A Learning-based Hybrid Tagging and Browsing Approach for Efficient Manual Image Annotation

Rong Yan, Apostol (Paul) Natsev, Murray Campbell
IBM T.J. Watson Research Center
19 Skyline Drive, Hawthorne, NY, 10532
{yanr, natsev, mcam}@us.ibm.com*

Abstract

In this paper we introduce a learning approach to improve the efficiency of manual image annotation. Although important in practice, manual image annotation has rarely been studied in a quantitative way. We propose formal models to characterize the annotation times for two commonly used manual annotation approaches, i.e., tagging and browsing. The formal models make clear the complementary properties of these two approaches, and inspire a learning-based hybrid annotation algorithm. Our experiments show that the proposed algorithm can achieve up to a 50% reduction in annotation time over baseline methods.

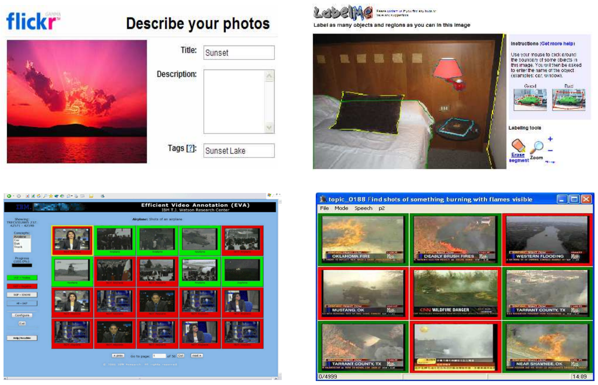


Figure 1. Examples of manual image annotation systems. Top: tagging (Flickr and LabelMe), bottom: browsing (EVA and XVR).

1. Introduction

Recent increases in the adoption of devices for capturing digital media and mass storage systems have led to an explosive amount of images and videos stored in personal collections or shared online. To effectively manage, access and retrieve these data, a widely adopted solution is to associate the image content with semantically meaningful labels, a.k.a. *image annotation* [10]. Two types of image annotation approaches are available: automatic and manual. Automatic image annotation, which aims to automatically detect the visual keywords from image content, have attracted a lot of attention from researchers in the last decade [2, 9, 11, 5, 7, 4, 3]. For instance, Barnard et al. [2] treated image annotation as a machine translation problem. Jeon et al. [9] proposed an annotation model called cross-media relevance model(CMRM), which directly computed the probability of annotations given an image. The ALIPR system [11] used advanced statistical learning techniques to provide fully automatic and real-time annotation for digital pictures. Fei-Fei et al. [5] showed that the object categories can be learned even with a handful of images. These automatic annotation approaches have achieved notable success, especially when the keywords have frequent occurrence and strong visual similarity. However, it remains a challenge to accurately annotate other more specific and less visually similar keywords. For example, the best algorithm for the CalTech-256 benchmark [7] reported a mean accuracy of 0.35 for 256 categories with 30 examples per category. Similarly, the best automatic annotation systems in TRECVID 2006 [14] produced a mean average precision of only 0.17 on 39 concepts.

Along another direction, recent years have seen a proliferation of manual image annotation systems for managing online/personal multimedia content. Examples include Aria [12] for personal archives, Flickr [1] , LabelMe [15] and ESP Game [18] for online content. This rise of manual annotation partially stems from its high annotation quality for self-organization/retrieval purpose, and its social bookmarking functionality in online communities. Manual image annotation approaches can be categorized into two types as shown in Figure 1 (details in Section 2). The most common approach is *tagging*, which allows users to annotate images with a chosen set of keywords ("tags") from a vo-

*This material is based upon work supported by the US Government. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the US Government.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

cabulary. Another approach is *browsing*, which requires users to sequentially browse a group of images and judge their relevance to a pre-defined keyword. Both approaches have strengths and weaknesses, and in many ways they are complementary to each other. But their successes in various scenarios have demonstrated the possibility to annotate a massive number of images by leveraging human power.

However, manual image annotation can be tedious and labor-intensive. Therefore, it is of great importance to consider using automatic techniques to speed up manual annotation. In this work, we assume users will drive the annotation process and manually examine each image in order to guarantee labeling accuracy, but in addition, we use automatic learning to improve the annotation efficiency by adaptively suggesting the right images, keywords and annotation interfaces. This is different from *automatic image annotation* that directly construct visual models based on training examples. It also differs from *active learning* [16] that aims to iteratively optimize automatic learning performance on visual features rather than minimizing annotation time. This may lead to inaccurate labeling results and poor user experience by showing the most ambiguous examples.

Although manual annotation currently provides a more mature and accurate solution for image annotation, it has attracted much less attention from the vision community. We attribute this to a lack of quantitative annotation time models to formalize the annotation process, and thus prevent it from being studied in large-scale collections. Therefore, what we first propose are a pair of formal annotation time models for two popular manual annotation approaches, i.e., tagging and browsing. To our best knowledge, this is the first attempt to quantify the manual image annotation process, and it serves as a theoretical foundation to analyze large-scale manual annotation without time-consuming user studies. Based on the time models, we further propose a learning-based hybrid annotation algorithm which automatically learns from visual features and adaptively chooses browsing/tagging interfaces for the right set of keywords/images. Both our simulation and empirical results on the TRECVID [14] and Corel [2] collections confirm the validity of the time models, as well as demonstrate that the proposed algorithm can achieve an up to 50% reduction in annotation times and can consistently outperform the baseline methods in the entire annotation process.

