AnnoSearch: Image Auto-Annotation by Search[1]

Xin-Jing Wang, Lei Zhang, Feng Jing, Wei-Ying Ma
Microsoft Research Asia, 49 Zhichun Road, Beijing (100080), China
xjwang@msrchina.research.microsoft.com, {leizhang, fengjing,wyma}@microsoft.com

Abstract

Although it has been studied for several years by computer vision and machine learning communities, image annotation is still far from practical. In this paper, we present AnnoSearch, a novel way to annotate images using search and data mining technologies. Leveraging the Web-scale images, we solve this problem in two-steps: 1) searching for semantically and visually similar images on the Web, 2) and mining annotations from them. Firstly, at least one accurate keyword is required to enable text-based search for a set of semantically similar images. Then content-based search is performed on this set to retrieve visually similar images. At last, annotations are mined from the descriptions (titles, URLs and surrounding texts) of these images. It worth highlighting that to ensure the efficiency, high dimensional visual features are mapped to hash codes which significantly speed up the content-based search process. Our proposed approach enables annotating with unlimited vocabulary, which is impossible for all existing approaches. Experimental results on real web images show the effectiveness and efficiency of the proposed algorithm.

1. Introduction

The number of digital images has exploded with the advent of digital cameras which requires effective search methods. However, due to the semantic gap between image visual features and human concepts, most users prefer textual queries. Since manually annotating images is a very tedious and expensive task, image auto-annotation has become a hot research topic in recent years.

However, although many previous works have been proposed using computer vision and machine learning techniques, image annotation is still far from practical. One reason is that it is still unclear how to model the semantic concepts effectively and efficiently. The other reason is the lack of training data, and hence the semantic gap can not be effectively bridged.

With the prosperity of the Web, it has become a huge deposit of almost all kinds of data and provides solutions to many problems that were believed to be "unsolvable" [6][14].

Motivated by this and the successful search technologies in many commercial systems, in this paper, we propose a novel solution to image auto-annotation problem. Rather than training a concept model using supervised learning techniques as most previous works do, we propose a data-driven approach leveraging the Web-scale image dataset and search technology to learn relevant annotations.

In an ideal case, if a well annotated and unlimited-scale image database is available, then for any query image, we can find its duplicates in this database and simply propagate its annotation to the query image. In a more realistic case that the image database is of limited scale, we can still find a group of very similar images in terms of either global features or local features, extract salient phrases from their descriptions, and select the most salient ones to annotate the query image.

Thus to by-pass the semantic gap, we can divide-and-conquer the annotation problem in two steps: 1) find one accurate keyword for a query image; 2) given one keyword, find complementary annotations to describe the details of this image. The requirement in the first step is not as lacking in subtlety as it may first seem. For example, in a desktop photo search, users usually provide a location or event name in the folder name. Or, in a Web image search, we can choose one of a Web image's surrounding keywords as the query keyword.

We focus on the second step in this paper and propose the so-called AnnoSearch system. Its inputs

[1] The work was done when Xin-Jing Wang was an intern in Microsoft Research Asia. Now she is with IBM China Research Lab in Beijing and her contact email is wangxinj@cn.ibm.com.

	I was taking a picture of the rose, when this little fellow buzzed in. I thought he was more interesting than the rose. All comments welcome
	Another autumn photo from city park... early morning
	Spring into that Swiss mountains, in the proximity of San Bernadino, comment welcom, thanks albauer

Figure 1. Example Images and Descriptions of the Database

are the image to be annotated and a keyword which describes a concept of this image. Given this keyword, the semantic gap is by-passed to a certain degree thus the annotation problem is more "solvable".

Given the inputs, text-based retrieval is conducted on a large-scale Web image database in which images are associated with textual descriptions (see Figure 1). Because keywords may be ambiguous, e.g. both "tiger lily" and "white tiger" are relevant to query "tiger", content-based search is applied on the retrieved images to ensure visual similarity as well, and rank them accordingly. This step is accelerated by mapping the visual features to hash codes (refer to Section 3.3.1). As the last step, descriptions of the top *N* ranked images are clustered and the key concepts learned from the top ranked clusters are output as the predicted annotations, and the entire process ends.

A notable advantage of our approach is that no supervised training process is adopted, and as a direct result, our method can handle unlimited vocabulary, which is apparently superior to the previous works. It also ensures a highly scalable image database, although we have already used 2.4 million images. Moreover, the newly annotated images can be directly fed into the database to cover more image concepts, and images inside the database can also reinforce their annotations with each other. Hopefully when the database is large enough, no keyword input is required as in the ideal case discussed above. We make these as our future works.

