A Bootstrapping Framework for Annotating and Retrieving WWW Images

Huamin Feng[1,2], Rui Shi[1] and Tat-Seng Chua[1]
[1] School of Computing, National University of Singapore, Singapore 117543
[2] Beijing Electronic Science & Technology Institute, 100070, China
{fenghm, shirui, chuats}@comp.nus.edu.sg

ABSTRACT

Most current image retrieval systems and commercial search engines use mainly text annotations to index and retrieve WWW images. This research explores the use of machine learning approaches to automatically annotate WWW images based on a predefined list of concepts by fusing evidences from image contents and their associated HTML text. One major practical limitation of employing supervised machine learning approaches is that for effective learning, a large set of labeled training samples is needed. This is tedious and severely impedes the practical development of effective search techniques for WWW images, which are dynamic and fast-changing. As web-based images possess both intrinsic visual contents and text annotations, they provide a strong basis to bootstrap the learning process by adopting a co-training approach involving classifiers based on two orthogonal set of features – visual and text. The idea of co-training is to start from a small set of labeled training samples, and successively annotate a larger set of unlabeled samples using the two orthogonal classifiers. We carry out experiments using a set of over 5,000 images acquired from the Web. We explore the use of different combinations of HTML text and visual representations. We find that our bootstrapping approach can achieve a performance comparable to that of the supervised learning approach with an F_1 measure of over 54%. At the same time, it offers the added advantage of requiring only a small initial set of training samples.

Categories and Subject Descriptors

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

General Terms

Algorithms, Performance, Design, Experimentation

Keywords

Bootstrapping, Co-training, Image annotation, WWW images

MM'04, OCTOBER 10–16, 2004, NEW YORK, NEW YORK, USA.
Copyright 2004 ACM 1-58113-000-0/00/0000…$5.00.

1. INTRODUCTION

With the explosive growth of information on the World-Wide Web (WWW), the need for effective tools to index and retrieve information has become ever more important. This is especially so for multimedia information such as images and videos. Current search engines are designed to handle text documents effectively, but they tend to treat multimedia contents as second-class objects. As a result, even though popular search engines such as Google™ and AltaVista™ offer a search function for images, the search is based only on textual evidences, which are taken from an image's filename, ALT-tag and/or associated web pages. This approach is similar to many recent ones for retrieving images from the Web [22, 23]. Although such approaches offer reasonable recall, they tend to be poor in precision as they are unable to confirm whether the retrieved images actually contain the desired concepts expressed in the queries.

Away from the Web, there are numerous attempts to use image contents as the basis to index and retrieve images [10, 26]. The current content-based approaches rely on color, texture and/or statistical shape features to model image contents. As these features are low-level, they are effective only in matching images almost identical in contents, but fail miserably when similarity or object-based matching is desired. As such, content-based techniques tend to be effective locally, when a good subset of images have been retrieved through other means based on, say, textual evidence or other classification schemes. In a way, content-based retrieval techniques can be thought of as a local precision enhancement device. This is the observation of the recent experiments on large scale video retrieval experiments [7, 10].

To address the problem of effectively indexing and retrieving a large amount of images from the Web, we need to tackle three challenges.

First, we need to explore techniques to extract appropriate textual hints from the associated HTML pages of images. Many research efforts focus on exploiting a range of HTML document content structures, including: (a) image title, ALT-tag, and some form of surrounding text [23, 32]; and (b) link structure and anchor text [22]. The resulting systems have been found to be effective on a subset of images extracted from the Web. In order to take image contents into consideration to improve precision, recent techniques also consider the modeling of visual contents in indexing and retrieving WWW images [32].

Second, we need to develop better representation to model image contents. Traditional content-based image retrieval approaches employ color, texture and/or statistical shape features to model image contents [10, 26]. It is well known that the use of such low-level content features is inadequate. As a result, the recall and precision of content-based image retrieval techniques are generally low. Moreover, retrieval effectiveness is highly dependent on the choice of query images and the diversity of relevant images. In order to ensure high retrieval effectiveness, special purpose systems tend to rely on specific features depending on the application, such as the use of a face detector and a face recognizer [1] to look for human images on the Web. As fixed content representation is unlikely to meet the needs of a range of applications, the challenge here is to explore an adaptive content representation scheme that can be applied to a wide range of image classification tasks.

Third, we need to explore techniques to fuse evidences from text and visual contents, and to ensure that such techniques scale up to the large amount of images on the Web. Most approaches explore the fusion of multi-source evidences by employing either heuristic techniques such as convex combination or voting scheme [7, 10], or the Dempster-Shafer combination technique [1]. Recent techniques classify images into one or more pre-defined categories by employing a learning-based approach to associate the visual information extracted from images with the semantic concepts provided by the associated text [2, 6, 30]. The main limitation of such learning-based approaches is that for effective learning, a large labeled training corpus is needed, which is hard to come by. [9] tackled this problem by exploiting a bootstrapping approach. The challenge here is to develop a bootstrapping learning scheme for WWW images that fuses evidences from visual contents and their associated HTML text, and which requires only a small number of labeled training samples to kick-start the learning process.

