A Bootstrapping Approach to Annotating Large Image Collection

HuaMin FENG and Tat-Seng CHUA
School of Computing, National University of Singapore
{fenghm, chuats}@comp.nus.edu.sg

ABSTRACT

Huge amount of manual efforts are required to annotate large image/video archives with text annotations. Several recent works attempted to automate this task by employing supervised learning approaches to associate visual information extracted in segmented images with semantic concepts provided by associated text. The main limitation of such approaches, however, is that large labeled training corpus is still needed for effective learning, and semantically meaningful segmentation for images is in general unavailable. This paper explores the use of bootstrapping approach to tackle this problem. The idea is to start from a small set of labeled training examples, and successively annotate a larger set of unlabeled examples. This is done using the co-training approach, in which two "statistically independent" classifiers are used to co-train and co-annotate the unlabeled examples. An active learning approach is used to select the best examples to label at each stage of learning in order to maximize the learning objective. To accomplish this, we break the task of annotating images into the sub-tasks of: (a) segmenting images into meaningful units, (b) extracting appropriate features for the units, and (c) associating these features with text. Because of the uncertainty in sub-tasks (a) and (b), we adopt two independent segmentation methods (task a) and two independent sets of features (task b) to support co-training. We carried out experiments using a mid-sized image collection (comprising about 6,000 images from CorelCD, PhotoCD and Web) and demonstrated that our bootstrapping approach significantly improve the performance of annotation by about 10% in terms of F_1 measure as compared to the best results obtained from the traditional supervised learning approach. Moreover, the bootstrapping approach has the key advantage of requiring much fewer labeled examples in training.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: linguistic processing, thesaurus

H.3.3 [**Information Search and Retrieval**]: selection process

I.4.6 [**Segmentation**]: region partitioning

General Terms

Algorithms, design, experimentation

Keywords

Bootstrapping, co-training, image annotation, active learning

1. INTRODUCTION

Effective techniques are needed to model and search the content of large digital image/video archives. One such technique is query-by-example (QBE), in which users provide visual examples of the contents they seek. This allows images to be retrieved on the basis of content features such as the color, texture etc. However, such low-level content-based retrieval schemes have some obvious limitations that it is non-exact and is unable to support retrieval based on abstract concepts. Since most users wish to search in term of semantic concepts rather than visual contents[1], work in the image/video retrieval research has begun to shift from QBE to query-by-keyword (QBK). QBK allows users to search for images by specifying their own query in terms of a (limited) vocabulary of semantic concepts [2]. The main problem with adopting such an approach is that it shifts the problem of imprecise content-based retrieval to one of annotating the images with meaningful concepts (or keywords). Although many useful image collections come with keyword annotations, the annotations are normally incomplete, and there are many more images that do not have such annotations. Thus there is a need to develop automated or semi-automated techniques to annotate images with semantic concepts accurately and completely. Throughout this paper, we liberally use the term concept and keyword interchangeably, and define annotation as the process of associating concepts with new images (which include videos). The automated system that performs the annotation is called the ***Classifier***.

A popular approach to tackling the annotation problem is to adopt a supervised learning approach to train the Classifiers to perform the annotation of new images. The main practical problem with this approach is that a large labeled training corpus of both positive and negative examples is needed, and it is very tedious, time-consuming and error-prone to provide such training examples. Moreover, during the learning and application stages, the training set is fixed and not incremented. Thus if the domain changes over time or when a new domain is introduced, new labeled examples must be provided to ensure the effectiveness of the Classifier. In a way, such approach takes the learner as a "passive" recipient of data to be processed. This "passive" approach ignores the fact that, in many situations, the learner's most powerful tool is its ability to act, to gather data, and to influence the world that it is trying to understand.

To overcome the problem of needing to provide large labeled examples, an alternative is to adopt a bootstrapping cum active learning approach that requires only a small set of labeled examples to kick-start the learning process. Bootstrapping [3] uses a small set of labeled examples to bootstrap the process of annotating and learning from large unlabeled examples. To achieve bootstrapping, we need a way for the system to evaluate the quality of new annotated examples. This can be achieved by using the co-training technique [4] in which two "statistically independent" methods independently confirm the quality of new annotated examples, and learn from each other's results. Active learning studies the closed-up phenomenon of a learner selecting actions or making queries that influence what data are added to its training set [5]. Instead of passively using all the available instances for training as in the supervised case, active learning selects those that it considers as the most critical instances and repeatedly asks the human users to label them and including them into the labeled example set. Thus, active learning can eventually create a reliable Classifier from fewer labeled examples than supervised learning.

