A Stratification-Based Approach to Accurate and Fast Image Annotation

Jianye Ye[1], Xiangdong Zhou[1], Jian Pei[2], Lian Chen[1], and Liang Zhang[1]

[1] Department of Computing and Information Technology, Fudan University Shanghai, China, 200433. {042021123,xdzhou,042021119,zhangl}@fudan.edu.cn
[2] School of Computing Science, Simon Fraser University, Canada. jpei@cs.sfu.ca

Abstract. Image annotation is an important research problem in content-based image retrieval (CBIR) and computer vision with broad applications. A major challenge is the so-called "semantic gap" between the low-level visual features and the high-level semantic concepts. It is difficult to effectively annotate and extract semantic concepts from an image. In an image with multiple semantic concepts, different objects corresponding to different concepts may often appear in different parts of the image. If we can properly partition the image into regions, it is likely that the semantic concepts are better represented in the regions and thus the annotation of the image as a whole can be more accurate. Motivated by this observation, in this paper we develop a novel *stratification-based* approach to image annotation. First, an image is segmented into some likely meaningful regions. Each region is represented by a set of discretized visual features. A naïve Bayesian method is proposed to model the relationship between the discrete visual features and the semantic concepts. The topic-concept distribution and the significance of the regions in the image are also considered. An extensive experimental study using real data sets shows that our method significantly outperforms many traditional methods. It is comparable to the state-of-the-art Continuous-space Relevance Model in accuracy, but is much more efficient – it is over 200 times faster in our experiments.

1 Introduction

Image annotation is an important research problem in content-based image retrieval (CBIR) and computer vision with broad applications. For a given image, such as a picture, we want to extract the semantics of the image in the form of a list of semantic concepts (or called semantic keywords). For example, the image at the top of Figure 1 can be annotated using three semantic concepts: *tiger*, *stone*, and *grass*.

Typically, automatic annotation can be achieved by supervised learning. A training set of images that are annotated by human experts is provided to train and test an annotation system. After training, the annotation system is to annotate images that are not in the training set. Therefore, the critical problem becomes how to efficiently build an accurate model from the training data set and apply the model in the annotation.

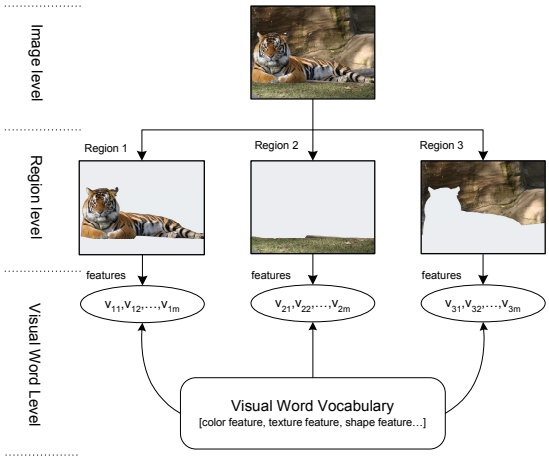


Fig. 1. The stratification Framework

Many methods were developed in the previous studies to build effective models for accurate image annotation. Please see Section 4 for a brief review. While the existing methods differ from each other in one way or the other, most of them *treat an image as a set of image blobs* and analyze the semantic concept of each image blob (*that is in the image region level*) to build a vocabulary of *blob word* to represent the whole image.

The essential idea is that an image often has more than one object and thus corresponds to multiple semantic concepts. Treating an image as a whole may not help to identify the features that are highly informative for some specific semantic concepts. For example, in the image shown at the top of Figure 1, the area of yellow/white stripes may strongly suggest the semantic concept of *tiger*, and the area of green with texture may indicate the existence of the semantic concept *grass*. If we can divide the image properly, then we may be able to make a good annotation of the image.

However, classifying image blobs into blob words with respect to correct semantic concepts is still a challenging problem due to the semantic gap. In fact, it is basically another image annotation. Therefore, instead of using blob words to describe the semantic of the image, we believe that it is more effective to represent semantic concepts of images by some low level visual feature descriptors (called "*visual words*") just analogous to the roles of keywords in text documents[3]. Moreover, *visual words* can be learned more accurately at the image blob level than from the whole image.