2. Manual Image Annotation Methods and Time Models

In this section, we introduce and discuss two types of manual image annotation approaches, i.e., tagging and browsing. We also propose two models to measure their annotation efficiency, which offers the foundation for the rest of our discussions. Formally, let us suppose we have to annotate a set of images $\mathcal{I} = \{I_l\}_{l=1..L}$ with a set of keywords $\mathcal{W} = \{W_k\}_{k=1..K}$.[1] L_k is the number of relevant images for W_k, and K_l is the number of keywords associated with I_l. The goal of manual annotation is to identify the relevance between each pair of image I_l and keyword W_k, or equivalently, *annotate image I_l with keyword W_k*. Once the relevance between I_l and all the keywords have been identified, we say I_l *is annotated*, otherwise I_l *is unannotated*.

2.1. Tagging

Tagging allows the users to annotate images with a chosen set of keywords ("tags") from a controlled or uncontrolled vocabulary. This type of approaches is the basis for most of the current image annotation/tagging systems, although it can be implemented in a variety of ways with respect to interface designs and user incentives. For example, Flickr [1] encourages users to create free-text tags for each uploaded image. ESP Game [18] motivates users to annotate photos with freely chosen keywords in a gaming environment. One advantage for tagging is that annotators can use any keywords in the vocabulary to annotate the target images. However, this flexibility might result in a "vocabulary problem" [6], which means multiple users or a single user in a long period of time can come up with different words to describe the same concept. Moreover, it can be more time-consuming for general users to input new keywords, as compared with simply browsing and judging the relevance between images and pre-defined keywords.

In order to quantitatively analyze the efficiency of tagging approaches, we must design a formal model to represent its expected annotation time for each image. To begin, we can assume that the more keywords users annotate, the larger the annotation time is. Our user study described in Section 4.1 confirmed that this assumption is reasonable, but it also shows that the annotation time is not exactly proportional to the number of keywords. This is because, for each image, users always need additional time up-front to understand the image content in order to make their decisions. The above observations suggest modeling the tagging time T_l for the l^{th} image as a function of four major factors, i.e., the number of image keywords K_l, the average time for designing/typing one word t_f, the initial setup time for annotation t_s and a noise term ϵ, which follows a zero-mean probability distribution. Based on our user study, we find it is sufficient to adopt a linear time model to represent the annotation time for each image, i.e., $T_l = K_l t_f + t_s + \epsilon$. Its mean can be derived as $t_l = K_l t_f + t_s$. For a total of L images, the overall expected annotation time is

$$t = \sum_{l=1}^{L} K_l t_f + L t_s \quad or \quad t = \sum_{k=1}^{K} L_k t_f + L t_s. \qquad (1)$$

[1] Theoretically, K can be infinity for an unlimited vocabulary. In this work, we assume K is bounded by the number of unique English terms.

Note that the parameters t_s and t_f are not required to be constant in all scenarios. Instead, they can be affected by a number of factors, such as interface design, input device, personal preference and so on. For example, annotation on cell phones will have a larger t_f than annotation on desktop computers. Therefore, rather than estimating fully accurate parameters for any specific settings, we mainly focus on examining the correctness of the model assumptions, and use them to develop better manual annotation algorithms.

2.2. Browsing

Another type of annotation approach, *browsing*, requires users to browse a group of images and judge the relevance of each image to a given keyword. Because browsing annotation needs to start with a controlled vocabulary defined by domain experts or a seeded keyword manually initialized, it is not as flexible and widely applied as tagging. However, browsing has advantages on several aspects. For instance, it allows users to provide more complete annotation than tagging [17], because users only focus on one specific keyword at a time. Moreover, the time to annotate one keyword by browsing is usually much shorter. Therefore, recent years have seen more and more browsing annotation systems being developed. One such example is the Efficient Video Annotation (EVA) system [17], which allows multiple users to collaboratively annotate the same image collection by browsing. Extreme video retrieval (XVR) [8] follows a similar idea by asking users to quickly browse the search results to judge their relevance.

Similar to tagging, we design a formal model to quantify the efficiency of browsing. First, the overall annotation time should be related to the number of images and the number of unique keywords. According to Section 4.1, we also find that the time for annotating a relevant image is significantly larger than the time for skipping an irrelevant image, because users tend to spend more time in examining the correctness on relevant images. Thus we model the browsing annotation time T_k for the k^{th} keyword using four major factors, i.e., the number of relevant images L_k, the average time to annotate a relevant image t_p, the average time to annotate an irrelevant image t_n and a zero-mean noise term ϵ. The number of irrelevant images is simply $\bar{L}_k = L - L_k$ and hence a reasonable linear time model is $T_k = L_k t_p + (L - L_k)t_n + \epsilon$. For a total of K keywords, the overall expected annotation time is

