correlation among different class labels, called the Correlated Label Propagation(CLP) framework [27]. Based on properties of the submodular function, the framework can obtain the optimal solution using a very efficient algorithm, although there are still an exponential number of constraints. Furthermore, we show how this framework can be applied to automatic image annotation using either discrete feature representation of images or continuous feature representation of images.

5.1 Problem Definition of Multi-label Learning

Let $\mathcal{D} = \{(\mathbf{x}_1, \mathcal{S}_1), \ldots, (\mathbf{x}_n, \mathcal{S}_n)\}$ denote the set of labeled examples, where n is the number of training examples. Each $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,d})$ is an input vector of d dimension. Each set $\mathcal{S}_i$ contains the class labels that are assigned to the i-th training example. For the convenience of presentation, we will employ a binary vector to represent a set of class labels. In particular, for a class label set S_i, its vector representation $t(S_i) = (t_{i,1}, \ldots, t_{i,m})$ has its j-th element set to 1 only when $j \in \mathcal{S}$ and zero otherwise. In total ,we have m different labels. Given a test point $\mathbf{x}_t$, our goal is to determine a confidence vector $\mathbf{z}_t = \{z_{t,1}, \ldots, z_{t,m}\}$ such that each component $z_{t,i}$ indicates the confidence of assigning $\mathbf{x}_t$ to the i-th class.

5.2 Pairwise Label Correlation based on Multi-label Maximum Entropy Model

Let $P(x|t)$, $Q(x|t)$ denote the empirical and the model distributions respectively, where x is data and t is the set of labels assigned to the data point as defined before. For the single label Maximum Entropy Model, the framework to maximize the entropy is formulated as:

$$\hat{q} = \max_q H(x, t|Q) = \min_q < \log q(t|x) >_Q$$

subject to:

$$< t >_Q = < t >_P$$

$$< tx_l >_Q = < tx_l >_P, \forall 1 \leq l \leq d$$

, where $H(x, t|Q)$ is the entropy of data x and labels t. d is the dimension of the data, and tx_l is the l-th feature with respect to each category in t. $< \cdot >_P$ denotes the expectations with respect to the distribution P. The constraints essentially force the expectation under the empirical distribution P consistent with the expectation under model distribution Q. The solution could be proved to take the format:

$$\hat{q}(t|x) = \frac{1}{Z(x)} \exp\left(t(b + w^T x)\right)$$

,where $Z(x) = \sum_t \exp\left(t(b + w^T x)\right)$ is the partition function. w and b are the parameters to optimize, which could be computed through numerical optimization methods.

To extend to the multi-label case, the correlations between pair-wise labels are added to the framework:

$$< t_i t_j >_Q = < t_i t_j >_P, \forall 1 \leq i \leq j \leq m$$

The solution of the Multilabel Maximum Entropy Model is:

$$\hat{q}(t|x) = \frac{1}{Z(x)} \exp\left(t^T(b + Rt + Wx)\right)$$

,where $Z(x) = \sum_t \exp\left(t(b + w^T x)\right)$ is the partition function. Note that t is a binary vector of labels to indicate whether label j appears in the annotation set. $Z(x)$ is to summarize all possible combinations of the labels. To predict the assignment of possible labels to a new test instance, the method enumerates all possible label sets to find the most probable one based on:

$$\hat{t} = \max_t t^T \left(\hat{b} + \hat{R}t + \hat{W}x\right) \tag{5.1}$$

From the above formula for Multi-label Maximum Entropy Model, we observe that the number of parameters to be estimated is at the order of $O(m * d + m^2)$, where m is the number of labels and d is the dimension of data. Due to the large number of parameters to be estimated, it's difficult to find the optimal solution when the number of labels is large since we need to compute the normalization factor $Z(x)$, which requires to compute

$O(2^m)$ conditional probabilities to get the gradient. As a matter of fact, when applying this method to the text classification task [51], the author only considers 10 possible class labels. Considering the size of labels in our annotation task, this method is not feasible to our image annotation task.

5.3 Correlated Label Propagation

In the following, we first describe the proposed framework for multi-label learning, followed by the efficient greedy algorithm using the concept of submodular functions and discussion of implementation issues in the framework.

5.3.1 Correlated Label Propagation for Multi-label Learning

To motivate our proposed framework, we first describe the kernel-based kNN approach, which is one of very popular learning methods.

Suppose the similarity of any two data points is measured by a kernel function $K(\cdot,\cdot)$: $\mathbf{R}^d \times \mathbf{R}^d \rightarrow \mathbf{R}$. Consider the case of single-step propagation. The score of assigning the j-th class to the test example $\mathbf{x}_t$, i.e., $z_{t,j}$, could be estimated by

$$z_{t,j} = \sum_{i=1}^{n} K(\mathbf{x}_t, \mathbf{x}_i) I(j \in \mathcal{S}_i), \tag{5.2}$$

where $I(j \in \mathcal{S})$ is an indicator function that outputs 1 when the j-th class belongs to set $\mathcal{S}$ and is zero otherwise. However, there are two problems with the expression in Equation 5.2:

- *Overestimated Confidence Score.* Equation 5.2 assumes that a training example $\mathbf{x}_i$ will propagate *all* of its class labels to the test example $\mathbf{x}_t$ according to the similarity $K(\mathbf{x}_t, \mathbf{x}_i)$. This is not necessarily true since maybe only *some* of the class labels of $\mathbf{x}_i$ should be propagated to $\mathbf{x}_t$ even though $\mathbf{x}_i$ is similar to $\mathbf{x}_t$.
- *Independent Label Propagation.* As indicated in Equation 5.2, each class label is propagated from training examples to the test example independently of the other

class labels. In particular, the computation of the confidence score $z_{t,j}$ for the j-th label is independent from the confidence scores assigned to other class labels.

To resolve the problem of overestimated confidence scores, we replace the equality constraint in Equation 5.2 with the following inequality constraint:

$$z_{t,j} \leq \sum_{i=1}^{n} K(\mathbf{x}_t, \mathbf{x}_i) I(j \in \mathcal{S}_i). \tag{5.3}$$

The above inequality indicates that the confidence score propagated from training examples to the test example is upper bounded by the sum of the pairwise similarity $K(\cdot, \cdot)$. Note that no explicit value of the confidence score $z_{t,j}$ is specified in the above expression.

To incorporate the label correlation information into label propagation, we consider the propagation of multiple labels. Let the binary vector $\mathcal{S}$ be the set of labels that are propagated from the training examples $\mathcal{D}$ to the test example $\mathbf{x}_t$. We denote by $s_t(\mathcal{S})$ the confidence score of assigning *any subset* of $\mathcal{S}$ to $\mathbf{x}_t$. Similar to Equation 5.3, we introduce the following constraint on the confidence score $s_t(\mathcal{S})$, i.e.,

$$s_t(\mathcal{S}) \leq \sum_{i=1}^{n} K(\mathbf{x}_t, \mathbf{x}_i) I(\mathcal{S} \cap \mathcal{S}_i \neq \phi) \tag{5.4}$$

where $I(\mathcal{S} \cap \mathcal{S}_i \neq \phi)$ is used to ensure that only the training examples whose class labels overlap with the set $\mathcal{S}$ are included in computing the confidence score. To link $z_{t,j}$, i.e., the confidence score of assigning individual classes, to $s_t(\mathcal{S})$, i.e., the confidence score of assigning multiple classes, we assume the following inequality,

$$\sum_{j=1}^{m} z_{t,j} I(j \in \mathcal{S}) \leq s_t(\mathcal{S}). \tag{5.5}$$

The above inequality implies that for a single data point, the confidence of assigning any subset of class label set $\mathcal{S}$ to $\mathbf{x}_t$ should be no less than the confidence of assigning the class label separately in $\mathcal{S}$ to $\mathbf{x}_t$. Combining Equation 5.5 with Equation 5.4, we obtain

$$\sum_{j=1}^{m} z_{t,j} I(j \in \mathcal{S}) \leq \sum_{i=1}^{n} K(\mathbf{x}_t, \mathbf{x}_i) I(\mathcal{S} \cap \mathcal{S}_i \neq \phi).$$