1.1 General Ideas

In this paper, we develop a novel *stratification-based* approach to effective and fast image annotation. The framework is shown in Figure 1.

[3] Recall that the essential building blocks of text documents are keywords instead of paragraphs.

First, *images are properly segmented into multiple regions so that each region likely corresponds to a single object in the image.* This stratification approach is an application of the divide-and-conquer strategy. The intuition is that fewer objects an area corresponds to, more probably the semantic concepts can be correctly annotated. In order to segment the image properly, we apply the *normalized-cuts method* [13], which partitions an image into regions such that each region is consistent in visual features.

Second, *the visual features of regions should be used to construct the features for annotation.* Naïvely, one may want to extract a semantic concept (i.e., a *blob word*) for each region, and simply take the set of the semantic concepts as the annotation of the whole image. However, such a naïve method may not be effective. The segmentation of images into regions may not be very reliable. Regions are related in an image. That is, directly extracting semantic concepts from a region may not be accurate. Moreover, regions should not be treated equally. For example, a region with a substantially larger area should carry a heavier weight in the determination of the semantics of an image.

Thus, instead of the naïve method, we adopt a more comprehensive approach. We extract the visual features from each region, such as color and texture. Then a learning method is employed to discretize the corresponding visual features of these regions to form a *visual word vocabulary*, analogous to the keyword vocabulary in the text document processing domain. With these visual words, we can describe the image more accurately than using *blob words*.

Last, *a naïve Bayesian method is used to annotate images, and the factors of topic-concept distribution and contributions of regions are also considered.* In the training phase, a naïve Bayesian model is built to capture the correlation between semantic concepts and visual words. Moreover, in many data sets, images are divided according to topics. Two images are in the same topic fold if they share similar semantic concepts. In the annotation phase, we also consider the topic-concept distribution, i.e., the topic information is used in the annotation. We also consider the size of regions – the corresponding visual words are weighted in the annotation phase.

1.2 Our Contributions and the Organization of the Paper

We develop a novel stratification-based approach for image annotation. While the general ideas are described in Section 1.1, the concrete technical approach is developed in the rest of the paper. We conduct an extensive performance study using real data sets and compare with the state-of-the-art methods. The experimental results clearly show that our approach outperforms many traditional image annotation methods, and has an accuracy comparable to the state-of-the-art Continuous-space Relevance Model, but our approach is dramatically more efficient – it is over 200 times faster in our experiments.

The rest of the paper is organized as follows. In Section 2, we discuss how to stratify images and how to extract visual words from regions. Annotation of images using visual words and corresponding algorithms are addressed in Section 3. The factors of topic information and weighted visual words are also considered. We review related work in Section 4. An extensive performance study is reported in Section 5. The paper concludes in Section 6.

2 Stratification of Images and Extraction of Visual Words

2.1 Stratification

Images can be segmented into several sub-regions with particular semantic meanings, analogous to partitioning a text document to paragraphs or sentences. For each region, more lower-level descriptors can be derived such as the visual features, which are analogous to the words in a text document. We can describe the structural organization of images in a hierarchy of three levels, namely images, regions, and visual descriptors, as shown in Figure 1.

According to the idea of stratified image representation, we process each image in the following steps:

1. Image segmentation.We segment each image into different regions using the normalized-cuts method [13], the region is consistent in the visual features.
2. Visual feature extraction for each region, such as color, texture, etc.
3. Image feature discretization and visual word vocabulary generation using a minimal-entropy based method [5].

Given a visual word vocabulary $V = \{v_1, v_2, \ldots, v_k\}$, an image I can be represented by a set of segmented regions: $I = \{r_1, \ldots, r_m\}$. Each region r_i is represented by a fixed number of visual words: $r_i = \{f_{i1}, \ldots, f_{iM}\}, f_{ij} \in V$. We define an *indicator function* g on regions and visual words:

$$g(r, f) = \begin{cases} 1 \text{ if region } r \text{ contains visual word } f \\ 0 \text{ otherwise} \end{cases} \quad (1)$$

By summing up visual words from all regions in an image, we represent an image as a vector of visual words: $I = (s_1, \ldots, s_k)$, where s_i is the number of regions containing visual word v_i: $s_i = \sum_{j=1}^{m} g(r_j, v_i)$.