This paper describes a bootstrapping framework to automatically annotate WWW images using a pre-defined list of concepts. The annotated concepts can then be used as the basis to support keyword-based image retrieval. Bootstrapping aims to use a small set of labeled samples to kick-start the learning process with a large unlabeled corpus. To achieve bootstrapping, we need a way for the system to evaluate the quality of newly annotated samples. This can be achieved by using the co-training technique [4] in which two "view-independent" methods independently confirm the quality of newly annotated samples, and learn from each other's results. To accomplish this, we exploit the evidences from both the HTML text and visual content features of an image by developing two "orthogonal" classifiers – one based on text, and the other on visual content features. The classifiers are developed using probabilistic Support Vector Machine (pSVM) [19]. In addition, we also explore the use of different representations to model HTML text and the visual contents of an image. We carry out experiments using a set of over 5,000 images acquired from the Web. The results demonstrate that the framework is effective and that it is able to achieve a level of effectiveness similar to that of traditional supervised learning approaches but requiring a much smaller set (<23%) of labeled training samples.

The main contribution of this research is two-fold. First, we develop a co-training framework to bootstrap the process of annotating large WWW image collections by exploiting both the visual contents and text annotations of the images. Second, we demonstrate that the framework could achieve a level of performance similar to or better than the traditional machine learning approach but requiring a much smaller set of labeled training samples. The framework can be used to develop deployable system for annotating the large amount of dynamic WWW images.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 discusses the co-training framework, describes the design of our text and visual classifiers, and details our adaptive matching pursuit feature scheme. Section 4 presents our evaluations and Section 5 concludes the paper.

2. RELATED WORK

Our work is related to research in three areas: web-based image retrieval, image annotation and co-training. The retrieval of images from the Web has received a lot of attention recently as a natural step to extending the mature text-based information retrieval techniques to images. Most early systems employ an essentially text-based approach that exploits how images are structured in Web documents. Sanderson and Dunlop [22] were among the first to model image contents using a combination of texts from associated HTML pages and pages that are one to several links away. They modeled the contents as a bag of words without any structure. Shen et al. [23] considered the use of more concise information on the associated web page such as the image file name, page tile, ALT-tag and image caption, and built a simple model to chain related terms. More recently, Yanai [32] used a combination of the above text features as well as link information and surrounding text to model an image. In addition, he incorporated visual features such as color and region signatures to model visual contents. He employed the EMD (earth mover distance) algorithm to compute the similarity between text and visual features, and reported an F_1 of about 34% for the retrieval of web-based images. The retrieval effectiveness of Yanai's approach is low as the visual content representation is too low-level and inadequate to model an image's contents. To address this problem, Leinhart and Hartmann [12] employed domain specific features to classify images on the Web into the categories of photo, non-photo, computer rendered realistic images, presentation slides and scientific posters, etc. The features used are tuned more to this domain including color regions, number of prevalent colors, sharpness of edges, median and Guassian feature values, etc. They reported an accuracy of over 97% in the above classification tasks. They, however, did not use any text feature. Aslandogan and Yu [1] developed a specialized people finder for the Web that combines evidences extracted from HTML text, a face detector and a face recognizer using the Dempster-Shafer algorithm. They reported a high retrieval accuracy of over 94%.

Differing from image retrieval, image annotation deals with the automated or semi-automated attachment of a pre-defined set of keywords [2, 6, 30] to images. It has received much attention recently with the desire to automatically annotate large amounts of images to facilitate keyword-based image retrieval. Starting from a training image set with text annotations, these approaches employ statistical learning models to associate visual features of sub-units within images with text annotations. The sub-unit employed could be the whole image, a fixed size block or a region. Almost all approaches use traditional color, texture and statistical shape as visual features. Mori et al. [15] were among the earliest to perform "image-to-word" transformation. They divided images into fixed-size blocks, and trained the clusters of blocks to predict keywords for new images using a co-occurrence model. Barnard and Forsyth [2] segmented images into regions using Blobworld segmentation [5], and employed a translation model to associate keywords to regions in the training set. Wang and Li [30] performed image analysis on fixed size blocks using a 2-D multi-resolution HMM to capture cross block and cross resolution dependencies for an entire image collection. Chang et al. [6] considered the problem at the image level, and experimented with using BPM (Bayes Point Machine) and SVM to link visual features with their associated keywords. They showed that BMP provides superior performance to SVM. Jeon et al. [11] also used Blobworld to segment images into regions, and learned the joint distribution of blob regions and words. They tested their system on over 5,000 images and reported an F_1 of about 44%.