The above expression can be simplified if we present it in the vector form of the class labels, i.e.,

$$\mathbf{z}_t^T \mathbf{t}(\mathcal{S}) \leq \sum_{i=1}^{n} K(\mathbf{x}_t, \mathbf{x}_i) I(\mathbf{t}(\mathcal{S})^T \mathbf{t}(\mathcal{S}_i)) \tag{5.6}$$

Hence, given m different class labels and multi-labeled training examples $\mathcal{D}$, the confidence $\mathbf{z}$ of assigning individual classes to the test example $\mathbf{x}_t$ is subject to the following constraints:

$$\begin{aligned} &\forall \mathbf{t} \in \{0,1\}^m,\ \mathbf{z}_t^T \mathbf{t} \leq \sum_{i=1}^{n} K(\mathbf{x}_t, \mathbf{x}_i) I(\mathbf{t}^T \mathbf{t}(\mathcal{S}_i)) \\ &\mathbf{z} \succeq 0. \end{aligned} \tag{5.7}$$

Furthermore, we can generalize the indicator function $I(x)$ to a *concave* function $\Omega(x)$, which we term the *Label Kernel Function*. Then, the constraints in Equation 5.7 are generalized to the following form:

$$\begin{aligned} &\forall \mathbf{t} \in \{0,1\}^m,\ \mathbf{z}_t^T \mathbf{t} \leq \sum_{i=1}^{n} K(\mathbf{x}_t, \mathbf{x}_i) \Omega(\mathbf{t}^T \mathbf{t}(\mathcal{S}_i)) \\ &\mathbf{z} \succeq 0 \end{aligned} \tag{5.8}$$

A detailed discussion of the label kernel function $\Omega(x)$ appears later.

It is insufficient to identify the appropriate confidence scores $\mathbf{z}$ only with the constraints. Thus, we assume that among all the confidence scores that satisfy the constraints in Equation 5.8, the optimal solution $\mathbf{z}$ is the one that 'maximally' satisfies the constraints based on certain weights. This assumption leads to the following optimization problem for $\mathbf{z}$:

$$\begin{aligned} \max_{\mathbf{z} \in \mathbf{R}^m} \quad & \sum_{k=1}^{m} \alpha_k z_k \\ \text{s.t.} \quad & \forall \mathbf{t} \in \{0,1\}^m : \mathbf{z}^T \mathbf{t} \leq \sum_{i=1}^{n} K(\mathbf{x}_i, \mathbf{x}_q) \Omega(\mathbf{t}^T \mathbf{t}(\mathbf{S}_i)) \\ & \mathbf{z} \succeq 0 \end{aligned} \tag{5.9}$$

where $\{\alpha_k > 0\}_{k=1}^m$ are the weights for the class labels. Notice that the problem in Equation5.9 is a linear programming problem, and therefore the solution will be on the extreme points of the region bounded by the constraints.

```
Input
    • $\mathbf{x}_t$: the test example
    • $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_m > 0$
Output: optimal label scores $(z_{t,1}, \ldots, z_{t,m})$ for $\mathbf{x}_t$
 For $k = 1, \ldots, m$
    • Let class label set $\mathcal{T}_k = \{1, 2, \ldots, k\}$.
    • $f(\mathcal{T}_k) = \sum_{i=1}^{n} K(\mathbf{x}_i, \mathbf{x}_t)\Omega(\mathbf{t}^T(\mathcal{T}_k)\mathbf{t}(\mathcal{S}_i))$
    • $z_{t,k} = f(\mathcal{T}_k) - f(\mathcal{T}_{k-1})$
```

Figure 5.1. Algorithm for finding the optimal solution to Equation 5.9

Despite the simplicity, solving Equation 5.9 efficiently is not trivial. This is because:

- *Efficiency*: The number of constraints in Equation 5.9 is exponential in the number of classes m. When m is large (e.g., 100), the number of constraints will be too large to be handled by any linear programming algorithm.

- *Undetermined Weights*: The solution to Equation 5.9 depends on the weights $\{\alpha_k\}_{k=1}^{m}$, whose exact values are difficult to determine.

5.3.2 Efficient Learning Algorithm

In this section, we show that when the label kernel function $\Omega(x)$ is a concave function, there is a simple greedy algorithm for finding the optimal solution to the problem in Equation 5.9. Furthermore, the solution only depends on the relative *order* of weights $\{\alpha_k\}_{k=1}^{m}$, and is independent of their exact values. The algorithm for estimating label confidence scores $\mathbf{z}$ is summarized in Figure 5.1. This greedy algorithm is based on the following theorem from discrete optimization [40]:

Given: (1) a finite set $\mathcal{N}$, (2) a set function $f : 2^{\mathcal{N}} \rightarrow \mathbf{R}$ with $f(\phi) \geq 0$, and (3) a weight vector $\mathbf{w} \in \mathbf{R}^{|\mathcal{N}|}$. Then, the linear programming problem:

$$\begin{array}{ll}\max\limits_{\mathbf{w}\in\mathbf{R}^{|\mathcal{N}|}} & \mathbf{w}^T\mathbf{x} \\ \text{s. t.} & \forall \mathcal{A} \subseteq \mathcal{N},\ \sum\limits_{e\in\mathcal{A}} x(e) \leq f(\mathcal{A}) \\ & \forall e \in \mathcal{N}, x(e) \geq 0\end{array}$$

can be solved by the following greedy algorithm if the set function f is submodular:

- Sort elements of $\mathcal{N}$ as $w(e_1) \geq w(e_2) \geq ... \geq w(e_n)$
- Let $\mathcal{V}_0 = \phi$

  For i =1,. . . ,n, let

  $\mathcal{V}_i = \mathcal{V}_{i-1} + e_i$, and $x(e_i) = f(\mathcal{V}_i) - f(\mathcal{V}_{i-1})$.

The validity of applying the above theorem to our problem defined in Equation 5.9 relies on the fact that the function f in our algorithm, i.e.,

$$f(\mathbf{u}) = \sum_{i=1}^{n} K(\mathbf{x}_t, \mathbf{x}_i)\Omega(\mathbf{u}^T\mathbf{t}_i) \tag{5.10}$$

is a submodular function. We present a proof that f is submodular if $\Omega(x)$ is a concave function in Section 5.4.