The paper is organized as follows. Section 2 discusses several related works. Section 3 presents the AnnoSearch system in detail and Section 4 gives experimental results. We bring several discussions in Section 5 and conclude this paper in Section 6 with some outlooks of possible improvements.

2. Related works

Relevance feedback is used to annotate images in some early works [16]. Recent approaches mainly work on two directions. One finds more representative features to model the objects, e.g. Duygulu et al. [5] represent images by a group of blobs, and then use statistical machine translation model to translate the blobs into a set of keywords. The other uses machine learning techniques to learn the joint probabilities [1][2][4][5][7] or correlations [10][11][12] between images and keywords. A notable work is presented by Jeon et al.[10]. They used about 56,000 Yahoo! news images with noisy descriptions

All these previous works require a supervised training stage to learn prediction models which limits their vocabulary. Moreover, most of them use manually labeled training image descriptions.

In recent years, some researchers began to leverage Web-scale data for image understanding [6][14]. An interesting work was proposed by Yeh et al. [14] which identifies locations by searching the Internet. Given a picture of an unknown place, they firstly obtain a small number of visually relevant Web images using content-based search. Then a few keywords are extracted from the descriptions of these images and text-based search is performed whose results are further filtered by visual features. The disadvantages of [14] are that due to the efficiency problem, only a small number of relevant images can be retrieved as seeds which possibly degrades the performance. And the semantic gap problem will inevitably bias the final results.

3. The AnnoSearch system

In this paper, we reformulate the image auto-annotation problem as searching for semantically and visually similar images on the Web, and mining common concepts from the descriptions of the retrieved images.

The framework of this system is shown in Figure 2. It contains three stages: the text-based search stage, the content-based search stage and the annotation learning stage, which are differentiated using different colors (black, brown, blue) and labels (①, ②, ③).

3.1. Inspirations of the proposed idea

This section provides some insights into the image auto-annotation problem, which directly inspired our idea of the AnnoSearch system.

Fundamentally, the aim of image auto-annotation is to find a group of keywords $\mathbf{w}^*$ that maximizes the conditional distributions $p(\mathbf{w} | I_q)$, where I_q is the uncaptioned image and $\mathbf{w}$ keywords in the vocabulary,

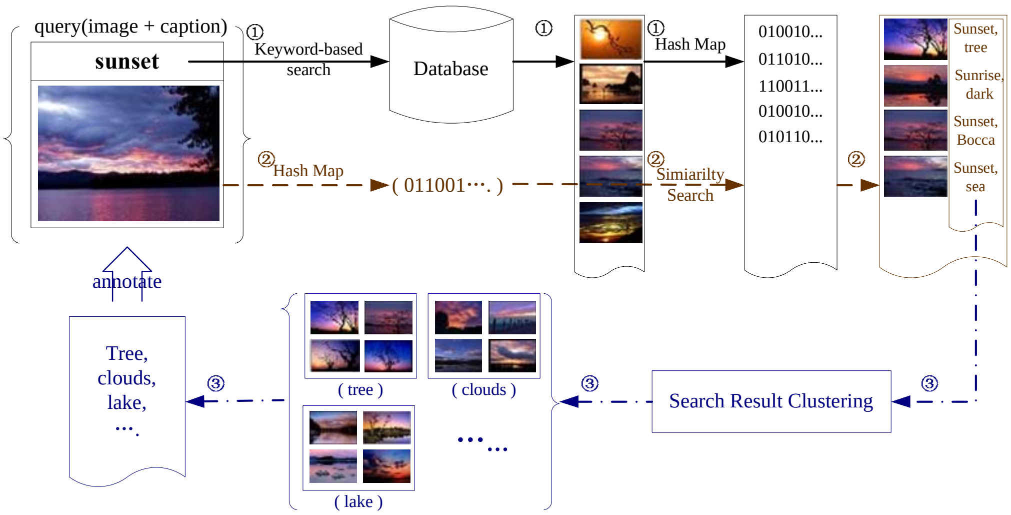


Figure 2. Framework of AnnoSearch System.

as formulated in Eq.1. Two types of approaches can be used, as indicated by (*) and (**) respectively in Eq.1.

$$\begin{aligned}\mathbf{w}^* &= \arg\max_{\mathbf{w}} p(\mathbf{w} \mid I_q) \\ &= \arg\max_{\mathbf{w}} \sum\nolimits_i p(\mathbf{w} \mid I_i) \cdot p(I_i \mid I_q) \qquad \cdots (*) \qquad (1) \\ &= \arg\max_{\mathbf{w}} \sum\nolimits_i \left(\sum\nolimits_j p(\mathbf{w} \mid t_j) \cdot p(t_j \mid I_i) \right) \cdot p(I_i \mid I_q) \quad \cdots (**)\end{aligned}$$

(*) corresponds to a two-layer model which directly learns the projections from images to keywords. Most of the previous works which learn classifiers or joint probabilities between images and keywords belong to this type. The key problem is to learn $p(\mathbf{w} \mid I_i)$, where I_i denotes the i-th image in the training dataset. And $p(I_i \mid I_q)$ is just like an index to locate $\mathbf{w}$.