The above approaches are based on the traditional supervised learning scheme. The training set is fixed and much manual annotation work is needed to arrive at a reasonable size labeled set. To overcome this problem, Blum and Mitchell [4] proposed a co-training algorithm based on the conditional (view) independence assumption. The algorithm repeatedly trains two classifiers from labeled data, labels some unlabelled data with the two classifiers, and swaps newly labeled data between the two classifiers. In this 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 [8] proposed a co-training algorithm based on "agreement" between the classifiers. Nigam and Ghani [16] empirically demonstrated that even bootstrapping (or co-training) that violates the view-independent assumption can still work better than the traditional learning approach. Pierce and Cardie [18] proposed a moderately supervised variant of co-training in which a human corrects the mistakes made in certain automatic labelings. The above co-training approaches have been applied to text processing areas, such as web documents classification and information extraction. Feng and Chua [9] employed the co-testing approach to bootstrap the process of image annotation by employing two "weakly independent" classifiers based on disjoint subsets of visual features. They performed annotation on the set of regions generated by two different segmentation methods and incorporated a contextual model to disambiguate the concepts learned by considering the relationships between concepts and overlapping regions.

3. CO-TRAINING FRAMEWORK FOR ANNOTATING WWW IMAGES

Image annotation has been referred to in recent literature as the process of associating keywords or concepts[1] with new images based on their visual contents. It can also be viewed as a classification problem aimed at determining if the contents of the whole or part of an image could be classified into one of N categories or concepts. The set of concepts depicts the major semantic contents or objects within the image collection. Once annotated, images can be retrieved by issuing concept (or keyword)-based queries. Although a web-based image comes with "intrinsic" text annotations from its associated HTML page, such annotations tend to be noisy and incomplete. Thus this research aims to explore the use of a bootstrapping framework to annotate WWW images using a pre-defined set of concepts accurately and completely through the support of both textual and visual evidences.

The idea of bootstrapping is to start from a small set of training samples, and successively acquire more training samples at each bootstrapping iteration to derive better classifiers. To achieve this, we need a way for the system to evaluate the quality of newly annotated samples. This can be achieved by using the co-training technique [4] in which two "orthogonal" classifiers independently confirm the quality of newly annotated samples, and learn from each other's results. Web-based images offer a natural way to accomplish this, because they come with intrinsic visual contents as well as text annotations derived from their associated HTML pages and links.

The following sub-sections discuss the development of the text-based classifier, the visual-based classifier, and the resulting co-training framework.

3.1 Text-based Classifier

The text descriptions of a web page often give useful hints on what an embedded image is about. However, while textual contents may contain information that captures the semantics of the embedded image, it also contains other descriptions that are not relevant to the image. These "noises" lead to poor retrieval performance.

To improve retrieval performance, we need to extract textual contents that are relevant to an image while avoiding irrelevant information. There are several places where relevant text may be found, namely, (i) image file name; (ii) page title; (iii) alternate text (ALT-tag); and (iv) surrounding text. Most current approaches employ the first three features [23, 32] as they are easy to extract, and tend to provide the most accurate description of the embedded image. However, our empirical studies show that they often do not give sufficient information on an image. The image file name is often abbreviated and may not be recognized as meaningful words. The page title may be too general for the embedded image as there may be more than one image or topic in a web page. Moreover, a large number of images do not even have alternate text. In fact, we found that only 21% of images that we have downloaded possess alternate text. Thus, while the first three features provide reasonably accurate description of an image, they are often inadequate, leading to poor recall.

In order to provide a more complete description of image contents, we need to incorporate the use of relevant surrounding text. However, the great variety in style and web page layout makes the extraction of surrounding text a challenging task. This is partly why most existing approaches do not consider surrounding text. Fortunately, there is regularity to the appearance of relevant surrounding text with respect to the position of an image in an HTML document. For example, relevant surrounding text often appears adjacent to or below an image, or in the table cell next to the one containing the image. Our study of over 1,000 web pages arrives at the following observations: (a) Relevant surrounding text can be identified by extracting the nearby text tagged by structural tags of appropriate types that we call separators. The list of useful separators is: {<br>, <hr>, <p>, <table>, <tbody>, <td>, <th> and <tr>}. (b) Surrounding text may appear to the left or right of the image in the HTML document. The probability of finding relevant surrounding text to the right is 73% while that to the left is 27%. (c) When there are

[1] Throughout this paper, we liberally use the terms "concept" and "keyword" interchangeably.

more sequences of surrounding text found around an image, we use all the surrounding text found as descriptor to the image. (d) According to the survey conducted by Google, the first or last 32 words in the text nearest to an image appear to be most descriptive of the image. So if the text description extracted in the left or right direction is longer than 32 words, we only keep the first 32 words as surrounding text. Further details of our algorithm for finding relevant surrounding text can be found in [17].

Given the set of text descriptors derived from the associated HTML pages, we perform standard stop word removal and stemming using Porter's algorithm, and assign tf.idf weights to the remaining terms [21].