Remark: it is interesting that the kernel-based k Nearest-Neighbor is a special case of the algorithm in Figure 5.1, given by setting $\Omega(x) = x$.[1] This is because

$$\begin{aligned} z_{t,k} &= f(\mathcal{T}_k) - f(\mathcal{T}_{k-1}) \\ &= \sum_{i=1}^{n} K(\mathbf{x}_t, \mathbf{x}_i)\left(\mathbf{t}(\mathcal{T}_k) - \mathbf{t}(\mathcal{T}_{k-1}))^T\mathbf{t}(\mathcal{S}_i))\right) \\ &= \sum_{i=1}^{n} K(\mathbf{x}_t, \mathbf{x}_i)\left(\mathbf{e}_k^T\mathbf{t}(\mathcal{S}_i))\right) \\ &= \sum_{i=1}^{n} K(\mathbf{x}_t, \mathbf{x}_i)I(k \in \mathcal{S}_i) \end{aligned}$$

[1] The linear function x is both a concave and a convex function.

where e_k is the vector whose elements are all zero except that the k-th element is 1. In the last step of the above derivation, we use the property:

$$\mathrm{e}_k^T \mathrm{t}(\mathcal{S}_i) = \begin{cases} 1 & k \in \mathcal{S}_i \\ 0 & \text{otherwise.} \end{cases}$$

5.3.3 Selection of the Label Weights and the Label Kernel Function

This framework contains three key factors: weights $\{\alpha_k\}_{k=1}^m$, the label kernel function $\Omega(x)$, and the similarity function $K(x_i, x_k)$. This section discusses the impact of different choices for weights $\{\alpha_k\}_{k=1}^m$ and the label kernel function $\Omega(x)$.

Choice of Weights: The solution returned by the label propagation algorithm is dependent only on the relative *order* of the weights, $\{\alpha_k\}_{k=1}^m$ but not the exact values. There are two straightforward choices for the weights:

1. order the weights α to be in the same order as class frequency, namely $\alpha_i \geq \alpha_j \longleftrightarrow p_i \geq p_j$;
2. order the weights to be in the reverse order of class frequency, namely $\alpha_i \geq \alpha_j \longleftrightarrow p_i \leq p_j$.

Above, p_i is the frequency of the i-th class in the training data. A potential problem with the first choice for α is that assigning large weights to the popular classes will mean that those classes will be selected before the rare classes. Since popular classes are correlated with many more classes than rare classes, choosing the weights for the popular classes first could cause the test sample to overlap with the labeled samples heavily from the beginning and thus the label kernel function saturates from the beginning due to the property of concave function. This is best illustrated using an example. Consider the two classes 'water' and 'whale', where the former is popular and the latter rare; every time 'whale' appears in the label set of an training data, 'water' also appears but not vice versa. If 'water' were chosen before 'whale', then no additional overlap information would be introduced when the weight for 'whale' was determined. That is, according to the computation formula, since the set $\{water\}$ and the set $\{water, whale\}$ have the same overlapping with the label set of each of the training example and thus the Ω function has the same value and thus the confidence for the word 'whale' will be **zero** if we already predicted the word 'water'. By contrast, the second choice (selecting weights in reverse order of class frequency) allows the rare classes to determine their confidence scores before the popular classes. In our example, this means that new overlapping information is introduced when the weight for 'water' is selected after the weight for 'whale'. For these reasons, our experiments employ the second choice for the weights.

Choice of the label kernel function: As discussed above, a prerequisite for the label kernel function $\Omega(x)$ is that it should be concave. In addition to ensuring that $f(\mathbf{u})$ in Equation 5.10 is a submodular function, the choice of a concave function is also consistent with the principle of *Decreasing Marginal Returns* in Economics. Namely, that more information is gained from the first few observations than from the repeated observation of the same evidence.

Table 5.1 lists four examples of label kernel functions: the δ function, the sigmoid function, the exponential function, and the count function. We also plot the curve for four different label kernel functions in Figure5.2 (The count function is multiplied with a small constant to stretch it in order to draw curves on the same graph).

As indicated by the expressions in Table 5.1 and Figure 5.2, these four functions behave very differently. The δ function outputs 1 whenever the input is positive. Thus, no matter how many class labels are shared between those of a training example and those of propagation, the amount of label confidence propagated from the training example remains the same. The exponential function is a monotonically-increasing function with a maximum value of 1 (for $x \geq 0$). Unlike both the δ function and the exponential function, which output zero when the input is zero, the sigmoid function has a non-zero output even when the input is zero. This allows for any training example to propagate confidence score to the test example even when its assigned classes do not overlap with the class labels of propagation. This property plays a role analogous to smoothing in information retrieval

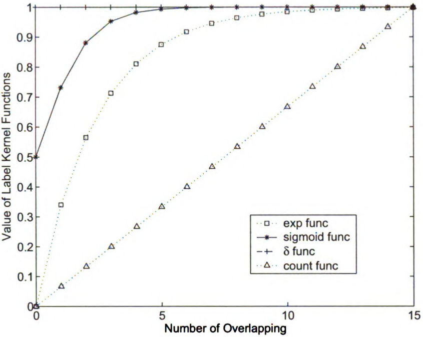


Figure 5.2. The curves for different label kernel functions

Table 5.1. Examples of label kernel functions used in experiments

δ function	$\Omega(x) = \delta(x) = \left\{ \begin{array}{l} 0 \text{ if } x = 0 \\ 1 \text{ if } x > 0 \end{array} \right.$
sigmoid function	$\Omega(x) = \frac{1}{1+e^{-x}}$
exponential function	$\Omega(x) = 1 - 2^{-\alpha x}$
count function	$\Omega(x) = x$

(e.g., [48]). Finally, as discussed in the previous section, the count function leads to the standard kernel-based kNN algorithm.

5.3.4 Choice of the Similarity Function K for the Correlated Label Propagation Framework

One of the key components in the CLP framework is the similarity function $K(;)$, which could be very domain specific. When applying to image annotation task, the similarity could be computed based on the discrete features or the continuous features extracted from the images. In the following, we will consider two methods to compute this measurement.

To compute the similarity based on discrete features, the kernel function $K(\mathbf{x}, \mathbf{x}')$ is computed based on the relevance language model [31], which has been successfully applied to automatic image annotation [24]. More specifically, the similarity of a test example $\mathbf{x}_t$ to a training example x_i, denoted by $K(\mathbf{x}_t, \mathbf{x}_i)$, is calculated as:

$$K(\mathbf{x}_t, \mathbf{x}_i) = \Pr(\mathbf{x}_t|\mathbf{x}_i) = \prod_k [p(k|\mathbf{x}_i)]^{x_{t,k}} \tag{5.11}$$

where

$$p(k|\mathbf{x}_i) = \beta \frac{x_{i,k}}{\sum_{k'} x_{i,k'}} + (1-\beta) \frac{\sum_{j=1}^{n} x_{j,k}}{\sum_j = 1^n \sum_{k'} x_{i,k'}}.$$

This similarity could also be constructed based on the continuous features. We take the same method as used in the continuous relevance model[32]. The similarity is computed as:

$$K(\mathbf{x}_t, \mathbf{x}_i) = \Pr(\mathbf{x}_t|\mathbf{x}_i) = \prod_{a=1}^{n_{x_t}} \int_{R^k} p_R(r_a|g_a) p_G(g_a|\mathbf{x}_k) dg_a \tag{5.12}$$

where $p_R(r_q|g_a)$ takes the values of a constant. Gaussian distribution $p_G(g_a|\mathbf{x}_k) = \frac{1}{n}\sum_{j=1}^{n} \frac{1}{\sqrt{2^k \pi^k \det(\Sigma)}} \exp\left(g_a - G(r_j)^T \Sigma^{-1} (g_a - G(r_j))\right)$ is used to generate the image features from a model $\mathbf{x}_k$. $G(r_j)$ is the feature vector of every region of image $\mathbf{x}_k$. In this model, it is assumed that the feature vector g in the test image could be generated from every region r_j in the image $\mathbf{x}_k$ following a Gaussian distribution with r_j as the mean and a diagonal matrix as the covariance matrix.