Comparing to previous discrete models which represent an image by a branch of blob words, our stratified image model has two major advantages.

First, generally, the number of blobs in an image is too small after segmentation. For example, in the Corel dataset used by [4, 6, 7, 11], an image only has 1-10 blobs. Nevertheless, an image can have 36-360 visual words in the same dataset, which help our model behave more accurately in probability estimation.

Second, in previous discrete models, a region corresponds to a single blob word after region clustering. All low-level visual feature information are ignored. There is no remedy if a region is clustered incorrectly. Although a similar problem exists in feature discretization, by keeping all low-level visual feature elements we greatly reduce the impact of an incorrect classification.

2.2 Extraction of Visual Words

To generate a visual word vocabulary, we need to convert feature vectors of regions into discrete-valued vectors. Many methods [5, 3, 9, 6] can be used to discretize real-valued data. We employ a supervised method presented by Fayyad and Irani [5] to discrete real-valued visual features. The method is based on a minimal-entropy heuristic. As shown in [3], this method often achieves the best performance among supervised discretization methods.

Minimal-Entropy based Discretization Given dataset $S = \{s_1, \ldots, s_n\}$ and class label set C, let v_i be the continuous data value of s_i, and c_i be the class label of s_i. The entropy of dataset S is:

$$E(S) = -\sum_{c \in C} \frac{|S(c)|}{|S|} log \frac{|S(c)|}{|S|}, \tag{2}$$

where $S(c)$ denotes all data points with class label c in S. A boundary T partitions S into two sets $S_1 = \{s_j | v_j < T\}$ and $S_2 = \{s_k | v_k > T\}$. The best boundary T minimizes the entropy after partition, which is:

$$E(T, S) = \frac{|S_1|}{|S|} E(S_1) + \frac{|S_2|}{|S|} E(S_2). \tag{3}$$

Then the boundary selection is repeated on S_1 and S_2. The partitioning process is applied recursively until some certain stop criteria is satisfied. Information Gain is defined as the entropy reduction after a partition.

$$Gain(T, S) = E(S) - E(T, S). \tag{4}$$

In Fayyad and Irani's approach [5], the recursive partitioning process stops iff:

$$Gain(T, S) < \frac{log_2(|S| - 1)}{|S|} + \frac{\Delta(T, S)}{|S|} \tag{5}$$

$$\Delta(T, S) = log_2(3^k - 2) - [k \cdot E(S) - k_1 \cdot E(S_1) - k_2 \cdot E(S_2)], \tag{6}$$

where k_i denotes the number of class labels represented in S_i. Since partitions are evaluated independently using this criteria during the recursive procedure, the continuous space is not evenly partitioned. Areas having relative high entropy are likely to be partitioned more finely.

Discretization of Visual Features We describe each region using a 36-dimensional visual feature vector, which includes the average rgb color, the average lab color, area, the mean oriented energy and the area, etc. We assume that each image region inherits all keywords from its parent image. The keywords associated with an image region serve as the class labels for visual features in discretization. Given all data values on one dimension of visual features along with associated class labels, the minimal-entropy based discretization(MED) is applied to this dimension. Since MED can only handle data with a single label, data with multiple labels needs to be decomposed into multiple data entries each with a single label. For example, we should decompose a data entry valued at 0.35 with class labels "tiger", "grass", "sky" into 3 data entries all valued at 0.35 but with class labels "tiger", "grass", and "sky", respectively.

After discretization, each discrete bin can be viewed as a visual word. All discrete bins on all dimensions form the visual word vocabulary. The size of the visual word vocabulary is one of the key aspects influencing the model performance. To control the granularity of discretization, we add a parameter ψ (the

value of ψ is 30 in our experiment) to tight the stop criteria:

$$Gain(T,S) < \frac{log_2(N-1)}{|S|} + \frac{\Delta(T,S)}{\psi \cdot |S|} \tag{7}$$

We also use a minimum partition size to constrain the discretization not to partition too finely in areas which have relative high entropy. After applying this modified MED on every dimension of visual features, we build a vocabulary having 424 visual words.