In order to evaluate the quality of text descriptors extracted in modeling image contents, we explore two textual representations for web images:

- The first representation, ***T1***, models the contents of an image using image file name, page title, and ALT-tag. This is the representation adopted in most existing systems. It emphasizes precision.
- The second representation, ***T2***, uses ***T1*** plus surrounding text to model the textual contents of an image. We expect ***T2*** representation to have higher recall than ***T1***, and may lead to higher performance since we can use image visual features to refine classifications.

For each textual representation, we train a set of SVM classifiers, one for each concept, by using the set of labeled training samples provided. The classifiers trained are then used to assign one or more pre-defined concepts to the image. In this research, we employ the soft-margin SVM [19] (also called the probabilistic SVM) that returns a probability value for each concept. This is in contrast to hard SVM that returns a binary value. In our earlier studies on image annotation tasks [9], probabilistic SVM has been found to be more effective than hard SVM. We select SVM with radial basis function (RBF) kernel [29], and use logistic regression to compute the probability of SVM [19]. The resulting classifier H^{Ti} is given by:

$$H^{Ti}:\ G^{text}\,(T_i)) \rightarrow \underline{\Phi}^{Ti} \qquad (1)$$

where $T_i \in \{\boldsymbol{T1}, \boldsymbol{T2}\}$, and $G^{text}(..)$ defines the mapping from text representation T_i of image i into concept space $\underline{L}_c$. $\underline{L}_c$ contains the set of Lexicon or N pre-defined concepts to be used to annotate the images. $\underline{\Phi}^{Ti} = \{\, v^{Ti}_1, v^{Ti}_2, .., v^{Ti}_N\}$, with v^{Ti}_k gives the confident value for concept $c_k \in \underline{L}_c$.

3.2 Visual-based Classifier

As discussed in Section 2, almost all existing systems use a combination of color histogram, texture and statistical shape features to model the visual contents of images [10, 26]. These visual features have been found to be too low-level to adequately model image contents. As a result, they are effective only in matching highly identical images, and will fail if there is diversity among relevant images, or when the query is looking for object segments in the images. These problems point to the need to develop an adaptive feature representation scheme, where the representation could adapt to the characteristics of the images in the category. Here, we explore the development of adaptive features for texture, to be used in conjunction with color histogram. We do not use shape feature as it is often unreliable and is easily affected by noise.

Signal processing features, such as DCT, wavelets and Gabor filters, have been widely used for texture analysis in many image retrieval systems [14, 24, 30, 31]. The main advantage of signal processing features is that they can characterize the local properties of an image very well in different frequency bands. However, an image usually contains many different local properties that need to be characterized individually. In order to facilitate adaptive image representation, we borrow the concept from matching pursuits [3, 13], and employ a combination of DCT and three wavelets as the basis functions to construct an over-complete dictionary in our system. The three wavelets chosen are Haar, Daubechies and Battle/Lemarie. The reason that DCT and these three wavelets are chosen is that they have different abilities to model the details of images with different local properties. For example, images with sharp edges such as modern buildings are better modeled using the Haar wavelet; signals with a sharp spike are better analyzed by Daubechies' wavelets; whereas images with soft edges such as clouds are better modeled using DCT transform. Thus given a band of basis functions for DCT and different wavelets, we should be able to find a representation that best matches a given image. We do not use Gabor filters because they are non-orthogonal and expensive to construct.

The basic ideas of wavelet and DCT transforms are similar. In DCT, a signal is decomposed into a number of cosines of different frequency bands; whereas in wavelet transform, a signal is decomposed into a number of chosen basis functions. To extract matching pursuit features, we divide an image into fixed-size blocks of 8x8 pixels. For each corresponding block for DCT and the three wavelets, we derive a total of 252 basis functions, which are then partitioned into 16 frequency bands. Through the application of matching pursuit algorithm over the whole image, we derive 16 mean and 16 variance energy values for all the blocks as the adaptive matching pursuit features. We combine these 32 adaptive matching pursuit features with the 53-bin luv color histogram to model the visual contents of the image. The details on how the adaptive matching pursuit features are extracted are given in [27].

Comparing with conventional wavelet-based texture features, the main advantages of adaptive texture features are that they are efficient and provide an accurate reflection of local texture properties. This is because through matching pursuit, we are able to obtain the most appropriate representation for an image with fewest significant coefficients.

As in text, we explore two visual representations of images to compare our adaptive matching pursuit features with traditional visual content features:

- The first representation, ***V1***, is based on the traditional combination of color histogram, DCT texture and statistical shape features.
- The second representation, ***V2***, is based on the combination of color histogram and adaptive matching pursuit features for texture.

For each visual representation, we again train a set of probabilistic SVM classifiers by using the set of labeled training samples. The classifiers trained are used to assign one or more pre-defined concepts to the image. The resulting classifier H^{Vi} is given by:

$$H^{Vj}:\ G^{visual}\,(V_j)) \rightarrow \underline{\Phi}^{Vj} \qquad (2)$$

where $V_j \in \{\boldsymbol{V1}, \boldsymbol{V2}\}$, $G^{visual}(..)$ defines the mapping from visual representation V_i of image j into concept space $\underline{L}_c$, and the rest of the variables are as defined in Equation (1).