$$t = \sum_{k=1}^{K} \left[L_k t_p + (L - L_k) t_n \right]. \qquad (2)$$

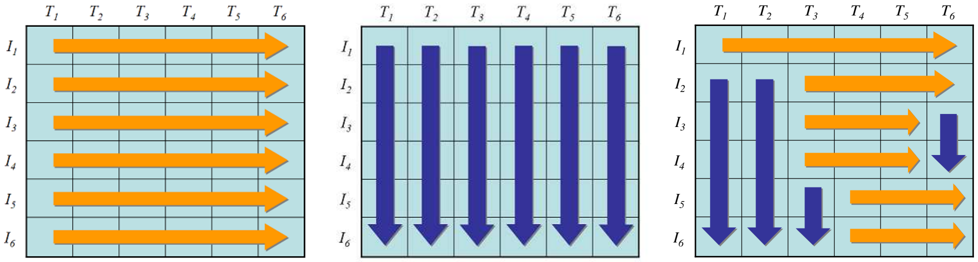
To summarize, these two annotation approaches are essentially complementary from many perspectives. For example, tagging has less limitations on the choice of words and users only need to consider relevant keywords for each image. But the annotated words must be re-calibrated due to the vocabulary problem. It also requires more time to determine and input the given keyword. On the contrary, browsing must work with one pre-defined keyword at a time and requires users to judge all possible pairs of images/keywords. But the effort to determine image relevance by browsing is usually much less than that by tagging, i.e., t_p, t_n is typically much smaller than t_f, t_s. Therefore, tagging is more suitable for annotating infrequent keywords such as specific person/location names, and browsing works better for frequent keywords such as “person”/“face”.

3. Learning-based Hybrid Annotation

By merging the strengths of tagging and browsing, we have developed a more efficient algorithm for manual image annotation. Because it is suggested that tagging/browsing is suitable for infrequent/frequent keywords respectively, we propose a learning-based hybrid annotation approach which automatically learns from visual features and adaptively chooses browsing/tagging interface for the right set of keywords/images. To illustrate, we can view image annotation as a problem of filling binary relevance judgments in a matrix of size $K \times L$. In this case, tagging is equivalent to annotating the matrix row by row, as shown in Figure 2(a), and browsing is equivalent to annotating column by column, as shown in Figure 2(b). However, neither of these approaches are always ideal for the entire space. In contrast, Figure 2(c) describes the learning-based annotation using the same annotation matrix. The algorithm starts by tagging some initial selected images. With more annotations collected, it attempts to dynamically find a batch of unannotated images for potential keywords and ask users to annotate them in a browsing interface. The algorithm will iterate until all the images are shown in the tagging interface so as to guarantee none of the keywords is missing. In the rest of this section, we provide more analysis and implementation details for the proposed algorithm.

3.1. Analysis

To design the learning-based annotation algorithm, it is instructive to study when it can have a smaller annotation time than simple tagging/browsing. Let us first break down its total annotation time by keywords, or equivalently, column by column in the annotation matrix. Suppose the proposed algorithm has obtained L_k relevant labels for keyword W_k. Then we can assume $\beta_k L_k$ relevant labels come from browsing and the other $(1 - \beta_k)L_k$ labels come from tagging, where β_k is called *browsing recall*. However, β_k is not enough to describe the total annotation time, because, based on the proposed time models, browsing irrelevant images will also introduce additional cost. Therefore we need to introduce *browsing precision* γ_k to represent the



(a) Tagging (b) Browsing (c) Learning

Figure 2. Illustration of three image annotation approaches, where rows stand for images and columns stand for keywords.

proportion of relevant images in all the browsed images. In this case, the number of irrelevant browsed images is $\beta_k L_k(1-\gamma_k)/\gamma_k$ and the total annotation time is,

$$t = \sum_{k=1}^{K}\left[\beta_k L_k\left(t_p + \frac{1-\gamma_k}{\gamma_k}t_n\right) + (1-\beta_k)L_k t_f\right] + Lt_s. \quad (3)$$

Our analysis in the Appendix shows that if an annotation algorithm is more efficient than tagging/browsing, its browsing recall and precision must satisfy

$$\beta_k \geq \max\left(0, \frac{1-ab_k}{1-a}\right), \gamma_k \geq \max\left(a, \frac{a\beta_k}{ab_k-(1-\beta_k)}\right) \quad (4)$$

where $a = t_n/(t_f+t_n-t_p)$, $b_k = L/L_k$. These inequalities offer us more insights to develop efficient annotation algorithms. Eqn(4) suggests that for keyword W_k, we should use browsing to annotate at least $(1-ab_k)L_k/(1-a)$ relevant images. Otherwise, simple browsing/tagging annotation can be a better choice in this case. Furthermore, it also indicates that the browsing precision should be larger than a lower bound related to a, b_k and β_k.

Above analysis shows that an efficient annotation algorithm should be able to select a sufficient number of images for users to browse, and the browsing precision should be higher than a fixed lower bound. Therefore, the goal of the annotation algorithms is to identify a set of unannotated images that are likely to be relevant for a given keyword, update the annotation parameters, and guarantee the browsing precision to be larger than the lower bound. We mainly discuss the first task in the rest of this section and leave the discussions of other tasks to Section 3.2.