3 Annotation of Images

3.1 A Naïve Bayesian Model

Given an un-annotated image I, we first employ a segmentation method [13] to split it into regions $r_1, \ldots, r_n$. Then a feature-extraction algorithm [4] is employed to derive a feature vector from a region. Let f_i denote the set of visual words extracted from region r_i , and $f_i = \{a_{i1}, a_{i2}, \ldots, a_{im}\}$. For each keyword w in pre-defined keyword vocabulary W, we use the logarithmic probability $\log P(w|f_1, \ldots, f_n)$ to estimate the relationship between the image and word w.

$$\log P(w|f_1, \ldots, f_n) = \log \frac{P(w) \cdot P(f_1, \ldots, f_n|w)}{P(f_1, \ldots, f_n)} \tag{8}$$

Without loss of generality, we can assume that $p(w)$ is a constant, and $P(f_1, \ldots, f_n)$ is independent of word w, which can be neglected in the computing of the annotation. Thus, we have:

$$\log P(w|f_1, \ldots, f_n) \propto \log P(f_1, \ldots, f_n|w). \tag{9}$$

The set of visual words from different regions are assumed to be independent.

$$\log P(f_1, \ldots, f_n|w) = \sum_{i=1}^{n} \log P(f_i|w). \tag{10}$$

For each $f_i = (a_{i1}, \ldots, a_{im})$, we again treat every visual word in the set statistically independent.

$$\log P(f_i|w) = \sum_{j=1}^{m} \log P(a_{ij}|w). \tag{11}$$

From (9), (10), and (11), the posterior probability distribution of words for a given un-annotated image I is derived as follows:

$$\log P(w|f_1, \ldots, f_n) \propto \sum_{i=1}^{n} \sum_{j=1}^{m} \log P(a_{ij}|w) \tag{12}$$

$$P(w|I) = P(w|f_1, \ldots, f_n) = \frac{\exp\Big(\sum_{i=1}^{n}\sum_{j=1}^{m} \log P(a_{ij}|w)\Big)}{\sum_{w \in W} \exp\Big(\sum_{i=1}^{n}\sum_{j=1}^{m} \log P(a_{ij}|w)\Big)} \tag{13}$$

The denominator of (13) is a normalizing factor. According to our stratification model, an image is represented by a visual word vector $(s_1, \ldots, s_k)$. Therefore, we can simplify (13):

$$P(w|I) = \frac{\exp(\sum_{i=1}^{k} s_i \log P(v_i|w))}{\sum_{w \in W} \exp(\sum_{i=1}^{k} s_i \log P(v_i|w))} \tag{14}$$

where $\{v_1, v_2, \ldots, v_k\}$ denotes the visual word vocabulary. $P(v_i|w)$ can be estimated by counting the occurence of v_i in all images annotated with w.

3.2 Annotation with Weighted Visual Words

The commonsense tells that *larger the regions, often more decisive to the theme of the image*. Thus, the larger regions should be assigned heavier weights. In this subsection, we discuss an extension to our basic method. In this extended method, the size of regions is considered.

Given image I, Let x_r denote the size proportion of region r to the whole image. The weight of region r is defined according to x_r:

$$w_r = n \cdot \frac{x_r^{\alpha}}{\sum_{r' \in I} x_{r'}^{\alpha}}, \tag{15}$$

where n denotes the number of regions in image I and α determines the degree of x_r affecting the weight of visual words in region r. x_r would have more significant influence on region weighting as α increases. We optimize the performance on a small test set to pick the best α value.

We substitute a weight function h for the indicator function g defined in section 3.2:

$$h(r, f) = \begin{cases} w_r \text{ if region } r \text{ contains visual word } f \\ 0 \quad \text{otherwise} \end{cases} \tag{16}$$

In a weighted stratified image model, we represent an image as a vector of visual words: $I = (s_1, \ldots, s_k)$, where s_i is the total weight of regions containing the visual word v_i: $s_i = \sum_{j=1}^{n} h(r_j, v_i)$.