5.4 Proof of Function f is a Submodular Function

Theorem 1 *Function $f(\mathbf{u})$ in Equation 5.10 is a submodular function if the label kernel function $\Omega(x)$ is concave.*

Proof

To show that $f(\mathbf{u})$ is submodular, we use the following necessary and sufficient conditions for submodular functions:

For any set $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{N}$ and element $e \in \mathcal{N} \setminus \mathcal{B}$,

$$f(\mathcal{A} \cup e) - f(\mathcal{A}) \geq f(\mathcal{B} \cup e) - f(\mathcal{B})$$

iff f is a submodular function. Using the binary vector representation of sets, the above condition can be written as:

$$f(\mathbf{t}(\mathcal{A} \cup e)) - f(\mathbf{t}(\mathcal{A})) \geq f(\mathbf{t}(\mathcal{B} \cup e)) - f(\mathbf{t}(\mathcal{B}))$$

holds when $\mathbf{t}(\mathcal{B}) \succeq \mathbf{t}(\mathrm{A})$ and $\mathbf{t}(e)^T\mathbf{t}(\mathcal{B}) = 0$. Using the expression of $f(\mathbf{u})$ in Equation 5.10, we have

$$\begin{aligned} & f(\mathbf{t}(\mathcal{A} \cup e)) - f(\mathbf{t}(\mathcal{A})) = \\ & \quad \sum_{i=1}^{n} K(x_i, x_t) \left(\Omega(\mathbf{t}^T(\mathcal{A} \cup e)\mathbf{t}(\mathcal{S}_i)) - \Omega(\mathbf{t}^T(\mathcal{A})\mathbf{t}(\mathcal{S}_i))\right) \end{aligned} \tag{5.13}$$

and

$$\begin{aligned} & f(\mathbf{t}(\mathcal{B} \cup e)) - f(\mathbf{t}(\mathcal{B})) = \\ & \quad \sum_{i=1}^{n} K(x_i, x_t) \left(\Omega(\mathbf{t}^T(\mathcal{B} \cup e)\mathbf{t}(\mathcal{S}_i)) - \Omega(\mathbf{t}^T(\mathcal{B})\mathbf{t}(\mathcal{S}_i))\right) \end{aligned} \tag{5.14}$$

Since $e \in \mathcal{N} \setminus \mathcal{B}$ and $\mathcal{A} \subseteq \mathcal{B}$, we have

$$\mathbf{t}(\mathcal{A} \cup e) = \mathbf{t}(\mathcal{A}) + \mathbf{t}(e), \ \mathbf{t}(\mathcal{B} \cup e) = \mathbf{t}(\mathcal{B}) + \mathbf{t}(e)$$

Thus,

$$\begin{aligned} (\mathbf{t}(\mathcal{A} \cup e) - \mathbf{t}(\mathcal{A}))^T\mathbf{t}(\mathcal{S}_i) &= (\mathbf{t}(\mathcal{B} \cup e) - \mathbf{t}(\mathcal{B}))^T\mathbf{t}(\mathcal{S}_i) \\ &= \mathbf{t}(e)^T\mathbf{t}(\mathcal{S}_i) \qquad (5.15) \end{aligned}$$

Furthermore, based on the property $\mathcal{A} \subseteq \mathcal{A} \cup e, \mathcal{B} \subseteq \mathcal{B} \cup e$, we have

$$\mathbf{t}(\mathcal{A})^T\mathbf{t}(\mathcal{S}_i) \leq \mathbf{t}(\mathcal{A} \cup e)^T\mathbf{t}(\mathcal{S}_i), \tag{5.16}$$

$$\mathbf{t}(\mathcal{B})^T\mathbf{t}(\mathcal{S}_i) \leq \mathbf{t}(\mathcal{B} \cup e)^T\mathbf{t}(\mathcal{S}_i). \tag{5.17}$$

Now, based on the properties in Equations 5.15 and 5.17, for any concave function $\Omega(x)$, we have

$$\begin{aligned} &\Omega(\mathbf{t}(\mathcal{A})^T\mathbf{t}(\mathcal{S}_i)) + \Omega(\mathbf{t}(\mathcal{B} \cup e)^T\mathbf{t}(\mathcal{S}_i)) \leq \\ &\quad \Omega(\mathbf{t}(\mathcal{B})^T\mathbf{t}(\mathcal{S}_i)) + \Omega(\mathbf{t}(\mathcal{A} \cup e)^T\mathbf{t}(\mathcal{S}_i)) \end{aligned} \tag{5.18}$$

The above inequality holds because of the following property of the concave function, i.e.,

$$\Omega(x) + \Omega(y) \leq \Omega(p) + \Omega(q)$$

if $x \leq p, q \leq y$ and $x + y = p + q$. By letting

$$\begin{aligned} x &= \mathbf{t}(\mathcal{A})^T\mathbf{t}(\mathcal{S}_i), \quad y = \mathbf{t}(\mathcal{B} \cup e)^T\mathbf{t}(\mathcal{S}_i) \\ q &= \mathbf{t}(\mathcal{B})^T\mathbf{t}(\mathcal{S}_i), \quad p = \mathbf{t}(\mathcal{A} \cup e)^T\mathbf{t}(\mathcal{S}_i) \end{aligned}$$

we have Equation 5.18.

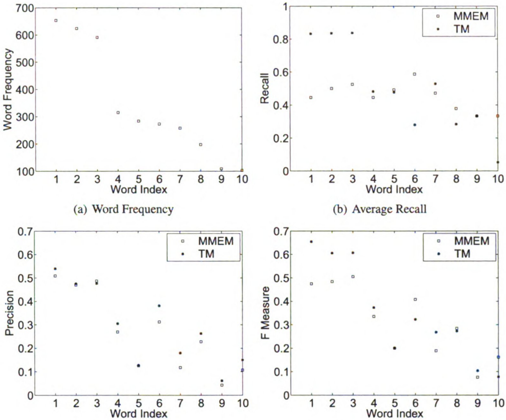
Finally, substituting the inequality in Equation 5.18 into Equations 5.13 and 5.14, we obtain

$$f(\mathbf{t}(\mathcal{A} \cup e)) - f(\mathbf{t}(\mathcal{A})) \geq f(\mathbf{t}(\mathcal{B} \cup e)) - f(\mathbf{t}(\mathcal{B}))$$

5.5 Experiments

In this section, we conduct three sets of experiments. The first set focuses on the study of the pairwise label constraints based on the Multi-label Maximum Entropy Model(MMEM). Due to its incapability to handle a large set of labels, we only evaluate the algorithm by a small subset of class labels. The second set of experiments is based on the Correlated Label Propagation(CLP) framework, which will be evaluated by a large number of class labels. Finally, we conduct the set of experiments to compare all the methods on automatic image annotation task.

Chapter 4's experiments use 5 annotation words to choose parameters. After choosing the parameters, we will train the model based on the whole set using these parameters. In



(a) Word Frequency

(b) Average Recall

(c) Average Precision

(d) Average F Measure

Figure 5.3. Performance for individual words based on the Translation Model(TM) and the Maximum Entropy Model(MEM). The words are ordered by decreasing frequency.

this section's experiments, the performances are evaluated at different annotation cut point, each point of the top k ranked class labels (k ranging from 1 to 10).