To identify relevant unannotated images, we translate manual annotation into an online learning problem, i.e., for keyword W_k, learn the visual model from available relevant images $\mathcal{I}'_k = \{I'_j\}_{j=1..m'}$, and use it to predict additional relevant images from the unannotated image pool $\mathcal{U}'_k = \{I_l | I_l \notin \mathcal{I}'_k\}$. Each image I is associated with a number of low-level features $\mathbf{x}$. Based on the annotation provided by users, we can learn their visual patterns by using kernel logistic regression, which aims to optimize the following empirical risk function for each keyword W_k,

$$R(f) = \sum_{i=1}^{m'} \log(1+e^{-y_i f(\mathbf{x}_i)}) + \lambda\|f\|^2_{\mathcal{H}},$$

where $\mathbf{x}_j$ is the feature for I'_j, $y_k \in \{-1, 1\}$ is binary relevance label, and $\mathcal{H}$ denotes a reproducing kernel Hilbert space(RKHS) generated by an arbitrary positive definite kernel K. According to the representer theorem, the relevance of unannotated images can be estimated from the minimizer $f(x)$ with weights α_i for each training example, $f(\cdot) = \sum_{i=1}^{m'} \alpha_i K(\mathbf{x}_i, \cdot)$. When users annotate an additional label $(y_m, \mathbf{x}_m), m = m'+1$, the optimization function must be updated accordingly. To reduce the computational demand, only the weight for the new example is updated based on the Newton-Raphson method. Since the optimization function is convex, the Newton method can guarantee to find the global optimum. To be more specific, the gradient and Hessian of the risk function with respect to $\alpha_{\bar{m}}$ can be written as,

$$\frac{\partial R(\alpha)}{\partial \alpha_{\mathbf{m}}} = \mathbf{K}_{\mathbf{m}}^{\mathbf{T}}\mathbf{p} + \lambda\mathbf{K}_{\mathbf{m}}^{\mathbf{T}}\alpha, \quad \frac{\partial^2 R(\alpha)}{\partial \alpha_{\mathbf{m}}^{\mathbf{2}}} = \mathbf{K}_{\mathbf{m}}^{\mathbf{T}}\mathbf{W}\mathbf{K}_{\mathbf{m}} + \lambda K_{mm}$$

where $\mathbf{K}$ is the kernel Gram matrix, $\mathbf{K_m}$ is the vector of $\{K(x_m, \cdot)\}$, K_{mm} is the element of $K(x_m, x_m)$, $\mathbf{p}$ denote the logistic model $1/(1+\exp(-\mathbf{K}\alpha))$, and $\mathbf{W}$ denote the matrix $diag(\mathbf{p_i}(1-\mathbf{p_i}))$. The Newton updates can be straightforwardly derived from the gradient and Hessian function. These updates are iterated until the risk function converges or the iteration number is larger than a threshold.

In our implementation, we select the RBF kernel, i.e., $K(x, y) = e^{-\rho\|x-y\|^2}$, to model non-linear decision boundary between positive/negative examples. Finally, after the optimal weight α_m is found, we can simply update the prediction function by $f(\cdot) \leftarrow f(\cdot) + \alpha_m K(x_m, \cdot)$.

3.2. Algorithm Details

The learning-based hybrid annotation is summarized in Algorithm 1. This algorithm starts with a number of unannotated images and a vocabulary of keywords. It first selects an image from the unannotated pool for tagging. After this image is completely tagged by the user, all the related variables are updated for its corresponding keywords W_k, i.e., the set of relevant images $\mathcal{I}'_k$, the kernel weight α as well as the prediction function $f_k(\cdot)$. By thresholding the prediction function, we generate the set of estimated relevant images $\mathcal{R}_k$ that represent the most potentially relevant

Algorithm 1 The learning-based annotation algorithm

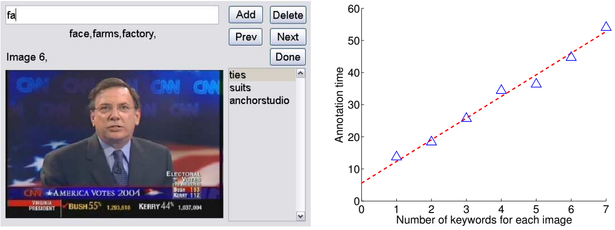
```
Input: Images $\{I_l\}$, keywords $\{W_k\}$, browsing batch size $S$
  1. Initialize adaptive threshold $\theta_k = 1$, browsing precision $\gamma_k = 1, \mathcal{I}'_k = \varnothing, \forall k = 1..K$ and $\mathcal{U} = \{I_l\}, \forall l = 1..L$;
  2. While there are unannotated images left, i.e., $\mathcal{U} \neq \varnothing$,
      (a) Ask user to tag the first image $I_l$ in $\mathcal{U}$;
      (b) For each keyword $W_k$ associated with $I_l$;
            i. Add $I_l$ into the labeled pool, $\mathcal{I}'_k = \mathcal{I}'_k \cup I_l$;
           ii. Update $\alpha$ and $f_k(\cdot)$;
          iii. Obtain the set of predicted relevant images $\mathcal{R}_k = \{I_l | f_k(\mathbf{x}_l) \geq \theta_k, I_l \notin \mathcal{I}'_k\}$;
           iv. If $|\mathcal{R}_k| \geq S$, then ask user to annotate $\mathcal{R}_k$ by browsing, update labeled pool $\mathcal{I}'_k$ and browsing precision $\gamma_k$, go to 2(b)ii;
            v. If $\gamma_k \geq \max(a, m'_k/m_k)$ (details below), then reduce $\theta_k$ to $\theta_k/2$, go to 2(b)iii;
      (c) Remove $I_l$ from $\mathcal{U}$, i.e., $\mathcal{U} = \mathcal{U} \setminus I_l$;
  3. Output the annotations for all images.
```