3.3 The Co-Training Framework

Given two view-independent classifiers, one based on text and the other on visual contents, our co-training framework for annotating web-based images proceeds as follows.

- Inputs:

  $\underline{R}_L$: an initial collection of (small) labeled images;

  $\underline{R}_U$: a large set of unlabeled images;

  $\underline{L}_c$: the concept labels of current classifiers;

  β: the number of unlabelled images to be considered in each iteration of co-training;

  M: the maximum number of iterations of the co-training process;

  m: the iteration number (m=0 initially);

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

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

- LOOP:

  While there exist images without concept labels and m <= M:

  - Train classifiers H^{Ti} and H^{Vj} using the current labeled training set $\underline{R}_L$.
  - Randomly select β unlabelled images, $\underline{R}_\beta$, from $\underline{R}_U$:
    - For $I_i \in \underline{R}_\beta$, compute the confidence values for all concepts in I_i using classifiers H^{Ti} and H^{Vj}.
    - Assign all concepts $c_j \in \underline{L}_c$ to image I_i based on the following conditions:

      Condition 1 (when both classifiers have high confidence in c_j): When the confidence value for concept c_j is larger than θ for both classifiers.

      Condition 2 (when only one classifier has high confidence in c_j): When condition 1 is not satisfied, but the confidence value for concept c_j is greater than τ_1 for one classifier and less than τ_2 for the other classifier.

      Add all $I_i \in \underline{R}_\beta$ with assigned labels to the labeled set $\underline{R}_L$.
    - Optionally, when both classifiers are uncertain, we can include an *active learning* [33] step to select k images with least entropy values and ask users to annotate. This step is not used in our experiments.

- Outputs: Two updated Classifiers H^{Ti} and H^{Vj} and an expanded labeled set $\underline{R}_L$.

4. EXPERIMENTS AND RESULTS

4.1 Test Data

We use a set of 15 concept labels listed in Table 1 to test the effectiveness of our bootstrapping approach to annotate WWW images. The concepts are chosen based on the following three criteria: (a) the concepts are non-abstract with distinct visual forms; (b) they represent objects frequently found in WWW images; and (c) we are able to gather a sufficient number of images for training and testing. Criterion (a) is imposed to ensure that visual features can be used effectively to help annotate the images. For more general concepts that do not have distinct visual forms, it is not reasonable to expect visual features to be useful. For general concepts such as travel, industry and transport which are typically found in text annotations in Coral CD, we need to develop other techniques to induce them based on the presence of lower-level concepts. We will extend our research to annotate general and hierarchical concepts in our next phase of work.

Table 1: List of concepts used to annotate WWW images

tiger, lion, dog, cat, bear, financial-chart, building, plane, beach, mountain, waterfall, sunset, flower, ski, river

We use the Google image search function to gather the images. Altogether, 5,418 images with associated web pages are downloaded and used in our evaluations. We ensure that there are at least 150 images for each concept to provide sufficient data for training and testing. For each image, we download the image itself, as well as the text descriptors from their embedding HTML page as outlined in Section 3.1. We randomly select 60% of images for training, and the rest for testing. The number of training images for each concept ranges from 97 to over 350, with an average of 216 training images per concept class. In selecting the training images, we ensure that the ratio of training and testing images for each concept class follows strictly the original distribution.

4.2 Experimental Setup

The aim of the evaluations is three-fold. First, we want to demonstrate the effectiveness of our co-training framework in integrating HTML text and visual features. Second, we want to verify that we are able to achieve similar or better performance as compared to the traditional supervised learning approach by using a smaller number of labeled training samples. Third, we want to show that our adaptive matching pursuit texture features are more effective than the traditional visual features in modeling the visual contents of images.

To achieve these aims, we design tests by varying content feature representations and test parameters as follows.

A. We test the systems based on different combinations of text and visual features:

- Feature Configuration 1: combination of ***T1*** and ***V1***
- Feature Configuration 2: combination of ***T2*** and ***V1***
- Feature Configuration 3: combination of ***T1*** and ***V2***
- Feature Configuration 4: combination of ***T2*** and ***V2***

where ***T1, T2, V1*** and ***V2*** are as defined in Sections 3.1 and 3.2. Configurations 1 and 2 are designed to test the combination of text and traditional visual content features. Configuration 3 and 4 are used to demonstrate the effectiveness of the new adaptive matching pursuit features.

B. For each feature configuration, we perform two tests based on two types of classifiers in order to compare the performance between the traditional machine learning approach and our co-training approach.

- The first classifier, called **soft-SVM**, is based on a supervised learning approach using the probabilistic SVM. It uses the full set of labeled data with an average of 216 images per concept class for training the text and visual classifiers separately. It then combines the list of concepts labeled by both classifiers during testing.