5.5.1 Application of the Multi-label Maximum Entropy Model to Automatic Image Annotation

To further investigate how pairwise correlations fit our exploration of the correlation between labels, we apply the Multi-label Maximum Entropy Model to the image annotation task. Due to its incapability to handle large sets of annotation labels, we consider a subset of 10 different class labels, which is also the number of labels used in the paper where the model was developed[51]. To have enough support of training examples for this method,

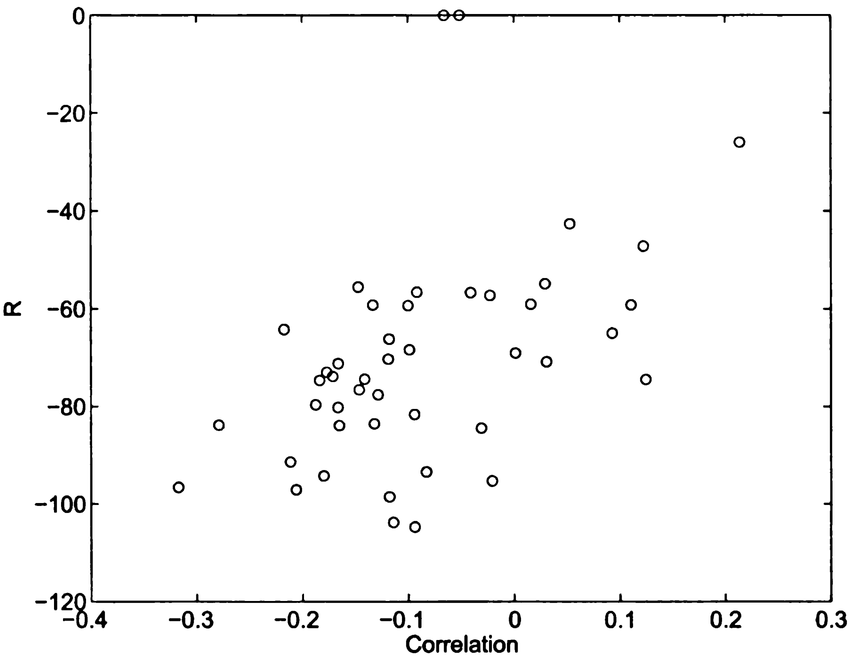


Figure 5.4. The correlation among categories and their corresponding parameters of R in Multilabel Maximum Entropy Model for the Dataset.

we choose the common words in the dataset. 1000 examples are used for training and 500 examples are used for testing. Since there is no ranking information in the results of Multi-label Maximum Entropy Model and the maximum number of predicted words from the Multi-label Maximum Entropy Model is 4, we thus choose 4 annotation words for the translation model to make them comparable. We plot the frequency of each of the annotation words and their corresponding performances in Figure 5.3. From Figure 5.3(a), which shows the frequency of each of the individual words, we observe that although we choose a small set of common words as labels, the distribution is still very skewed. Some words happen more frequently than the rest. We also observe that the Multi-label Maximum Entropy Model performs better than the Translation Model on the relatively less frequent words, especially for the average recall, while the Translation Model performs better than the Multi-label Maximum Entropy Model on the common word set.

To further investigate the role of the correlation in this framework, we investigate the

	common words	uncommon words
common words	-74.3507	-93.7711
uncommon words	-93.7711	-53.6874

Table 5.2. Average correlation between the common words and rare words.

relationship between the Pearson correlation coefficients and the values of R in Equation 5.1 for pairs of labels. The result is listed in Figure 5.4. The Pearson correlation coefficient between label t_i and t_j is defined as,

$$r_{t_j,t_k} = \frac{\sum(t_{ij} - \bar{t_j})(t_{ik} - \bar{t_k})}{(n-1)s_{t_j}s_{t_k}} \tag{5.19}$$

, where $\bar{t_j}$ is the average of label t_j across all the images and s_{t_j} is the sample standard deviation for label t_j.

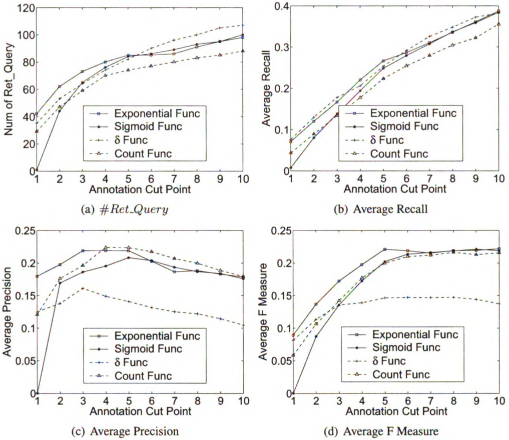
We can observe a general trend from these data. When the correlation gets larger, the coefficient values of R, the values of correlation between two labels used in Equation 5.1, also get larger. This demonstrates that the values of R obtained from the framework reflect the correlation between the labels.

We further investigate the values of R between individual words. We divide the words into two groups: the top 5 frequent words as common words and the remaining 5 words as rare words. We compute the average values of the R matrix among each set of words and between different sets of words. The result is listed in Table 5.2.

From the table, we observe that the rare words have the largest R values among themselves. This indicates that the Multi-label Maximum Entropy Model tends to predict the words as rare words due to the fact that they have a larger R values and thus the model has good performance on rare words but not on the common words.

5.5.2 Application of the Correlated Label Propagation Framework on Discrete Features

In this experiment, we compute the similarity between the test image and the training images based on the discrete representation of images and the formula is in section 5.3.4. The



(a) $\#Ret\_Query$

(b) Average Recall

(c) Average Precision

(d) Average F Measure

Figure 5.5. Performance comparison of correlated label propagation framework based on different label kernel functions.

parameter β is set to be 0.9 as suggested in paper[24]. The value of the exponent in the exponential label kernel function is set to 0.6 through the evaluation set.

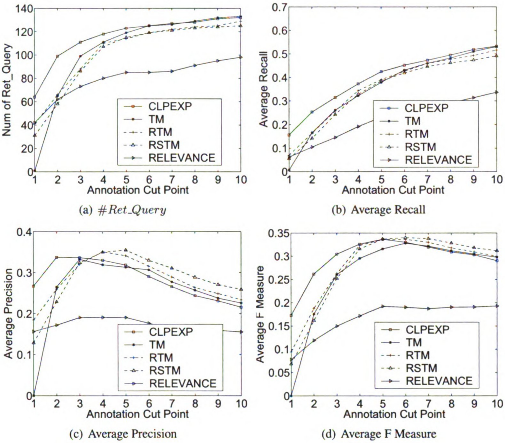
We first compare the performances of different label kernel function $\Omega(x)$. Figure 5.5 shows the performances of the four kernel label functions on the COREL data set. Note that the count function corresponds to the kernel-based kNN approach.

From Figure5.5, we observe that, among the four label kernel functions, the δ function performs the worst for almost all ranks. This is due to the property of the δ function that gives a constant value regardless of the number of class labels overlapping between the assigned class labels of training examples and the propagated class labels. Second, the performance of the three kernel label functions, namely the sigmoid function, the exponential function, and the count function, differ significantly when the cut-off rank is small. Once the cut-off rank is large (e.g., 5 for the COREL dataset), the three label kernel functions deliver similar results. This is because when the number of selected class labels is large, all of the label kernel functions will be able to identify more or less the similar set of class labels. As a result, the F measure of all three label kernel functions are close to each other when the cut-off rank is large. Third, we see that, among the four functions, the exponential function appears to provide the best or close to the best performance.