images identified by the system. If the number of these images is larger than a pre-defined batch size S, the browsing interface is activated to annotate all the images in $\mathcal{R}_k$. To avoid switching interfaces too frequently and disturbing the user experience, we typically set the batch size to be a large number and only invoke the browsing interface when there are a large number of relevant images available.

Since the set of estimated images are not guaranteed to be relevant, we use the new image annotations to predict the browsing precision and update the learning parameters accordingly. As shown in Eqn(4), browsing precision needs to be sufficiently large for the proposed algorithm to improve efficiency. Therefore, in the next step, if we find the browsing precision is larger than a lower bound, we will reduce the adaptive threshold θ_k and thus the browsing interface can be used to annotate as many images as possible until the browsing precision is too low. The bound, i.e., $\max(a, m'_k/m_k)$, is derived from Eqn(4) with $\beta_k = 1$ and m'_k/m_k to approximate the true frequency ratio L_k/L, where m'_k is the number of relevant images and m_k is the total annotated images for W_k so far. Above steps will be iterated until all the images are tagged.

4. Experiments

Our experiments are carried out on two large-scale image/video collections. The first collection is generally referred to as the TRECVID collection [14], which is the largest video collection with manual annotations available to the research community. We use the TRECVID 2005 development set which includes a total of 74,523 keyframes.


fa

face,farms,factory,

Image 6,

ties
suits
anchorstudio

Figure 3. Left: tagging interface. Right: tagging time statistics. The dash line is fitted based on the tagging model.

This collection consists of broadcast news videos captured from 11 different broadcasting organizations across several countries. Each image is officially annotated with 449 semantic labels [13]. For each image, we generate a 150-dimensional color histogram as the visual features.

The second collection is compiled from the Corel image dataset on 155 keywords and currently shared online by Barnard et al. [2]. The collection is organized into 10 different samples of roughly 16,000 images, where each image is associated with a vocabulary of 155 keywords such as “aircraft”, “sky”, “water” and so on. We use the first sample of the images in our experiments, which contains a total of 15,766 images. The low-level visual features are generated on image segments provided by N-Cuts. Each segment is associated with 46 features including size, position, color, oriented energy (12 filters), and a few simple shape features.

In the following discussions, we first present two user studies to confirm the validity of the tagging and browsing time models proposed in Section 2.2. We use them to simulate and examine the efficiency of the learning-based hybrid annotation algorithm. Finally, we report the real-world annotation performance of hybrid annotation by asking a user to annotate the image collection in one hour.

4.1. Annotation Time Models

To examine whether the proposed tagging/browsing time models are reasonable in practice, we conducted user studies on two different types of annotation systems. To validate the tagging time model, we developed a keyword-based annotation system. Figure 3 shows the snapshot of the tagging system, together with the distribution of the average tagging time against the number of keywords. The TRECVID collection with its controlled vocabulary is used in this experiment. A total of 100 randomly selected images have been annotated by a user. The number of annotated keywords per image ranges from 1 to 7. We also generate a dashed line by fitting the time statistics with the tagging model. It can be observed that the average annotation time for an image has a clear linear correlation to the number of keywords. Another interesting observation is that the dashed line does not go across the origin after an extrapolation to zero keywords. This means it takes additional time for a user to start annotating a new image due to a switch of the annotated

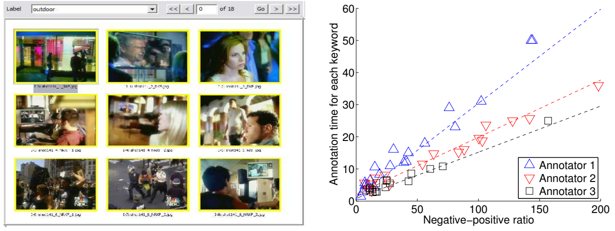


Figure 4. Left: browsing interface. Right: browsing time statistics. Dash lines are fitted based on the browsing model.

images. Based on the linear fitting, we can estimate the tagging model parameters for this user to be $t_f = 6.8$ seconds and $t_s = 5.6$ seconds.