3.3 Annotation with Topic Information

Most datasets, which can be used as training sets like Corel dataset [4, 7, 11, 6], are categorized into folds. Images within the same fold may have different manual

annotation but they all share a similar topic. *Can we use the topic information to improve the annotation?*

For each fold in a training set, a topic model is a word distribution on the annotations of the images in this fold. We estimate the probability of word occurrence by counting the manually annotated words of images in the fold. The heuristics here, which incorporates the topic distribution in the annotation, is that *the keywords with a higher probability in the fold are "popular", and thus may have a higher probability to be used to label new images exhibiting the similar topic.*

In order to use the topic information, the annotating process for a given un-annotated image is divided into two phases. First, we estimate the probability distribution of the keywords in the un-annotated image by employing equation 13.

Second, given all topic models in the training set, we pick a topic model t which has the most similar word distribution to the un-annotated image. We employ the negative Kullback-Liebler divergence to model the similarity of two distribution [2]. To be more specific, let $P(.|I)$ denote the keyword distribution of un-annotated image I, and TS denote the set of all topic models in the training set. The topic $T(I)$ of image I is defined as follows:

$$T(I) = \arg\max_{T \in TS} \sum_{w \in W} P(w|I) \cdot \log \frac{P(w|T)}{P(w|I)} \tag{17}$$

Then the keyword distribution of I and $T(I)$ are incorporated to determine the final annotation of the image I. Let $G(.|I)$ denote the above mixture distribution, then for each keyword w, we have:

$$G(w|I) = \beta \cdot P(w|I) + (1 - \beta) \cdot P(w|T(I)) \tag{18}$$

where β determines the degree of interpolation between $P(.|I)$ and $P(.|T(I))$. $P(w|T(I))$ can be estimated by counting the occurence of w in all images in Topic $T(I)$. All the keywords are ranked according to $G(.|I)$, and the top-k keywords with the largest probability are picked out as the annotation of the image.

4 Related Work

To label an image, most previous methods employ segmentation to divide the images into blobs [4, 7, 11, 8], or even grids [12, 6], and then the joint probability is estimated on the image regions and the keywords of a pre-defined vocabulary.

As one of the early studies on image annotation, the co-occurrence model proposed by Mori, et al. [12] assigns image annotations to every region of the image. In this model, auto annotation is based on the frequency of the co-occurrence of a word and an image region. Li and Wang [10] proposed a multi-resolution 2D hidden markov model to view the generation of images as a random process following an underlying probability distribution. The model represents an image by feature vectors under several resolutions. This hierarchical structure helps to catch the spatial context of an image.

More recently, Duygulu, et al. [4] proposed the translation model. In this model, image regions are clustered into a vocabulary of blobs. Each image is represented by a number of blobs from the vocabulary. To annotate images, they employ a classical translation machine model to translate a vocabulary of blobs to a vocabulary of words. Jeon, et al. [7] proposed a cross-media relevance model. Instead of finding the corresponding word for each blob, this model learns the joint probability of all blobs and words within an image. Their experiments showed that the method outperforms the co-occurrence model and translation model significantly on a 5000-image corel dataset. Lavrenko, et al. [11] proposed the continuous-space relevance model which can be viewed as a continuous version of CMRM. They employ a non-parametric kernel-based density estimate to learn the probability of continuous feature vector occurrence. It has a much better performance than the cross-media relevance model but suffers from low efficiency.

Blei and Jordan [1] extended the latent Dirichlet allocation (LDA) model to relate words with images. Feng, Manmatha and Lavrenko [6] proposed the multiple Bernoulli relevance model which uses a Bernoulli process to generate words instead of assuming a multinomial distribution over words. Jin, Cai and Si [8] proposed a coherent language model for image annotation that takes into account the word-word correlation.