5.5.3 Application of the Correlated Label Propagation Framework on Continuous Features

In this section, we evaluate the performance of the CLP framework on the set of continuous features. That is, we compute the similarity measurement in the CLP framework based on the Equation 5.12 in section 5.3.4. For the baseline model, we choose the Continuous Relevance Model(CRM). To compare them with the models based on discrete features, we also include the method with the best performance on discrete features, the CLP framework with the exponential function as the label kernel function. The results at different annotation cut points are presented in Figure 5.6. We observe that the CLP framework based on the exponential function achieves best performance and is better than the continuous relevance model, especially when the number of annotation words is less than or equal to 4. This might be related to the average number of annotation words for this dataset, which is 3.5. We can also observe that the performance of the CLP framework based on continuous features are much better than the performance based on discrete features.

This demonstrates that while the more powerful model could improve the performance, the simple modification of the similarity measurement could be another effective way to improve the performance.



(a) $\#Ret\_Query$

(b) Average Recall

(c) Average Precision

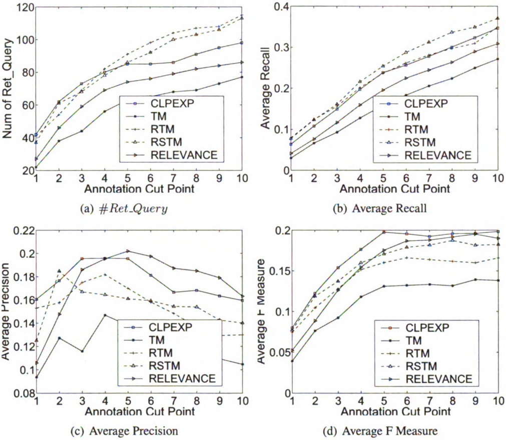
(d) Average F Measure

Figure 5.6. Performance measurement for the CLP framework with continuous features on different annotation cut points

5.5.4 Comparison of All the Proposed Models on Discrete Features

In this section,we compare the performances of the enhanced translation models, the relevance model, and the CLP framework on automatic image annotation task based on discrete features. The result is given in Figure 5.7. Same as before, we use the union of the words for evaluation, which in total is 141 words, and the performance is evaluated at different cut points.

From Figure 5.7, we can see that overall the CLP framework with the exponential function performs best among all the models. When we have 5 annotation words, the average F measure for the method reaches 0.1975, while the next best, regularized symmetric trans-



(a) *#Ret_Query*

(b) Average Recall

(c) Average Precision

(d) Average F Measure

Figure 5.7. Performance comparison of different models. TM: Translation Model. RTM: Regularized Translation Model. RSTM: Regularized Symmetric Translation Model. CLPEXP: the CLP framework with the Exponential Function as the Label Kernel Function, RELEVANCE:Relevance Model

lation model is 0.1706, followed by the regularized translation model at 0.1600 and the original translation model with only about 0.1310.

We further split the annotation words into two groups according to their frequency: 30 most common words in the union set and the remaining 111 words as uncommon words. We further compare the common word performance and the uncommon word performance. The result is listed in Figure 5.8and Figure 5.9 respectively.

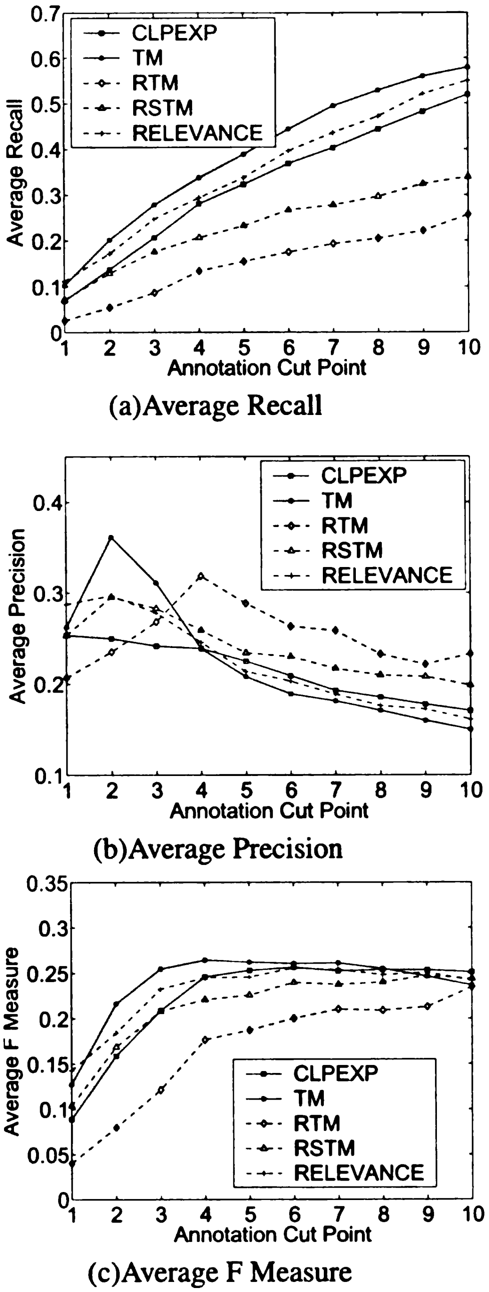
From these figures, we can see that for the common words, the translation model performs best since it predicts most of the annotation words as common words. The second is the CLP framework with the exponential function, while the two modified translation models do not perform well on the common words.

However, for the uncommon words, the trend is reversed. The two modified translation models perform very well and the CLP framework with exponential function performs second best and the translation model performs very bad on the rare words.

While the two modified translation models reduce the likelihood to predict the common words by boosting the performance of the rare words, it over-corrected the performance of the common words. The CLP framework reaches a balance between the performance for common words and rare words and obtains a good overall performance.

5.5.5 A Further Exploration of the Confidence Scores in the CLP Framework

To further investigate the different impact of the label kernel function for the CLP framework. We plot the values of the f function in the computation procedure for the exponential function and the count function. The results are shown in Figure 5.10. The x axis is the order of labels in the computation procedure, starting from the least frequent word. We observe that the f function is a nondecreasing function due to the fact that both the label kernel functions increase as more labels are introduced in the computation procedure. At the beginning, the f values of both functions are close to 0 and the changes of the values are very small. This is because the f function is determined by two parts, both the similarity function K and the label kernel function Ω. At the beginning, because we start with the rare words, there is very little overlap with the annotation words in training data. Most of the values obtained from the label kernel functions are zero and only a small number of training images are considered to be similar to the test image and contribute to the value of f function. The two curves increase more as it comes to more and more popular words, which have more overlapping with the training data. We also plot the change of f values of the neighboring labels, which is the confidence score for the corresponding label. From the figure, we observe that for the CLPEXP model, the change of the f values decreases when the curve extends to the common words. This is because as it becomes close to common words, the overlapping between the label set for the test image and the training images get larger. However, the amount of increase is decreased because the increase of the exponential label kernel function Ω is decreased due to the property of concave function. Although more similar images are introduced into the computation of the f values, the overall increase is decreased. However, the count function still gives big boost to the change of the f values, since every new overlapping introduces the whole similarity to the f values and thus the confidence score.