To validate the browsing time model, we asked three different users to browse 25 keywords (extracted from 25 TRECVID'05 queries) in the TRECVID collection. The system snapshot is shown in Figure 4. For the purpose of better visualization, we slightly rewrite the browsing model to be $t_p + t_n \bar{L}_k / L_k = t / L_k$, where $\bar{L}_k / L_k$ is the ratio between the number of irrelevant images and relevant images, or called *negative-positive ratio*, t/L_k is the average time for collecting one positive image. If the proposed browsing model is satisfied, we should be able to identify a linear relation between $\bar{L}_k / L_k$ and t/L_k. Figure 4 plots the distributions of these two factors for the three users. As we can see, the dashed lines estimated by linear regression fit the true distribution quite well, which confirmed the browsing time model is reasonable for the practical annotation environment. We estimate the model parameters by averaging all three users, i.e., $t_p = 1.4$ seconds, and $t_n = 0.2$ seconds.

4.2. Annotation Results on Image Collections

In this section, we evaluate the performance of the annotation algorithms in large-scale image collections. The ground-truth image labels are used to simulate real user annotations in order to avoid prohibitive human resources for evaluation. The annotation time is obtained based on model parameters t_f, t_s, t_p and t_n estimated in the previous section. For learning-based annotation, we set the RBF kernel parameter ρ to 1 and the batch size S in Algorithm 1 to 50.[2]

To illustrate how learning-based annotation can automatically switch the annotation methods and select a batch of images to annotate by browsing, Figure 5 shows the initial 12 images which learning-based annotation asks for tagging in the TRECVID collection, and the first batch of 40 images selected for browsing. The top of this figure lists all the initial tagged images and their associated keywords. After the initial images are tagged, the algorithm found 5 of these images are annotated as "politicians" and predicted a number of other images can be potentially related to the keyword "politicians" based on visual appearance. Thus, it switched to present a batch of 30 images for annotating the keyword "politicians" by browsing. Since a significant faction of these browsed images are related to "politician", the user can save a lot of time without re-typing the same keyword again and again. This learning-based algorithm also helps to calibrate annotated keywords, provide more complete annotations on each concept and make browsing annotation applicable even for an uncontrolled vocabulary.

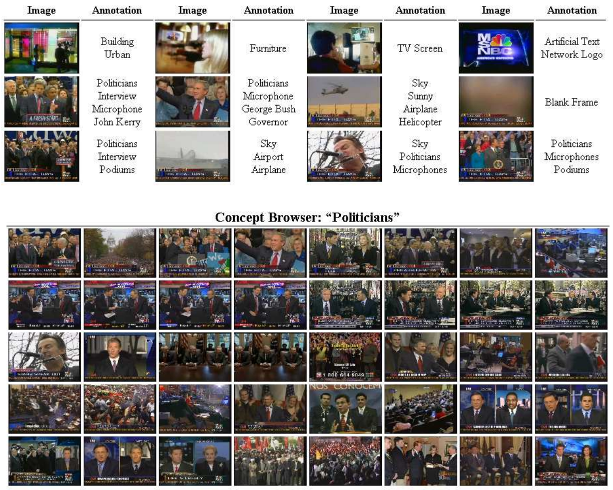


Image
Annotation
Image
Annotation
Image
Annotation
Image
Annotation
Building Urban
Furniture
TV Screen
Artificial Text Network Logo
Politicians Interview Microphone John Kerry
Politicians Microphone George Bush Governor
Sky Sunny Airplane Helicopter
Blank Frame
Politicians Interview Podiums
Sky Airport Airplane
Sky Politicians Microphones
Politicians Microphones Podiums
Concept Browser: "Politicians"

Figure 5. The images selected by learning-based annotation for the TRECVID collection. Top: initial 12 images for tagging (with keywords shown). Bottom: first group of 40 images for browsing.

Next we present the annotation performance of learning-based annotation (**LBA**) together with two baseline algorithms, i.e., tagging (**Tag**) and browsing (**Browse**). All the algorithms are evaluated on both the TRECVID and Corel collection. The first performance criterion is the total annotation time when the annotation process is completed. In addition, we also propose three new measures – *macro-recall*, *micro-recall* and *hybrid-recall* – to evaluate the annotation quality, because the accuracy-based performance measure is no longer applicable in this case given that all user annotations are correct. In this work, *recall* is defined as the ratio of the number of annotated relevant images to the total number of relevant images. Similar to text classification [19], macro-recall r_a is the average of the recalls for each keyword, and micro-recall r_i is the recall on the entire image-keyword space. In some sense, macro-recall measures the annotation diversity and micro-recall measures the annotation completeness, where both of them are important to describe the annotation quality. Hybrid-recall is the harmonic mean of the macro-recall and micro-recall, $r_h = 2r_a r_i / (r_a + r_i)$, designed in the same principle as the F1 measure [19].

Figure 6 provides a detailed comparison between three annotation algorithms. All recall measures are reported with a growing annotation time until the annotation pro-

[2] Our following observations and discussions still hold if we shifted one of these 6 parameters x to be any values between $[0.5x, 2x]$, although these experiments are not shown in the paper due to the space limit.