- The second classifier is based on the co-training approach as discussed in Section 3.3. It is called **Co-Train (n)**, where n is the average number of initial training samples used in each concept class to kick-start the training process. For this test, we want to verify that we can use a small number of training samples (i.e. small n) as compared to the above supervised learning approach in achieving comparable performance. At the end of each co-training iteration, we randomly choose up to *p* (*p=3*) new sample images for each category and add them to the training set. We perform up to 50 iterations or when no more new samples can be found.

4.3 Parameter for Co-Training

In order to determine a good value for n, we conduct a series of tests based on feature configuration 4. We vary the value of n from 20 to 90, and compute the classification performance of the resulting classifier **Co-Train(n)** in terms of F_1 measure [21] at the end of training process. The results are presented in Figure 1.

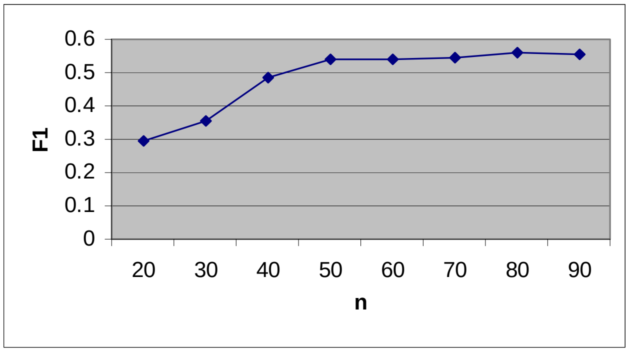


Figure 1: Performance of Co-Train(n) for different n values

From the Figure, we can clearly observe that the performance of the classifier increases rapidly as n is increased from 20 to 50, after which its performance is relatively steady. The results show that n=50 offers a good number of initial training samples to use to kick start the co-training process. At n=50, **Co-Train(n)** uses only about 23% of training samples as compared to the average of 216 samples per concept class used in **soft-SVM**.

4.4 Co-Training Comparison

Table 2 presents the comparative results between **Co-Train(n)** and **soft-SVM** on the range of feature configurations. In order to provide the baselines for the evaluation, we carry out four tests by using only textual or visual features to represent the image contents: (a) ***T1*** textual representation (or classifier $\boldsymbol{H^{T1}}$), as is done in many existing web-based systems; (b) ***T2*** textual representation (or classifier $\boldsymbol{H^{T2}}$); (c) ***V1*** visual representation (or classifier $\boldsymbol{H^{V1}}$); and (d) ***V2*** visual representation (or classifier $\boldsymbol{H^{V3}}$). In addition, we also include the results of applying the traditional learning approach based on Configuration 4 by using only an average of 50 training samples per concept class as is done in **Co-Train(50)**; the resulting system is called **soft-SVM(50)**.

From Table 2, we can draw the following observations.

First, the use of purely textual or visual representation (the baselines) could achieve an F_1 measure of between 15% to 24%. This is far below what is achievable by systems that combine both textual and visual features.

Second, **soft-SVM(50)** could achieve an F_1 performance of 37%, which is much lower than what is achievable by **soft-SVM** that uses the full set of training samples (or **soft-SVM(216)**).

Third, configurations incorporating ***T2*** text representation perform better than the corresponding configurations utilizing the ***T1*** text representation. This demonstrates that the judicious use of surrounding text, in conjunction with visual features, helps to improve both recall and precision of image annotation.

Fourth, the combination of color and adaptive matching pursuit texture features (***V2***) significantly outperforms systems that use only traditional visual features (***V1***). In fact, the combination of the ***T2*** and ***V2*** feature sets (configuration 4) achieves the best F_1 measure of over 54%. This result points to the use of adaptive features to overcome the fundamental problems of content-based image retrieval. With additional tuning of the system, we expect the results to improve further.

Fifth, the co-training framework could achieve comparable or marginally better performance than the traditional machine learning approach in all test configurations. The result is significant as co-training requires a much smaller number of training samples (<23%) than needed for effective learning by the traditional machine learning approach.

Table 2: Results of co-training experiments

CONFIGURATIONS and TESTS	RECALL	PRECISION	F_1
Baseline (T1)	0.34	0.10	0.15
Baseline (T2)	0.36	0.11	0.17
Baseline (V1)	0.29	0.15	0.19
Baseline (V2)	0.31	0.21	0.24
Configuration 1			
Soft-SVM	0.48	0.24	0.32
Co-Train(50)	0.50	0.25	0.33
Configuration 2			
Soft-SVM	0.52	0.26	0.35
Co-Train(50)	0.55	0.27	0.36
Configuration 3			
Soft-SVM	0.53	0.33	0.41
Co-Train(50)	0.56	0.34	0.42
Configuration 4			
Soft-SVM(50)	0.67	0.26	0.37
Soft-SVM	0.79	0.39	0.52
Co-Train(50)	0.81	0.41	0.54

4.5 Examples of Image Annotation

Figure 2 gives some examples of images annotated using our co-training approach. Column 1 of Figure 2 gives the original web pages with the image and embedded text, while column 2 shows both the original text annotations extracted from the web pages as well as the annotation learned by our system. The results demonstrate that our annotation scheme could give reasonably accurate and complete annotations.