5 Empirical Study

5.1 Dataset

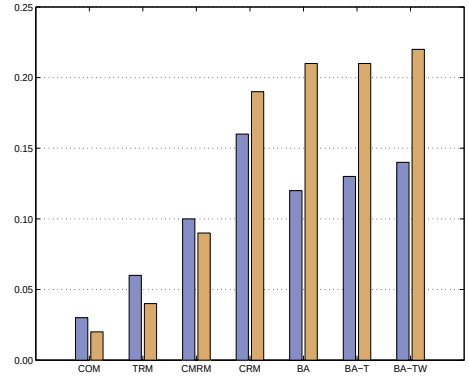
In our evaluation, we use a 5000-image corel dataset provided by Duygulu [4]. The data set is available at http://www.cs.arizona.edu/people/kobus/research/data/eccv2002. All images are already segmented into regions using normalized-cut algorithm [13]. Each image contains 1-10 image regions. A 36 dimensional feature vector is extracted from each region. There are 371 different concepts used in the manual annotation. Each image is associated with 1-5 words to describe the essential semantics of the image. This dataset was first used by Duygulu to evaluate the translation model [4]. It was also used by Jeon and Lavrenko to evaluate the cross-media relevance model [7] and the continuous-space relevance model [11].

We randomly select 500 images in the data set as the test set and the other 4500 images as the training set.

5.2 Performance Comparison

We compare the annotation performance of our model with other four different models, including the co-occurrence model [12], the translation model [4], the cross-media relevance model [7] and the continuous-space relevance model [11].

In our experiments, we first annotate each image in the test set with 5 words. After auto annotation, we use each word from the vocabulary to perform single-word retrieval in the test set. We judge whether an image is correctly retrieved



Models	COM	TRM	CMRM	CRM	BA	BA-T	BA-TW
mean-Precision	0.03	0.06	0.10	0.16	0.12	0.13	0.14
mean-Recall	0.02	0.04	0.09	0.19	0.21	0.21	0.22

Fig. 2. mean-Precision and mean-Recall

by looking at its manual annotation. We define F measure:

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

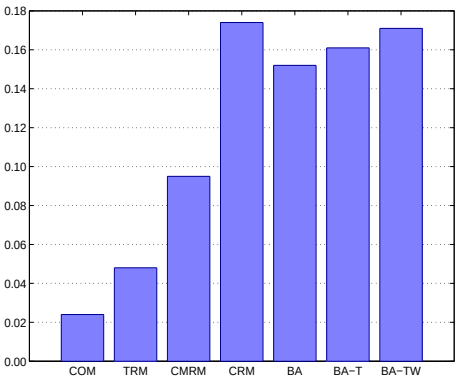
To examine the contributions of different aspects of our model more clearly, we denote the Bayesian approach without incorporating region weighting and topic model as BA, the Bayesian approach only incorporating topic model as BA-T, the Bayesian approach with region weighting and topic model as BA-TW.

Figures 2 and 3 clearly show that the annotation performance of our model is significantly better than the co-occurrence model, the translation model and the cross-media relevance model. BA-TW has a noticeable improvement over BA and BA-T. The continuous-space relevance model and BA-TW have similar performance in F-measure. Please note that the continuous-space relevance model uses real-valued feature vectors. This is often more effective than the discretize values, but is also much more costly in terms of runtime, as shown in the next subsection.

5.3 Efficiency Comparison

We compare the annotation efficiency of our model with the cross-media relevance model [7] and the continuous-space relevance model [11]. We focus on the efficiency of the three models in annotating new images, so the time cost in training and the time cost in segmentation of un-annotated images are not taken into consideration. We record the time for the three models to annotate 500 images. All experiments is done on a laptop PC which has one P4 1.8Ghz CPU and 384 Megabytes main memory.

Table 1 shows that our model is about two orders of magnitude faster than CRM and about 3 times faster than CMRM. To understand the experimental



Models	COM	TRM	CMRM	CRM	BA	BA-T	BA-TW
F-measure	0.024	0.048	0.095	0.174	0.152	0.161	0.171

Fig. 3. F-measure in Models

Table 1. Models Efficiency Comparison

Models	CMRM	CRM	BA	BA-T	BA-TW
Time(s)	59.8	4513.1	16.8	21.7	21.8

results, we analyze the computational complexity of annotating one image in the three models as follows.