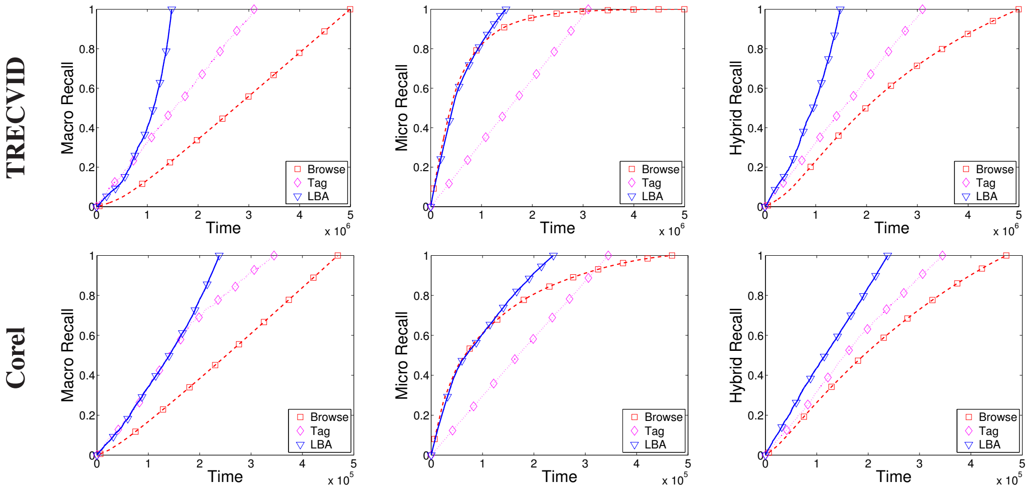


Figure 6. Annotation performance as a function of annotation time on the TRECVID and Corel collections.

cess ends. It is obvious to see that the learning-based annotation is superior to the baseline tagging/browsing methods in terms of total annotation time. For example, in the TRECVID dataset, learning-based annotation reduces the annotation time to be 50% of tagging time and 30% of browsing time. Their improvement in the Corel collection is also considerable, although it is relatively smaller than that in the TRECVID collection because of a lower number of keywords per image on average. A comparison between the curves on micro-recall and macro-recall shows that tagging is good at improving the macro-recall, while browsing does well in improving micro-recall. This is because tagging random images can bring a wide coverage of various keywords at the beginning, but browsing methods only focus on annotating the most frequent keywords at the early stage. This shows an important trade-off between micro-recall and macro-recall. Thus it is more impressive to observe that learning-based annotation is able to outperform tagging/browsing in terms of both macro-recall and micro-recall, as well as hybrid-recall. Our last observation is that learning-based annotation has a significantly higher hybrid-recall at the end of its labeling process. This indicates that within the same amount of time, learning-based annotation allows us to collect more image keywords with a higher annotation diversity.

4.3. Empirical Annotation Results

To verify the simulation results, we implemented a hybrid annotation system and asked a user to manually annotate 39 LSCOM-lite keywords [13] on a subset of the TRECVID collection, which includes the keyframes of 10 randomly selected videos. We recorded the statistics for three annotation approaches, i.e., tagging, browsing and learning-based annotation, around every five seconds. Each annotation process lasted for one hour. The annotation parameters are re-estimated using linear regression on all 2142 annotation statistics. For this user, these parameters are set to $t_p = 1.16s, t_n = 0.17s, t_f = 4.01s, t_s = 2.02s$, which are slightly less than the value used in the simulation.

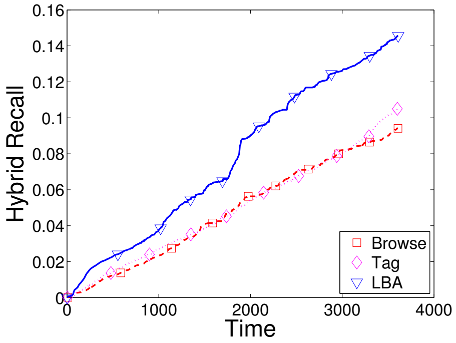


Figure 7. Comparing hybrid recall of three manual annotation approaches against the annotation time on the TRECVID collection.

Method	$\mathbf{N_i}$	$\mathbf{N_t}$	$\mathbf{N_p}$	$\mathbf{N_n}$	$\mathbf{T_{est.}}$	$\mathbf{T_{true}}$
Tag	405	706	0	0	3649s	3600s
Browse	0	0	1393	11693	3603s	3608s
Hybrid	194	321	1009	3527	3478s	3601s

Table 1. Comparing estimated annotation time ($T_{est.}$) with true annotation time (T_{true}). The number of annotated images are also shown (N_i: tagged images, N_t: tagged words, N_p: relevant browsed images, N_n: irrelevant browsed images).

Table 1 shows the number of tagged/browsed images at the end of the annotation process, and compares the estimated annotation time (based on time models) with the true annotation time recorded. It can be found that the estimated time closely approximates the true annotation time and the error is less than 4% in all three cases. This again confirms the correctness of the proposed time models in a large-scale annotation environment. Figure 7 shows the curve of hybrid recall as a function of annotation time [3]. The hybrid recall

[3] The scale in Fig. 7 is smaller since the annotation is incomplete.

curves of tagging and browsing are similar to each other because of their complementary properties on the macro-recall and micro-recall. The learning-based annotation, on the other hand, achieves a 50% improvement over tagging and browsing in terms of hybrid recall. This observation is in line with the simulation results presented before.