In this paper, we propose a framework based on bootstrapping and active learning to annotate large online image collections. To accomplish this, we break the task of annotating images into three sub-tasks of: (a) segmenting images into meaningful units, (b) extracting appropriate features for the units, and (c) associating the units in images with concepts. Thus, the problem of image annotation can be expressed as:

$$G^l (I_i) \approx G^l (S^p (I_i)) \approx \sum G^l (F^q (R^p_{ij})) \rightarrow \underline{L}_c \qquad (1)$$

$$S^p (I_i) \approx F^q (R^p_{ij}) \qquad (2)$$

Here $\underline{L}_c$ is the set of Lexicon or concepts used to annotate the images. Function $S^p(I_i)$ refers to a transformation of the content of image I_i. An example of such transformation is the segmentation of the image by converting its contents into meaningful units (or regions/ blocks), i.e $S^p(I_i) \rightarrow \sum R^p_{ij}$. The function $F^q(R_{ij})$ selects a set of features to model each unit/region, R_{ij}. Finally, the function $G^l(I_i)$ performs the annotation that maps an image to a set of concepts in the Lexicon $\underline{L}_c$. As expressed in Equation (1), if we adopt an approach to segment the image contents into sub-units R_{ij}, then $G^l(I_i)$ can be approximated by an equivalent function to annotate each sub-unit separately and integrating the results of annotations for the overall image.

Equations (1-2) indicate that we are able to substitute different models to accomplish each function in the annotation process independently. For example, we may choose to perform $S^p(I_i)$ by either segmenting the image I_i into regions or dividing it equally into fixed blocks. We may use different function $F^q(R_{ij})$ to map the content of each sub-unit R_{ij} into different set of features. In this research, we aim to perform co-training at two levels. First at the image level by adopting different function $S^p(I_i)$ through the use of different region segmentation methods. Second, at the region level, by adopting different function $F^q(R_{ij})$ in selecting different set of features to represent the sub-unit contents. Finally we employ a learning function $G^l(I_i)$ to perform the annotation by associating the contents of sub-unit R_{ij} with a set of concepts in $\underline{L}_c$. Throughout the above three stages, we use the idea of co-training to complement the strength and overcome the weakness of each parallel model. We tested our bootstrapping framework using a mid-sized image collection (comprising about 6,000 images from CorelCD, PhotoCD and Web) and demonstrated that our bootstrapping approach significantly improve the performance of annotation by about 10% in terms of F_1 measure, as compared to the best results obtained for the traditional supervised learning approach. Of course, the bootstrapping approach has the key advantage that it requires much fewer labeled examples during training.

The rest of the paper is organized as follows: Section 2 reviews related research, and Section 3 presents our bootstrapping framework. Section 4 describes details of our co-training of region classifiers. Section 5 presents the concept disambiguation at the image level. Section 6 gives the initial experiment results and discussion. Section 7 concludes the paper with discussion for future work.

2. RELATED WORK

Several recent works deal with the automatic or semi-automatic attachment of keywords[6-8] and semantic search [9, 10] for image databases. Mori et al. [6] were among the earliest to perform "image-to-word transformation based on dividing and vector quantizing images with words". They assigned keywords to images in the training set at the image level. They divided the image into fixed-size blocks (function $S^p(I_i)$) where each block inherits the whole set of keywords associated with the image. Blocks are clustered based on the vector quantization feature (function $F^q(R_{ij})$) and the clusters are in turn used to predict the keywords for new images. The advantage of this approach is that it does not need to perform image segmentation, which is often unreliable. However, due to rough fixed block size segmentation, the extracted blocks are unable to model objects effectively, leading to poor annotation performance.

Instead of using fixed size blocks, Barnard and Forsyth [7] performed Blob-World segmentation [11] (function $S^p(I_i)$) and associate keywords to Blob-world regions in the training set. For each region, they extracted the color, texture and shape as features (function $F^q(R_{ij})$). They employed a hierarchical model (function $G^l(I_i)$) in the form of a tree. The model combines the asymmetric clustering model which maps the documents (words and image segments) into clusters, and symmetric clustering model which models the joint distribution of documents and features. Document clusters correspond to leaves of tree, while node of the tree is uniquely determined by the level and cluster. The hierarchical structure is constructed via EM algorithm. Due to EM algorithm and the high dimensionality of the image feature vector, it is time-consuming to train and apply the model to perform annotation. Also, there are the problems of unreliable region segmentation, and over-fitting.

Chang et al. [12] proposed a content-based soft annotation for multimodal image retrieval using Bayes point machine (BPM). BMP is a learning approach to approximate the Bayesian inference for linear classifiers in a kernel space [13]. They employed image level content analysis (function $S^p(I_i)$) and associated selected set of keywords with each image. Through the application of BMP, each image is assigned one keyword vector, with each keyword in the vector assigned a confidence factor. Thus during the annotation process, they can choose those words with high confidence as the annotations of new images. The main limitation of this approach is that it does not associate keywords with meaningful units and may thus suffer from poor annotation accuracy.

Another approach to overcome the segmentation problem is proposed by Wang and Li [14]. They assigned a textual description of concepts for an image collection and employed a 2-

D multi-resolution HMM (function $G^l(I_i)$) to capture the cross blocks and cross resolution dependencies between blocks for the entire image collection. Given a new image, the feature vector (function $F^q(R_{ij})$) of the image is compared with the trained models, and statistically significant terms are extracted to annotate the image. However, because of the use of 4x4 fixed-size block (function $S^p(I_i)$), this approach might inherit the same problems as in [6].