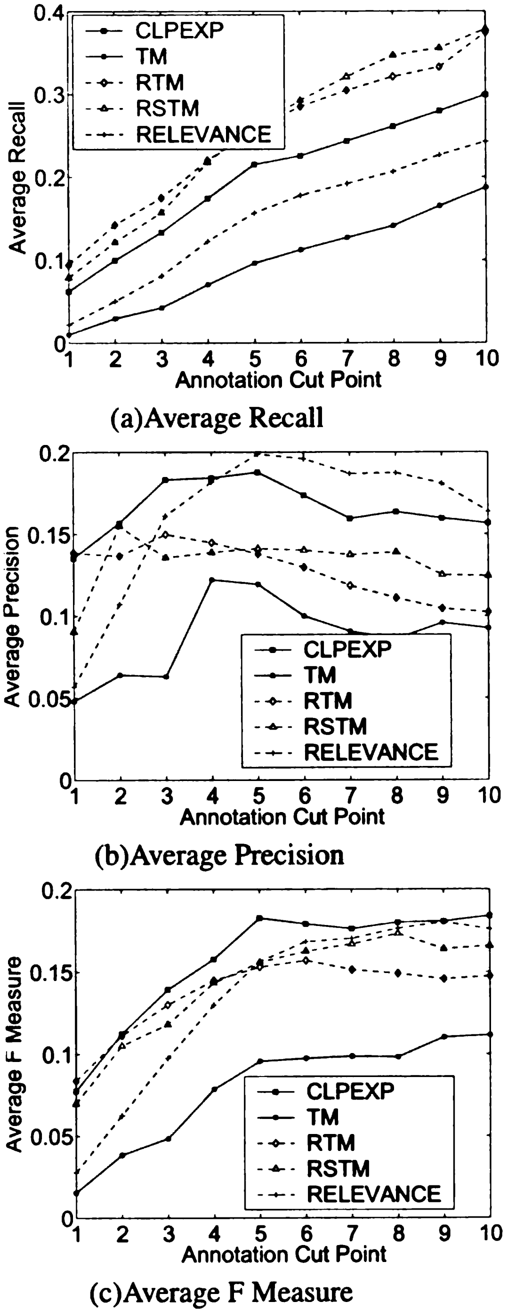


(a)Average Recall

(b)Average Precision

(c)Average F Measure

Figure 5.8. Performance comparison of different models for 30 most common words. TM: Translation Model. RTM: Regularized Translation Model. RSTM: Regularized Symmetric Translation Model. CLPEXP: the CLP framework with exponential function as the label kernel function,RELEVANCE: Relevance Model

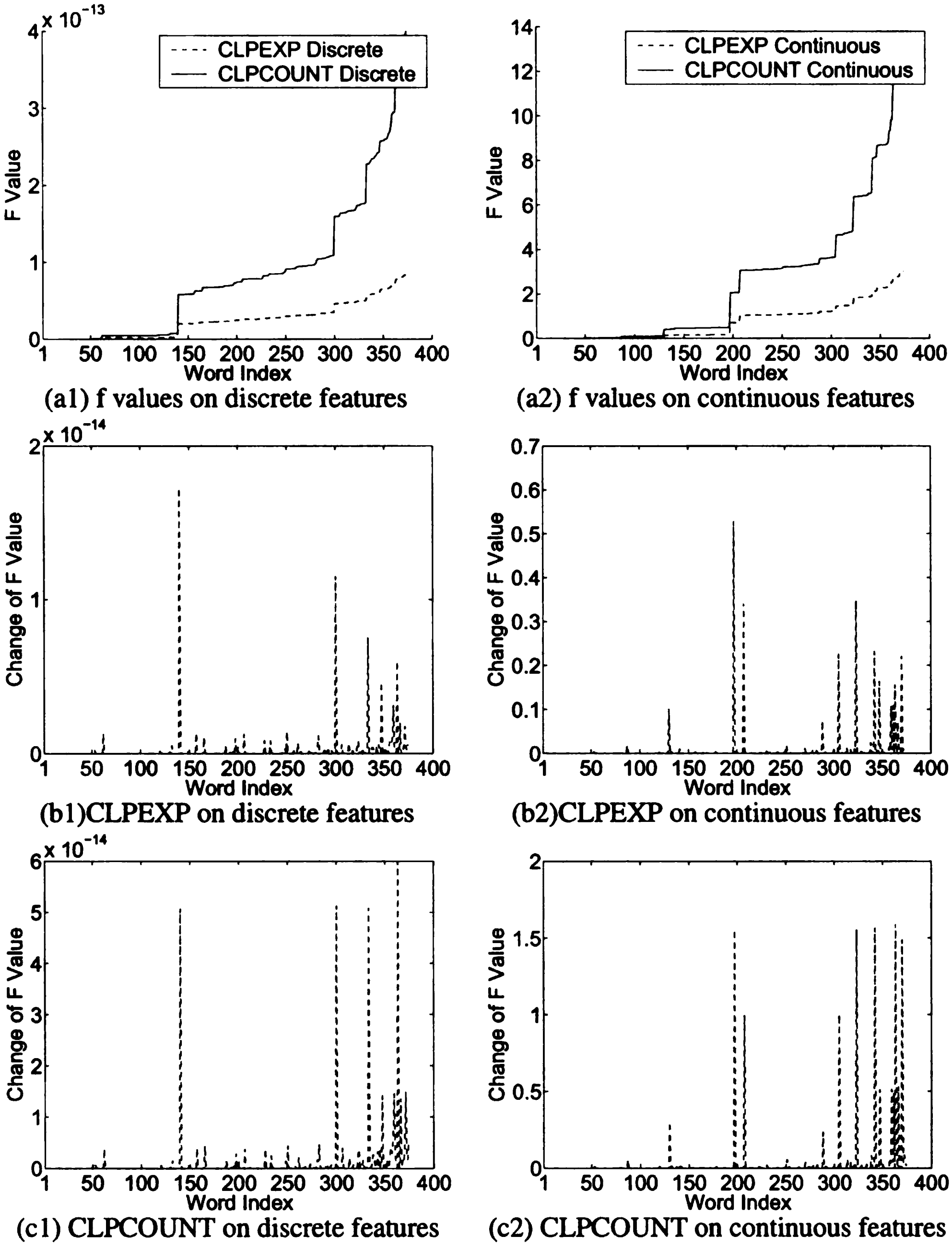


(a)Average Recall

(b)Average Precision

(c)Average F Measure

Figure 5.9. Performance comparison of different models for the remaining rare words. TM: Translation Model. RTM: Regularized Translation Model. RSTM: Regularized Symmetric Translation Model. CLPEXP: the CLP framework with exponential function as the label kernel function. RELEVANCE: Relevance Model



(a1) f values on discrete features

(a2) f values on continuous features

(b1)CLPEXP on discrete features

(b2)CLPEXP on continuous features

(c1) CLPCOUNT on discrete features

(c2) CLPCOUNT on continuous features

Figure 5.10. Values for *f* function and their change

CHAPTER 6

Conclusion and Future Work

Automatic image annotation is to generate a set of textual words to describe the content of images. With annotated textual words, a number of tasks related to images could be constructed, such as image retrieval, image browsing etc. The key in this procedure is to construct a machine learning model to correlate the image features with textual words. Then we can apply this model to annotate new images. This thesis presents several statistical models to conduct automatic image annotation.

6.1 Summary

We first make an extensive study on the statistical translation model. We identified the problems of the statistical translation model, which is the overly estimated common words. Then we propose two methods to enhance the statistical translation model. One is based on limiting the number of blobs associated with each of the word, the regularized translation model. Another one is based on two kinds of translation probabilities: forward translation model, which is the traditional translation model; backward translation model, which is to translate from words to image blobs. By minimizing the difference between these two kinds of translation probabilities, better results are produced.