5. Conclusion

In this paper we have presented a learning approach for improving the efficiency of manual image annotation. This approach was inspired by our quantitative study of two widely used annotation approaches, i.e., tagging and browsing. We have proposed models to describe the processing time for tagging and browsing, and the validity of these models has been confirmed by our user studies. The quantitative analysis makes clear the complementary nature of tagging and browsing, and led us to propose a learning-based hybrid annotation algorithm which adaptively learns the most efficient annotation interface for selected keywords and images. Our simulation and empirical results on the TRECVID and Corel collections show that the proposed algorithm achieves up to a 50% reduction in annotation time and considerably outperforms the baseline approaches in terms of macro-recall, micro-recall, and hybrid-recall.

We expect this work to open up new research directions in modeling manual image annotation. For instance, the current annotation time models can be refined to incorporate more user factors such as keyword missing rate, context switching cost, or vocabulary size. We can also consider other learning algorithms to support fast visual model updates. Finally, the interface design and user incentive for hybrid annotation algorithms can be further discussed.

Appendix: Derivation for Eqn(4)

We analyze the annotation time for each keyword separately. For a given keyword W_k, the time of learning-based annotation should be lower than simply tagging or browsing W_k, otherwise either tagging or browsing should be used. When the learning-based annotation time is smaller than the tagging time for W_k, we can have

$$\beta_k L_k \left(t_p + \frac{1-\gamma_k}{\gamma_k} t_n\right) + (1-\beta_k) L_k t_f \leq L_k t_f$$
$$\Leftrightarrow \quad t_p + \frac{1-\gamma_k}{\gamma_k} t_n \leq t_f \quad \Leftrightarrow \quad \gamma_k \geq \frac{t_n}{t_f + t_n - t_p}. \qquad (5)$$

Similarly, when the learning-based annotation time is smaller than the browsing time for W_k, we can have

$$\beta_k L_k \left(t_p + \frac{1-\gamma_k}{\gamma_k} t_n\right) + (1-\beta_k) L_k t_f \leq L_k t_p + (L - L_k) t_n$$
$$\Leftrightarrow \quad (1-\beta_k)(t_f + t_n - t_p) + \frac{\beta_k}{\gamma_k} t_n - \frac{L}{L_k} t_n \leq 0.$$

By defining $a = t_n/(t_f + t_n - t_p)$ and $b_k = L/L_k$, we can simplify this inequality to be,

$$(1-\beta_k) + a\left(\frac{\beta_k}{\gamma_k} - b\right) \leq 0 \quad \Leftrightarrow \quad \frac{1}{\gamma_k} \leq \frac{ab + \beta_k - 1}{a\beta_k}.$$

The above inequality holds if and only if the following two conditions hold (since $\gamma_k \leq 1$),

$$\frac{\beta_k - 1 + ab}{a\beta_k} \geq 1, \quad \gamma_k \geq \frac{a\beta_k}{ab + \beta_k - 1}. \qquad (6)$$

Eqn(4) can be obtained by merging inequalities (5) and (6).

References

[1] Flickr. http://www.flickr.com.

[2] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3, 2002.

[3] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):394–410, 2007.

[4] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[5] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):594–611, 2006.

[6] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Comm. of the ACM*, 30(11):964–971, 1987.

[7] G. Griffin, A. Holub, and P. Perona. The caltech-256. Technical report, Caltech, 2006.

[8] A. G. Hauptmann, W.-H. Lin, R. Yan, J. Yang, and M.-Y. Chen. Extreme video retrieval: joint maximization of human and computer performance. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 385–394, New York, NY, USA, 2006. ACM Press.

[9] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 119–126, 2003.

[10] J. Kustanowitz and B. Shneiderman. Motivating annotation for personal digital photo libraries: Lowering barriers while raising incentives. Technical report, HCIL, Univ. of Maryland, 2004.

[11] J. Li and J. Z. Wang. Real-time computerized annotation of pictures. In *Proceedings of ACM Intl. Conf. on Multimedia*, pages 911–920, 2006.

[12] H. Lieberman, E. Rozenweig, and P. Singh. Aria: An agent for annotating and retrieving images. *Computer*, 34:57–62, 2001.

[13] M. Naphade, J. R. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-scale concept ontology for multimedia. *IEEE MultiMedia*, 13(3):86–91, 2006.

[14] P. Over, T. Ianeva, W. Kraaij, and A. F. Smeaton. Trecvid 2006 overview. In *NIST TRECVID-2006*, 2006.

[15] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. Technical report, MIT AI Lab Memo AIM-2005-025, 2005.

[16] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *ACM Intl. Conf. on Multimedia*, pages 107–118, 2001.

[17] T. Volkmer, J. R. Smith, and A. Natsev. A web-based system for collaborative annotation of large image and video collections: an evaluation and user study. In *Proceedings of the 13th ACM international conference on Multimedia*, 2005.

[18] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 2004.

[19] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proc. of the 14th ICML*, pages 412–420, 1997.