The above approaches are based on the traditional "passive" supervised learning scheme. The training set is fixed and much manual annotation work is needed to come up with reasonable sized labeled set. Worse still, if the domain changes and the training becomes "inappropriate", the system is unable to adjust or augment the training set during the learning or application stage. Therefore, the final result will be badly affected. Bootstrapping and active learning have been proposed as possible solution to alleviate these problems. Bootstrapping methods aim to use their own ability to collect new data to augment the training and gradually move towards the "optimal" learning state. There are many literatures on this topic. Blum and Mitchell [4] proposed a co-training algorithm, a collaborative bootstrapping approach, based on the conditional independence ("view independence") assumption. The algorithm conducts two bootstrapping processes in alternative mode and makes them collaborate with each other. More specifically, it repeatedly trains two classifiers from the labeled data, labels some unlabelled data with the two classifiers, and exchanges the newly labeled data between the two classifiers. In the co-training algorithm, one classifier always asks the other classifier to label the most certain data for the collaborator. Since the assumption of view independence cannot always be met in practice, Collins and Singer[15] proposed a co-training algorithm based on "agreement" between the classifiers. Muslea et al. [16] introduced an algorithm called co-testing and is designed to apply to problems with redundant views or problems with multiple disjoint sets of attributes (features) that can be used to learn the target attributes (class labels). Nigam and Ghani [17] empirically demonstrated that even bootstrapping (co-training) that violates the view independent assumption (by simply randomly splitting the feature set to derive two classifiers) can still work better than bootstrapping without a feature split (i.e., bootstrapping with a single classifier). Last but not least, Cao and Li, et al. [18] proposed the use of uncertainty reduction in co-training, and indicated that uncertainty reduction is important for enhancing the performance of collaborative bootstrapping. They showed that the natural split of feature in co-training algorithm produced the best results. They also gave their own collaborative bootstrapping algorithm driven by the uncertainty reduction. Specifically, they used one classifier to select the most uncertain unlabelled data and ask the other classifier to label.

The main issue in active learning is how to choose the most critical instances. The use of uncertainty measurement is one of the popular strategies. Lewis and Gale [19] proposed an approach called uncertainty sampling. The idea is to use only one classifier to not only tell which class a sample in the unlabelled set is, but also to give an uncertain score to that sample. The next sample that requires manually labeling by human is chosen based on one which the classifier has the least confidence. Zhang et Chen[20] proposed an active learning framework for content-based information retrieval. They used active learning to determine which objects should be annotated. During the learning stage, the system selects samples automatically for the human annotator based on the criterion that annotating these samples will lead to an overall decrease in the uncertainty of the system.

3. THE BOOTSTRAPPING FRAMEWORK FOR IMAGE ANNOTATION

As discussed in Section 1, the problem of image annotation can be divided into 3 separate stages as expressed in Equations (1-2). We can apply the concept of co-training and active learning at each stage to perform bootstrapping. In order to perform co-training at the appropriate stage of image annotation, we need to devise separate "conditional independent" models.

For the annotation stage at the image collection level, we are interested in deriving a concept mapping function $G^l(I_i)$ that maps an image I_i into a set of concepts in $\underline{L}_c$. In order to simplify the problem, we divide the image content into smaller and hopefully more meaningful sub-units based on regions [11, 21] or fixed sized blocks [6]. We then perform the association of regions to concepts. Many learning functions may be used to perform the association. Here we adopt different variants of SVM with decision tree learning to train $G^l(R_{ij})$.

For the segmentation stage at the image level, we are concerned with segmenting the image into appropriate content units (or function $S^p(I_i)$) that best approximate the image content. The idea here is to adopt two independent image segmentation methods, denoted say by functions $S^p(I_i)$, for $p \in [u, b]$, to segment the image into two separate sets of objects/regions. Because of the unreliability and uncertainty in image segmentation, we expect the same image to be segmented into different set of often overlapping regions by different methods. Thus in addition to playing the role of independent methods for the purpose of co-training, the two segmentation methods can also be used to overcome the unreliability of object or region segmentation. One possible use here is to train the function G^l that maps each region independently into concepts, and use the correlation between the overlapping regions, and conflicting concepts, to disambiguate the learnt concepts to arrive at the final annotation for each region. We call this process *concept disambiguation*.

For the feature extraction stage at the region level, we are concerned with extracting the appropriate set of features, or function $F^q(R_{ij})$, $q \in [1,2]$, to represent the contents of each unit R_{ij} in the image. To arrive at two independent $F^q(R_{ij})$ models, we could adopt the approach taken in Nigam and Ghani [10] to split the feature set into two disjoint sets. Here we adopt a natural split of features into: (a) Set 1: color histogram, and, (b) Set 2: texture and shape features. We can then used these two models to independently annotate the same regions using the appropriate G^l and F^q functions as expressed in Equation (1). By applying the bootstrapping framework, we can utilize the agreement between the collaborating Classifiers to confirm good labels. At the same time, we can use disagreement between classifiers to select good samples for users to annotate manually in order to maximize the learning objective in an active learning approach.