However, in all the previous translation models, the labels for images are assumed to be independent of each other. This assumption might ignore one of the very important information in image annotation task, the correlation between annotation words. In addition, the translation models use discrete features and thus miss the information encoded in continuous image features.

Based on this observation, we propose to develop models based on the correlations between labels. We first study the Multi-label Maximum Entropy Model, which explores the correlation information through pairwise constraints between labels. However, due to the computation issue, it's infeasible to apply this model to our task. We then present the Correlated Label Propagation (CLP) framework, which explores the correlation between large set of labels. We show that our framework could have an efficient greedy algorithm to obtain the optimal solution and thus apply to large set of labels. Furthermore, the continuous features could be incorporated into our framework to achieve even better performance. One thing worth pointing out is that we prove that the kernel based K Nearest Neighbor approach is a special case of our CLP framework.

6.2 Future Work

- *Better Clustering Results.* From the previous study, we know that the quality of the clustering results is one of the key issues for the models based on the discrete features. The blob information in our experiments is obtained through K-means algorithm. It does not consider the annotation words while performing the clustering procedure. How to incorporate this word information into the clustering procedure could be one of the methods to obtain better clustering results. Another problem is how to determine the appropriate number of clusters, the size of the visual vocabulary. In this study, the number of clusters is predefined as 500. Determine more appropriate number of clusters could be another way to obtain better clustering results.

- *Better Similarity Measurement.* We also observe that the similarity measurement plays a key role in our Correlated Label Propagation framework. After we apply the similarity measurement computed based on continuous features, the performance is improved substantially compared with the similarity measurement computed from discrete features. How to obtain a better similarity measurement is one of the key issues in further improving the performance.

Converting images into textual words is a very challenging task, especially when one image could have multiple annotation words. However, it's also a very interesting topic and holds the promises to constructing better image search engines and many other applications such as image browsing.

BIBLIOGRAPHY

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

[2] K. Barnard, P. Duygulu, R. Guru, P. Gabbur, and D. Forsyth. The effects of segmentation and feature choice in a translation model of object recognition. In *CVPR '03: Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 675–682, 2003.

[3] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. In *ICCV '01: Proceedings of the International Conference on Computer Vision*, pages 408–415, 2001.

[4] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71, 1996.

[5] D. M. Blei and M. I. Jordan. Modeling annotated data. In *SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 127–134, 2003.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[7] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311, 1993.

[8] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2), 1998.

[9] E. Chang, K. Goh, G. Sychay, and G. Wu. Cbsa: Content-based soft annotation for multimodal image retrieval using bayes point machines. *IEEE Transactions on Circuits and Systems for Video Technology Special Issue on Conceptual and Dynamical Aspects of Multimedia Content Description*, 13(1):26–38, 2003.

[10] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, 1996.

[11] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, pages 1197–1203, 1999.

[12] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[13] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[14] P. Duygulu, O. C. Ozcanli, and N. Papernick. Comparison of feature sets using multimedia translation. In *ISCIS XVIII - Eighteenth International Symposium on Computer and Information Sciences*, 2003.

[15] H. Feng and T.-S. Chua. A bootstrapping approach to annotating large image collection. In *MIR '03: Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 55–62, 2003.

[16] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[17] N. Ghamrawi and A. McCallum. Collective multi-label classification. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 195–200, 2005.

[18] S. E. Grigorescu, N. Petkov, and P. Kruizinga. Comparison of texture features based on gabor filters. *IEEE Transactions on Image Processing*, 11(10):1160–1167, 2002.

[19] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2001.

[20] R. Herbrich, T. Graepel, and C. Campbell. Bayes point machines. *Journal of Machine Learning Research*, 1:245–279, 2001.

[21] T. Hofmann. Learning and representing topic. a hierarchical mixture model for word occurrences in document databases. In *Proceedings of the Conference for Automated Learning and Discovery (CONALD)*, 1998.

[22] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196, 2001.

[23] T. Hofmann and J. Puzicha. Statistical models for co-occurrence data. In *AI Memo 1625, CBCL Memo 159, Artificial Intelligence Laboratory and Center for Biological and Computational Learning*, Cambridge, MA, USA, 1998. Massachusetts Institute of Technology.

[24] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 119–126, 2003.

[25] F. Kang and R. Jin. Symmetric statistical translation models for automatic image annotation. In *Proceedings of the fifth SIAM International Conference on Data Mining*, pages 616–620, 2005.

[26] F. Kang, R. Jin, and J. Y. Chai. Regularizing translation models for better automatic image annotation. In *CIKM '04: Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management*, pages 350–359, 2004.

[27] F. Kang, R. Jin, and R. Sukthankar. Correlated label propagation with application to multi-label learning. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1719–1726, 2006.

[28] P. D. Kobus Barnard and D. Forsyth. Clustering art. In *CVPR '01: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 434–439, 2001.

[29] E. B. Kong and T. G. Dietterich. Error correcting output coding corrects bias and variance. In *ICML '95: Proceedings of the 24th International Conference on Machine learning*, pages 313–321, 1995.

[30] V. Lavrenko, M. Choquette, and W. B. Croft. Cross-lingual relevance models. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 175–182, 2002.

[31] V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR '01: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 120–127, 2001.

[32] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *Advances in Neural Information Processing Systems 16*, 2003.

[33] J. Li and J. Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9):1075–1088, 2003.

[34] O. Maron and T. Lozano-Prez. A framework for multiple-instance learning. In *NIPS '97: Proceedings of the 1997 conference on Advances in neural information processing systems 10*, 1998.

[35] O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classification. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 341–349, 1998.

[36] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[37] F. Monay and D. Gatica-Perez. On image auto-annotation with latent apace models. In *MULTIMEDIA '03: Proceedings of the eleventh ACM International Conference on Multimedia*, pages 275–278, 2003.

[38] F. Monay and D. Gatica-Perez. Plsa-based image auto-annotation: constraining the latent space. In *MULTIMEDIA '04: Proceedings of the 12th Annual ACM International Conference on Multimedia*, pages 348–351, 2004.

[39] Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *Proceedings of the International Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999.

[40] R. G. Parker and R. L. Rardin. *Discrete Optimization*. Academic Press, 1988.

[41] N. d. F. Pinar Duygulu, Kobus Barnard and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV' 02, Proceddings of the Seventh European Conference on Computer Vision*, pages 97–112, 2002.

[42] Y. Rui, T. S. Huang, and S.-F. Chang. Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10(4):39–62, 1999.

[43] L. G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice Hall, 2001.

[44] J. Shi and J. Malik. Normalized cuts and image segmentation. In *CVPR '97: Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 731–737, 1997.

[45] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[46] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *Advances in Neural Information Processing Systems 16*, 2003.

[47] T. P. Weldon, W. E. Higgins, and D. F. Dunn. Efficient gabor filter design for texture segmentation. *Pattern Recognition*, 29(12):2005–2015, Dec. 1996.

[48] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Information Systems*, 2(2):179–214, 2004.

[49] C. Zhai and J. D. Lafferty. Two-stage language models for information retrieval. In *SIGIR '02: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 49–56, 2002.

[50] D. Zhang and G. Lu. Content based shape retrieval using different shape descriptors: A comparative study. In *2001 IEEE International Conference on Multimedia and Expo (ICME2001)*, pages 317–320, 2001.

[51] S. Zhu, X. Ji, W. Xu, and Y. Gong. Multi-labelled classification using maximum entropy method. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 274–281, 2005.