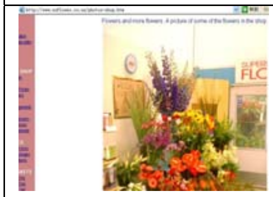
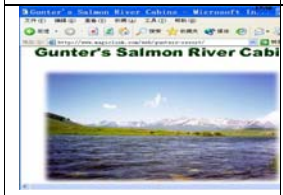
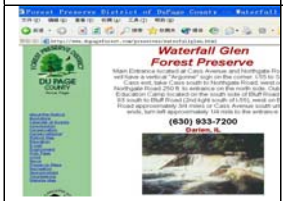
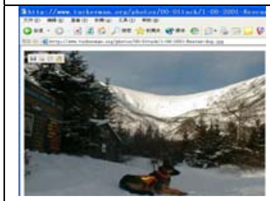
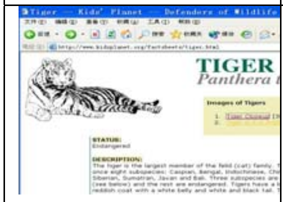
1) Original web pages	2) Original text annotations and learned concepts
	Original: Page title: Auckland Flower Wholesalers Ltd- Florist Shop Photos Image name: shop-interior-flower-stand Image ALT: Flowers and more flowers, A picture of our retail flower stand in the shop Learned: flower
	Original: Page title: Gunter's Salmon River Cabins Image name: River Image ALT: none Context: none Learned: River, Mountain
 Waterfall Glen Forest Preserve DU PAGE COUNTY (630) 933-7200	Original: Page title: Forest Preserve District of Dupage County -- Waterfall Image name: waterfall Image ALT: none Context: Waterfall Glen, Forest Preserve; Main Entrance located at Cass Avenue Learned: waterfall, river
	Original: Page title: Directory of photos Image name: rescue dog Image ALT: none Context: none Learned: Dog, mountain
	Original: Page Title: Tiger -- Kids' Planet -- Defenders of Wildlife Image name: tiger Image ALT: none Context: none Learned: Tiger

Figure 2: Examples of image annotations using our approach

Further analysis of the results reveals that the system performs well for concept classes like *tiger*, *financial-chart*, *waterfall* and *ski*, each with F_1 score of over 60%. The reasons are: (a) the images in these classes tend to have coherent color and texture characteristics, and/or (b) these concepts usually appear as primary focuses in these images with relevant text annotation. On the other hand, the system does badly on concept classes like *dog*, *beach* and *river*, each with F_1 score of less than 40%. There are several possible reasons. For example, the dogs in the *dog* images tend to be small with no fixed colors, while images of *river* and *beach* tend to be confused visually. Moreover, these concepts are often the secondary focuses of these images, and thus their text annotations often do not contain good descriptions of these concepts directly. The problems indicate that we need to develop better image region extraction and annotation capability, and better linguistic analysis techniques to extract secondary subjects in text annotations of such images.

5. CONCLUSION

This paper has explored the use of a bootstrapping approach to automatically annotate the large number of images obtainable from the Web using a pre-defined set of concepts. The main contribution of this work is two-fold. (a) We develop a co-training approach that fuses evidences from image contents and their associated HTML text. (b) We demonstrate that the co-training approach could achieve a level of performance comparable to that of the supervised learning approach but requiring a much smaller set of labeled training samples (<23% in our test). The need to acquire a large number of labeled training samples is both time consuming and error prone especially on the huge and dynamic Web. The use of a bootstrapping framework therefore points towards the development of practical and maintainable image retrieval systems.

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 on the Web. We will carry out further research in the following areas. First, we will further investigate the consistency and scalability of the co-training approach by carrying out both theoretical studies and large-scale empirical experiments. Second, we will refine the adaptive content features we have developed to better model a wide variety of image contents. Finally, we will extend our work to annotate large scale video collections available at the TREC video forum [25] using a larger set of pre-defined concepts.

6. REFERENCES

[1] Yuksel Alp Aslandogan & Clement T. Yu. "Multiple evidence combinationm in image retrieval: Diohenese searches for people on the Web". ACM SIGIR '2000, Athens, Greece, Jul 2000.

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

[3] F. Bergeaud & S. Mallat. Matching pursuits of images. ICIP '95, 1, 53-56, Washington DC, Oct, 1995.

[4] A. Blum & T. Mitchell. Combined labeled data and unlabelled data with co-training. Proceeding of Annual Conference on Computational Learning Theory. 1998.