The following sections describe the details of our approach at the feature and segmentation stages.

4. CO-TRAINING FOR REGION CLASSIFIERS

Given a scenario that we have a (small) set of labeled regions $\underline{R}_z$, and a (large) set of unlabeled regions $\underline{R}_u$, we now discuss how to employ co-training and active learning framework at the region level. The regions are extracted using one of the segmentation method $S^p(I_i)$ adopted.

As discussed previously, we can develop two independent classifiers, H^1 and H^2, using SVM technique based on the set of labeled regions $\underline{R}_z$. The two Classifiers differ only on the set of features used during training. To ensure that they are statistically independent, we extract two independent set of disjoint features for region R_{ij}, denoted by $F^1(R_{ij})$ and $F^2(R_{ij})$. The corresponding classifiers, which map a region into a confident vector of concepts, are given by:

$$H^1\text{:}\ \ G\,(S^p(F^1(R^p{}_{ij}))\rightarrow \underline{\Phi}^1 \qquad (3)$$

$$H^2\text{:}\ \ G\,(S^p(F^2(R^p{}_{ij}))\rightarrow \underline{\Phi}^2$$

where $\underline{\Phi}^q = \{\ v^q{}_1, v^q{}_2, .., v^q{}_N\}$ with $q\in[4, 18]$. $v^q{}_j$ is the confident value for concept $c_j\in\underline{L}_c$, and N is the total number of concepts in $\underline{L}_c$.

Once the Classifiers are trained, given a new unlabeled region R_u, we can apply the classifiers to derive two set of concept confident vectors for R_u as: $H^1(R_u)\rightarrow\underline{\Phi}^1{}_u$ and $H^2(R_u)\rightarrow\underline{\Phi}^2{}_u$. By combining the outputs from H^1 and H^2, the final confidence vector for region R_u is:

$$\underline{\phi}_u = \frac{\sum_{j=1}^{n} v_j^1 * v_j^2}{\underline{\phi}_u^1 \bullet \underline{\phi}_u^2} \qquad (4)$$

The final confidence vector $\underline{\Phi}_u$ can be used to link and control the choice of concepts to region. Due to the unreliability of region segmentation method, a single concept may be inappropriate to describe the region's contents. The region, if given an inappropriate segmentation, even human often has problem assigning the precise concept label. With the use of confidence vector, we can circumvent this problem by choosing one or more concepts for each region depending on the strategies we choose. Two strategies that have been adopted in our framework are as follows:

Strategy 1: We pick pne semantic concept for each region. In this case, we choose only the concept with the highest confidence value. We formally describe the process as the following,

$$(c_i,\ v_i^q) = \underset{i}{\arg\max}\{v_i^q \mid v_i^q \in \underline{\phi}_u\} \qquad (5)$$

where $c_i \in \underline{L}_c$.

Strategy 2: We select one or more semantic concepts to annotate each region. Here we simply choose between 1 to k (k is set to 4 here) concepts whose confidence values are larger than a predefined threshold τ . Formally we have:

$$\{(c_i,\ v_i^q)\} = \begin{cases} \{c_i \mid v_i^q \geq \tau, v_i^q \in \underline{\phi}_u\}, \text{if } \exists v_i^q \geq \tau \\ \{c_i \mid \underset{i}{\arg\max}\{v_i^q \mid v_i^q \in \underline{\phi}_u\}\}, \text{if } \forall v_i^q < \tau \end{cases} \qquad (6)$$

We now outline the details of the co-training framework as follows.

- Inputs:

  An initial collection of (small) labeled regions $\underline{R}_z$;

  A large set of unlabeled regions $\underline{R}_u$;

  C_z: the class label of current classifiers;

  β: number of unlabelled regions to be considered in each round of co-training;

  θ : the predefined threshold for selecting the most confident class label;

  τ_1, τ_2 : the threshold for selecting one classifier to label over the other.

  ε : the tolerance for the least uncertainty regions

- Loop:

  While there exist regions without class labels or the rounds exceed the time limits:

  - Build classifiers H^1 and H^2 from the current labeled training set $\underline{R}_z$.
  - Pick up the next set of β unlabelled regions from $\underline{R}_u$:
    - For each unlabeled region, computing the confidence values for classifiers H1 and H2 using Equation (3).
    - The following conditions are used to choose and label the above set of unlabelled regions:

      Condition 1 (when both Classifier have high confidence): Choose those class labels whose confidence values are larger than θ in both H^1 and H^2. If they are the same as the class label C_z, then label the region and add it to the labeled set $\underline{R}_z$.

      Condition 2 (when only one Classifier has high confidence): If condition 1 is not satisfied, but the confidence value of the class label C_z for one classifier is larger than τ_1, while the that of the other classifier is less than τ_2, then simply use the classifier that gives higher confidence value to label the region and add the region to the labeled set $\underline{R}_z$.