From (13), we derive that the computational complexity for our model to annotate one image is $O(|W| \times |V|)$, where W is the text word vocabulary and V is the visual word vocabulary. A large-scale training set is very useful for models to learn the relationships between images and words. Since the computational cost of annotation in CMRM and CRM depends on the size of training set, our model has a better scalability than these two models on large-scale training sets.

6 Conclusions

In this paper, we proposed a stratified image description and a Bayesian model for image annotation. We showed that this model has a good balance between performance and efficiency, as well as a good scalability. Our model employs the Minimal-Entropy based Method for feature discretization. We use the region weighting technique and the topic model-based enhancement to improve the annotation performance.

As the future work, we will explore some semi-naïve Bayesian methods for our model. We believe that a better understanding in the relationship between regions is very useful for learning the semantics of an image. We will also consider using some multi-dimensional discretization methods to discretize feature vectors in order to catch the dependency of features.

7 Acknowledgement

This work was supported in part by the NSF of China under grant number 60403018, NSF of Shanghai of China under grant number 04ZR14011, the NSF Grant IIS-0308001, and the NSERC Discovery Grant 312194-05. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

1. Blei, D., Jordan, M.I. (2003): Modeling annotated data. Proc. of the 26th Annual International ACM SIGIR Conference(pp. 127–134). Toronto, Canada: ACM.
2. Croft, W.B. (2000): Combining approaches to information retrieval. In W. B. Croft (Ed.), Advances in information retrieval. Cambridge, MA, USA: MIT Press.
3. Dougherty, J., Kohavi, R., Sahami, M. (1995): Supervised and Unsupervised Discretization of Continuous Features. Proc. of the Twelfth International Conference on Machine Learning (pp. 194–202). Tahoe City, California, USA: Morgan Kaufmann.
4. Duygulu, P., Barnard, K., de Freitas, N., Forsyth, D. (2002): Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. Proc. of the Seventh European Conference on Computer Vision (pp. 97–112). Copenhagen, Denmark: Springer.
5. Fayyad, U.M., Irani, K.B. (1993): Multi-interval discretization of continuous-valued attributes for classification learning. Proc. of the 13th International Joint Conference on Artificial Intelligence (pp. 1022–1029). Chambery, France: Morgan Kaufmann.
6. Feng, S.L., Manmatha, R., Lavrenko, V. (2004): Multiple Bernoulli Relevance Models for Image and Video Annotation. IEEE Conference on Computer Vision and Pattern Recognition (pp. 1002–1009). Washington, DC.
7. Jeon, J., Lavrenko, V., Manmatha, R. (2003): Automatic Image Annotation and Retrieval using Cross-Media Relevance Models. Proc. of the 26th Annual International ACM SIGIR Conference (pp. 119–126). Toronto, Canada: ACM.
8. Jin, R., Cai, J.Y., Si, L. (2004): Effective Automatic Image Annotation Via A Coherent Language Model and Active Learning. Proc. of the 12th ACM Annual Conference on Multimedia (ACM MM 2004) New York, USA.
9. Kohavi, R., Sahami, M. (1996): Error-Based and Entropy-Based Discretization of Continuous Features. Proc. of the Second International Conference on Knowledge Discovery and Data Mining (pp. 114–119). Portland, Oregon, USA: AAAI Press.
10. Li, J., Wang, J.Z. (2003): Automatic linguistic indexing of pictures by a statistical modeling approach. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(9), 1075–1088.
11. Lavrenko, V., Manmatha, R., Jeon,J. (2004): A Model for Learning the Semantics of Pictures. Advances in Neural Information Processing Systems. Vancouver, British Columbia, Canada: MIT Press.
12. Mori, Y., Takahashi, H., Oka, R. (1999): Image-to-word transformation based on dividing and vector quantizing images with words. Proc. of the First International Workshop on Multimedia Intelligent Storage and Retrieval Management.
13. Shi, J., Malik, J. (2000): Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(9), 1075–1088.
14. Yang, Y., Webb, G.I. (2005): Discretization for data mining. In J. Wang (Ed.), Encyclopedia of data warehousing and mining. Idea Group Reference.