[5] C. Carson, M. Thomas, J.M.Hellerstein & J. Malik. BlobWorld: A system for region-based image indexing and retrieval. Int Conf Visual Inf Sys, 1999.

[6] Edward Chang, Kingshy Goh, Gerard Sychay & 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).

[7] T. S. Chua, Y. Zhao, L. Chaisorn, C.-K. Koh, H. Yang, H. Xu and Q. Tian. TREC 2003 Video Retrieval and Story Segmentation Task at NUS PRIS. 2003. http://www-nlpir.nist.gov/projects/tv.pubs.org

[8] M. Collins & Y. Singer. Unsupervised models for name entity classification. Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural language Processing and Very Large Corpora. 1999.

[9] H. Feng & T.-S. Chua. "A bootstrapping approach to annotating large image collection". Workshop on "Multimedia Information Retrieval", organized in part of ACM Multimedia 2003. Berkeley, USA. Nov 2003, 55-62.

[10] A. Hauptman, R.V. Baron, M.-Y. Chen, M. Christel, P. Duygulu, C. Huang, R. Jin, W.-H Lin, T. Ng, N. Moraveji, N. Papernick, C.G.M. Snoek, G. Tzanetakis, J. Yang, & H.D Wactlar. Informedia at TRECVID 2003: analyzing and searching broadcast news video, 2003, http://www-nlpir.nist.gov/projects/tv.pubs.org

[11] J Jeon, V. Lavrenko & R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. ACM AIGIR '2003, Toronto, Canada. 119-126.

[12] Rainer Lienhart & Alex Hartmann. "Classifiying images ont eh web automatically". Joutrnal of electronic imaging. 11(4), Oct 2002. 1-10.

[13] S.G.. Mallat & Z.F. Zhang. Matching pursuits with time-frequency dictionaries. IEEE Transactions on Signal Processing, 41(12), 3397-3415, 1993.

[14] B.S. Manjunath & W.Y. Ma. Texture features for browsing and retrieval of image data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(8), 837-842, 1996.

[15] Y. Mori, H. Takahashi & 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).

[16] K. Nigam & R. Ghani. Analyzing the effectiveness and applicability of co-training. Proceedings of the 9th International Conference on Information and Knowledge management. 2000.

[17] Lexin Pan. Image8: an image search engine for the Internet. Honours year project report. School of Computing, National University of Singapore, Apr 2003.

[18] D. Pierce & C. Cardie. Limitations of co-training for natural language learning from large datasets. Proceeding of Empirical Methods in Natural Language Processing. 2001.

[19] J.C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In 'Advances in Large Margin Classifiers', A.J. Smola, P. Bartlett, B. Scholkopf & D. Schuurmans (Eds). MIT Press, 1999.

[20] H.R. Rabiee, R.L. kashyap & S.R. Safavian. Adaptive multiresolution image coding with matching and basis pursuits. IEEE ICIP '96, EPFL, Switzerland, Sept, 1996.

[21] G. Salton & M.J. McGill. Introduction to modern information retrieval. McGraw Hill. 1983.

[22] H.M. Sanderson & M.D. Dunlop. "Image retrieval by hypertext links". ACM SIGIR' 1997. 296-303.

[23] H.-T. Shen, B.-C. Ooi & K.-L. Tan. "Giving meaning to WWW images". ACM Multiemdia '2000. LA, USA. 39-47.

[24] M. Shenier & M. Abedel-Mottaleb. Exploiting the JPEG compression scheme for image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(8), 849-853, 1996.

[25] A. Smeanton, W. Kraaij & P. Over. TRECVID 2003 - An Introduction. http://www-nlpir.nist.gov/projects/tv.pubs.org, 2003.

[26] J.R. Smith and S.-F. Chang. VisualSeek: A fully automated content-based query system. ACM Multimedia '1996. 87-92.

[27] R. Shi, H. Feng, T.-S. Chua & C.-H. Lee. An adaptive image content representation and segmentation approach to automatic image annotation. To appear in Conference on Image and Video Retrieval (CIVR'04), Dublin, Jul 2004.

[28] M. Unser. Texture classification and segmentation using Wavelet frames. IEEE Transactions on Image Processing, 4(11), 1549-1560, 1995.

[29] V. Vapnik. The nature of statistical learning theory. Springer, New York, 1995.

[30] J.Z. Wang & J. Li. Learning-based linguistic indexing of pictures with 2-D MHHMs. ACM Multimedia '2002, 436-445.

[31] J.Z. Wang, J. Li & G. Wiederhold. SIMPLIcity: semantics-sensitive integrated matching for picture libraries. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(9), 947-963, 2001.

[32] K. Yanai. "Generic image classificaiton using visual knowledge on the web". ACM Multiemdia '2003. Berkeley, USA. 167-176.

[33] C. Zhang & T. Chen. An active learning framework for content-based information retrieval. IEEE transactions on multimedia. 4, 260-268, 2002.