      Condition 3 (both Classifiers are uncertain): If the above two conditions are not met, but the confidence values of two classifiers for the class label C_z are around $0.5 \pm \varepsilon$, then choose and ask the user/expert to label the region and add it to the labeled set $\underline{R}_z$.

      Otherwise, just drop and repeat.

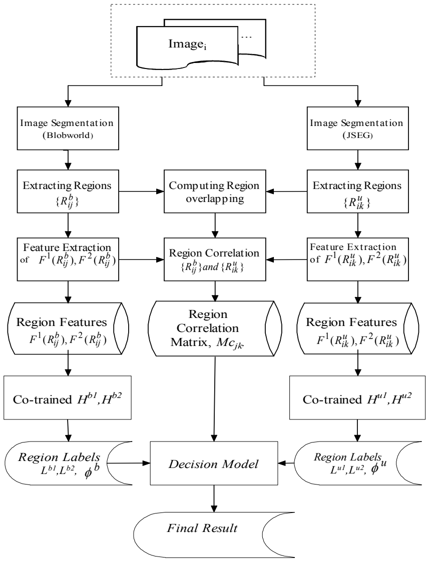
- Outputs: Two updated Classifiers H^1 and H^2 and an expanded labeled set $\underline{R}_z$.

The co-training and active learning procedures for the Classifiers as described above are performed for each segmentation method $S^p(I_i)$ separately.

5. CONCEPT DISAMBIGUATION AT THE IMAGE LEVEL

Section 4 describes the training of two independent Classifiers for each region generated by a segmentation method $S^p(I_i)$. However, because the region segmentation methods are unreliable and unstable in generating meaningful regions, we need to devise technique to ensure that the regions obtained and the annotations derived are reliable. To achieve this, we employ two segmentation methods and utilize the regions generated from two different methods to co-train the system. As the regions obtained by one segmentation method may correlate with several other regions generated from the other method, we make use of the redundancy of regions and independence of methods to improve the performance. The basic principle is that the regions and the derived semantic concepts do not exist independently, but are correlated. Thus, when evaluating the semantic concept for one region, we must consider the contextual effects, which can be inferred from the visual attributes and semantic concepts found in the overlapping regions.

To take the context of regions into consideration, we must first be able to evaluate the relationship between different overlapping regions, and from these derive appropriate features for use in a decision model to arbitrate ambiguous concepts. The overall process of our concept disambiguation process is given in Figure 1. Our approach is to some extend inspired by the framework for Chinese named entity extraction [22], in which multiple methods are used to extract Chinese named entities, and a decision tree is employed to disambiguate the conflicting named entities.



Image$_i$
...
Image Segmentation (Blobworld)
Image Segmentation (JSEG)
Extracting Regions $\{R_{ij}^b\}$
Computing Region overlapping
Extracting Regions $\{R_{ik}^u\}$
Feature Extraction of $F^1(R_{ij}^b), F^2(R_{ij}^b)$
Region Correlation $\{R_{ij}^b\}$ and $\{R_{ik}^u\}$
Feature Extraction of $F^1(R_{ik}^u), F^2(R_{ik}^u)$
Region Features $F^1(R_{ij}^b), F^2(R_{ij}^b)$
Region Correlation Matrix, Mc_{jk}
Region Features $F^1(R_{ik}^u), F^2(R_{ik}^u)$
Co-trained H^{b1}, H^{b2}
Co-trained H^{u1}, H^{u2}
Region Labels L^{b1}, L^{b2}, ϕ^b
Decision Model
Region Labels L^{u1}, L^{u2}, ϕ^u
Final Result

Figure 1 The concept disambiguation framework at the image level

For this research, we employ two segmentation methods based on Blobworld of UC Berkeley [11] and JSEG from UCSB [21]. They are denoted as $S^B(I_i)$ and $S^U(I_i)$ respectively. Applying these methods to an image I_i, we have:

$$S^B(I_i) \rightarrow \sum R^B_{ij}, j = 1, .. N^B \qquad (7)$$
$$S^U(I_i) \rightarrow \sum R^U_{ik}, k = 1, .. N^U$$

The two sets of regions for the same image I_i are linked via the correlation matrix: (Mc_{jk}), $j=1..N^B$, $k=1..\ N^U$. The correlation matrix encodes the overlaps between regions in the same image. We compute the overlap between every region R^B_{ij} and region R^U_{ik}, and normalizing the overlapping area by the size of image as:

$$Mc_{jk} = U_{R^B_{ij}, R^U_{ik}} = \frac{R^B_{ij} \cap R^U_{ik}}{|\mathrm{Im}age\ I_i|} \qquad (8)$$

From the two set of regions obtained from the two segmentation methods, we employ the Classifiers H^1 and H^2 developed in Section 4 to annotate the regions. That is, for each segmentation method, there will be two classifiers as follows:

$$H^{P1}:\ G\ (S^p(F^1(R^P_{ij})) \rightarrow \underline{L}^{P1} \qquad (9)$$
$$H^{P2}:\ G\ (S^p(F^2(R^P_{ij})) \rightarrow \underline{L}^{P2}$$

where $p \in [B,\ U]$, which stands for the Blobworld and JSEG segmentation methods respectively. The overall annotation of region R^P_{ij} is:

$$L(R^P_{ij}) = \underline{L}^{P1}\ U\ \underline{L}^{P2} \qquad (10)$$

As we expect the regions generated by different methods to be correlated, we expect the regions R^B_{ij} and R^U_{ik} to have some overlaps (i.e. $Mc_{jk} \neq 0$ for some j and k), and they share common concepts (i.e. $L(R^B_{ij}) \cap L(R^U_{ik}) \neq 0$ for some j and k). The disambiguation process will make use of a decision model to identify the best regions and labels based on this contextual information. However, because of the asymmetric structure of region correlation, we use the idea of master and slave regions to differentiate the processing order. That is, we use R^B_{ij} as the master and R^U_{ik} as the slave, and vice versa, to perform the disambiguation using a decision model. For each master region, we derive its concept vector as $\underline{L}^M$. Through the correlation matrix Mc_{jk}, we find the overlapping slave regions and their corresponding concept vectors $\underline{L}^S$'s.

We employ the decision tree SEE5 to evaluate the confidence level of the concepts derived for the overlapping regions. The inputs to the decision tree are the master region, its concept vector $\underline{L}^M$, and the list of slave regions and their corresponding concept vector $\underline{L}^S$'s. The output of the decision tree is a confidence vector for the master region, $\underline{\Phi}^M$, where the elements of $\underline{\Phi}^M$ are as defined in Equation (3).

From $\underline{\Phi}^M$, it is easy to choose the concept for the region. We again employ the same strategies as in Section 3 (Equations 5-6) to select one or more concepts to annotate the Master regions.

The same process is repeated by using R^U_{ik} as the master and R^B_{ij} as the slave. The unions of all resulting concepts are used as final annotation of the image.

6. EXPERIMENTAL RESULTS AND DISCUSSIONS

6.1 Test Data and Methods

To test the effectiveness of our approach, we use an image collection comprising about 6,000 images. The images come from PhotoCD, web and parts from CorelCD. We randomly selected as sub-set of images for training, and the rest for testing. For the bootstrapping experiment, we choose only 20 labeled seed regions for each concept label to kick-start the bootstrapping process. More details of the training and testing setup will be discussed later when we describe the methods.

For the bootstrapping framework described earlier, we need to select different models at different stages of the process. The models we used are:

a) Feature selection function $F^q(R^p_{ij})$. For each region R^p_{ij}, we use the standard color histogram, texture and shape as the features. For the co-training experiments, we divide the feature set as: F^1 contains the color histogram, and F^2 includes only the texture and shape features.

b) Segmentation methods $S^p(I_i)$. We employ two segmentation methods based on Blobworld of UC Berkeley [11] ($S^B(I_i)$) and JSEG from UCSB [21] ($S^U(I_i)$).

c) Image annotation function $G^l(I_i)$. Here we use SVM to train the Classifiers, and Decision Tree to disambiguate the concepts learned from different Classifiers based on different segmentation methods. For our research, we experiment with using two types of SVM -- the soft-margin SVM (also called the probabilistic SVM, or PSVM) that returns multiple decisions with confidence values, and hard SVM, that returns only a single answer. We select SVM with radial basis function (RBF) kernel [23], and using logistic regression for computing the probability of SVM [24].

In order to test the effectiveness of our bootstrapping method against traditional machine learning methods, we carried out experiments using the following methods:

a) Traditional Machine learning Approaches based on Soft-SVM. Here we combine the feature sets F^1 and F^2 into one set. For test data, we use 400 labeled images for training and remaining 5,600 for testing. We experiment with three variants of method:

   Method 1: Use both the Blobworld and Jseg segmentation methods separately and simply integrate the results of region annotation at image level. No disambiguation step is performed here.

   Method 2: Employ two segmentation methods and use Decision Tree to perform concept disambiguation. It uses strategy 1 to select only one concept for each region (Equation 5).

   Method 3: Same as Method 2 except that it selects multiple concepts for each region (Equation 6)

b) Traditional Machine learning Approaches based on Hard-SVM. The two variants we experimented are:

   Method 4: Use both Blobworld and Jseg segmentation methods, as in **Method 1**.

   Method 5: Same as in **Method 2**.

c) The Bootstrapping Framework. We tested our framework as **Method 6**.

6.2 Initial Experiment Results

Table 1 shows our initial results for the 5 methods. The results are presented in terms of recall, precision and F_1 measures. In addition, we also differentiate between two kinds of results. The first set, which we termed “automatically checked Result (ACR)”, compares the learned concepts for each image against the ground truth (i.e those provided by the original image authors). It did not consider whether the additional concepts learned by the system that are not present in the ground truth are correct. In general, we found that most images are assigned only one or few keywords, and these keywords often do not explain the image details and are thus incomplete. Thus the ground truth often misses some details of images that were found by the automated methods. As a result, ACR tends to report lower precision for our methods, as we tend to find more concepts that are correct. In order to fairly evaluate the automated techniques, we present another set of results, which we termed “manually checked Result (MCR)”. In MCR, we manually checked the learned concepts against the ground truth or the image contents. We consider the learned keywords as correct if it presents in the ground truth as well as in the image contents. MCR allows us to add more meaningful keywords into the ground truth. For example, the image with only the keywords plane often has sky, cloud, etc. The cloud and sky are likely to be learned in the automated approach, which should be considered as correct.

From Table 1, we found that methods (**Methods 1 and 4**) that did not make use of the decision making disambiguation step generally have low performance. They could achieve about an average of 20%-27% for ACR and 28%-34% for MCR in F_1 measures. When we use two segmentation methods to overcome the basic problem of region segmentation, and incorporate decision model in concept disambiguation using contextual information (**Methods 2 and 5**), we found considerable improvement in performance of the methods. In particular, we found improvement in the F_1 measures of more than 11% for ACR and 8.7% for MCR, as compare to **Methods 1 and 4**. In addition, if we also adopt the strategy of “one region, one or more concepts” (**Method 3**), we could further improve the performance, attaining the F_1 measures of 35% for ACR and 47% for MCR.

By employing our bootstrapping method (**Method 6**), we could achieve the highest F_1 measure of over 52% for MCR case. This is an improvement of over 10% as compared to the best of traditional methods tested (**Method 5**). Although the results of **Method 6** is not satisfactory for ACR measure, we attribute this to the fact that ACR measure is flaw and inappropriate as the ground truth is incomplete. The results are encouraging as the bootstrapping process is not only effective, but it also requires very few labeled examples as compared to the other methods (**Methods 1-5**) that are based on the traditional machine learning approach.

Table 1 **Initial results on for image annotation experiments**

Approaches	**Mth**	**Automatically Checked Results (ACR)**			**Manually Checked Results (MCR)**			**Comments**
		Rec.	**Pre.**	**F1**	**Rec.**	**Pre.**	**F1**	
Soft Binary SVM (probabilistic binary SVM Classifier)	Mth 1	21.05	19.47	20.23	30.66	27.31	28.88	Region only
	Mth 2	45.00	26.49	33.35	50.45	39.03	44.02	With DT
	Mth 3	47.58	28.52	**35.35**	51.10	44.17	**47.38**	With DT
Hard SVM Classifier	Mth 4	25.68	29.67	27.53	32.18	36.9	34.38	Region only
	Mth 5	50.28	31.55	38.77	50.55	37.57	43.10	With DT
Bootstrapping	Mth 6	36.94	27.20	**31.33**	58.59	47.54	**52.49**	With DT

6.3 Examples of Image Annotation

Figure 2 gives some examples of images annotated using our approach. Column 2 of Figure 2 shows both the original annotation provided by the authors, as well as the annotation learned by our system. The results show that our annotation scheme could give reasonably accurate and complete annotation. Note that as we support only “animal” as the general concept for all types of animals, specific animals such as “dog”, “tiger” etc. are tagged as “animals”, which are considered to be correct.

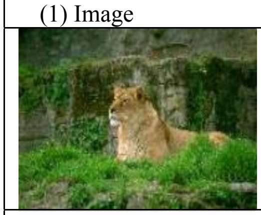
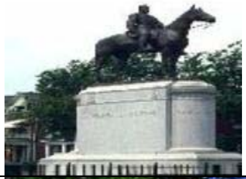
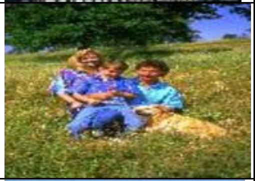
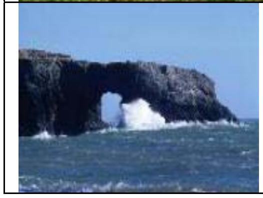
(1) Image	(2) Keywords
	Original: tiger, grass, rock Learned: animals, grass
	Original: travel Learned: plant, travel, rock
	Original: people, plant, travel, animals Learned: people, travel, grass, sky
	Original: water Learned: travel, rock, water

Figure 2 Examples of image annotation using our approach

7. SUMMARY

As many large image/video collections become available, the effective access to such information becomes a major problem. Because of the ambiguity in content-based retrieval techniques, most users prefer to access such information via keywords. This brings in a major practical problem of how to (semi-) automatically annotate large image/video archives with text annotations. Several recent works attempted to tackle this problem by adopting supervised learning approaches to associate visual information extracted in segmented images with semantic concepts provided by associated text. The main limitation of such approaches, however, is that large labeled training corpus is still needed for effective learning, and semantically meaningful segmentation for images is in general unavailable. This research explores the use of bootstrapping approach to tackle this problem. The idea is to start from only a small set of labeled training examples, and successively annotate a larger set of unlabeled examples. This is done using the co-training approach, in which two “statistically independent” classifiers are used to co-train and co-annotate the unlabeled examples. In addition, we incorporate a decision model to disambiguate the concepts learned from regions extracted from different segmentation methods. We carried out experiments using a mid-sized image collection, comprising about 6,000 images from CorelCD, PhotoCD and Web. Our results demonstrated that our bootstrapping approach could significantly out-perform the traditional supervised learning approach in image annotation. It also has the added advantage that it requires very few labeled examples as compared to the traditional methods.

We will continue to improve our bootstrapping framework with active learning. Our results demonstrate that the collaborative bootstrapping approach, initially developed for text processing, can be employed effectively to tackle the challenging problems of multimedia information retrieval. We will exploit this approach in the semantic video indexing and retrieval, where collaborative bootstrapping approach will integrate the video and audio features into the framework. We will also explore the web image mining based on the images obtained from the web and their surrounding context.

ACKNOWLEDGEMENT

The first author would like to thank National University of Singapore (NUS) for providing scholarship. The authors also would like to thank to Dr.Chih-Jen Lin for sharing with his LIBSVM library.

References:

[1] John R.Smith and S-F Chang, *VisualSeek: a fully automated content-based query system*. In Proc Fourth int conf multimedia, ACM 87-92 (1996).

[2] John R.Smilth, milind Naphade and Apostol (Paul) Natsev, *Multimedia semantic indexing using model vectors*. ICME (2003).

[3] Steven Abney, *Bootstrapping*. 40th Annual Meeting of the Association for Computational Linguistics (2002).

[4] A.Blum and T.Mitchell, *Combined Labeled Data and Unlabelled Data with Co-training.* In Proceeding of the 11th Annual Conference on Computational Learning Theory (1998).

[5] David A.Cohn, Zoubin Ghahramani and Michael I.Jordan, *Active learning with statistical models*. Journal of Artificial Intelligience Reseach *4,* 129-145 (1996).

[6] Y.Mori, H.Takahashi and R.Oka, *Image-to-word transformation based on dividing and vector quantizing images with words.* First International Workshop on multimedia Intelligent Storage and Retrieval Management (1999).

[7] K.Barnard and D.A.Forsyth, *Learning the semantics of words and pictures*. IEEE International Conference on Computer Vision *II,* 408-415 (2001).

[8] Edward Chang, Kingshy Goh, Gerard Sychay and Gang 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,* 26-38 (2003).

[9] K.Barnard, P.Duygulu and D.Forsyth, *Clustering art.* In Proc of IEEE Computer Vision and Pattern Recognition 434-441 (2001).

[10] S.Belongie, C.Carson, H.Greenspan and J.Malik, *Recognition of images in large databases using a learning framework*. Technical report 07-939, UC Berkeley CS Tech Report *07-939,* (1997).

[11] C.Carson, M.Thomas, S. B. , J.M.Hellerstein and J.Malik, *BlobWorld: A System for region-based image indexing and Retrieval.* Int Conf Visual Inf Sys (1999).

[12] Edward Chang, Kingshy Goh, Gerard Sychay and Gang 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,* 26-38 (2003).

[13] R.Herbrick, T.Graepel and C.Campbell, *Bayes Point Machines*. Journal of Machine Learning Research *1,* 245-279 (2001).

[14] James Z.Wang and Jia Li, *Learning-based Linguistic Indexing of Pictures with 2-D MHHMs*. The 10th ACM Int Conference on Multimedia 436-445 (2002).

[15] M.Collins and Y.Singer, *Unsupervised Models for Name Entity Classification.* In Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural language Processing and Very Large Corpora (1999).

[16] I.Muslea, S.Minton and C.A.Knoblock, *Selective sampling with co-testing*. in CRM workshop on Combining and Selecting Multiple Models with Machine Learning (2000).

[17] K.Nigam and R.Ghani, *Analyzing the Effectiveness and Applicability of Co-training.* In Proceedings of the 9th International Coference on Information and Knowledge Management (2000).

[18] Y.Cao, H.Li and L.Lian, *Uncertainty reduction in collaborative bootstapping:measure and algorithm*. In proceeding of the 41th Annual Meeting of the Association for computational Linguistics (2003).

[19] D.D Lewis and W.A.Gale, *A sequential algorithm for training text classifiers*. In proceeding of ACM SIGIR 3-12 (1994).

[20] Cha Zhang and Tsuhan Chen, *An active learning framework for content-based information retrieval*. IEEE transactions on multimedia *4,* 260-268 (2002).

[21] Y.Deng and B.S.Manjunath, *Unsupervised segmentation of color-texture regions in images and video.* IEEE Trans on Pattern Analysis and Machine Intelligence *23,* 800-810 (2001).

[22] Tat_Seng Chua and Jimin Liu, *Learning pattern rules for Chinese named-entity extraction.* AAAI'2002 411-418 (2002).

[23] Vladimir Vapnik, The nature of Statistical Learning Theory, Springer, New York 1995.

[24] John C.Platt, *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods*. In Advances in Large Margin Classifiers, Alexander J Smola, Peter Bartlett, Bernhard Scholkopf, Dale Schuurmans, eds MIT Press (1999).