(**) corresponds to a three-layer model including “keywords”, “topics”, and “images”, and “topics” (t_j in Eq.1) is the hidden layer. Previous works [1][2] show that model (**) generally outperforms model (*) due to the more exhaustive investigation of object relationships.

However, previous approaches that use model (**) generally assume certain distributions of exponential family on images and texts, and the goal is to optimize the distribution parameters. Since such distributions may be far from the intrinsic ones, the performance is doomed.

Now suppose we view this model from an angel of search and mining process, we can then reformulate Eq.1 as Eq.2, where $\mathbf{\Theta}_q := \bigcup_i I_i^{(q)}$ denotes the set of images relevant to the query I_q and $p(\Theta_I \mid I_q)$ simulates the search process. Now the goal is to find a few keywords $\mathbf{w}$ that best represent the concept of dataset $\mathbf{\Theta}_q$, i.e. $p(\mathbf{w} \mid \mathbf{\Theta}_q)$. Moreover, images in $\mathbf{\Theta}_q$ may have multiple topics $\mathbf{t}$, so we can mine the topics (i.e. $p(\mathbf{t} \mid \mathbf{\Theta}_q)$) and use the most representative ones ($p(\mathbf{w} \mid \mathbf{t})$) to annotate the query image I_q.

$$\begin{aligned}\mathbf{w}^* &= \arg\max_{\mathbf{w}} p(\mathbf{w} \mid I_q) \\ &= \arg\max_{\mathbf{w}} p(\mathbf{w} \mid \mathbf{\Theta}_q) \cdot p(\mathbf{\Theta}_q \mid I_q) \qquad (2) \\ &= \arg\max_{\mathbf{w}} \left[p(\mathbf{w} \mid \mathbf{t}) \cdot p(\mathbf{t} \mid \mathbf{\Theta}_q) \right] \cdot p(\mathbf{\Theta}_q \mid I_q)\end{aligned}$$

Based on these analyses, the model of our AnnoSearch system corresponds to Eq.3 below. A query keywords $\mathbf{w}_q$ are provided along with the query image I_q to by-pass the semantic gap, and select semantically and visually relevant image set $\mathbf{\Theta}^*$. A clustering technique [15] is used to mine $p(\mathbf{t} \mid \mathbf{\Theta}^*)$. And $\mathbf{w}^*$ is given by the most representative topics $\mathbf{t}^*$ learnt from the most relevant image subsets $\mathbf{\Theta}^{**}$.

$$\begin{aligned}p(\mathbf{w}^* \mid I_q, \mathbf{w}_q) &= p(\mathbf{w}^* \mid \mathbf{\Theta}^*) \cdot p(\mathbf{\Theta}^* \mid I_q, \mathbf{w}_q) \\ &= \left[p(\mathbf{w}^* \mid \mathbf{t}) \cdot p(\mathbf{t} \mid \mathbf{\Theta}^*) \right] \cdot p(\mathbf{\Theta}^* \mid I_q, \mathbf{w}_q) \qquad (3) \\ &\approx p(\mathbf{w}^* \mid \mathbf{t}^*) \cdot p(\mathbf{t}^* \mid \mathbf{\Theta}^{**}) \cdot p(\mathbf{\Theta}^{**} \mid I_q, \mathbf{w}_q)\end{aligned}$$

3.2. Text-based search

Jeon et al. [10] recommend using high quality training data to learn prediction models as it affects greatly the annotation performance.

Hence in our approach, we collected about 2.4 million high-quality Web images associated with meaningful descriptions from online photo forums. These descriptions capture the corresponding images’ contents to certain degrees, as shown in **Figure 1**.

3.3. Content-based search

Because visual features are generally of high dimensional, similarity-oriented search based on visual features is always a bottleneck for large-scale image database retrieval on search efficiency. To overcome this problem, we adopt a hash encoding algorithm [3] to speed up this procedure.

3.3.1. Mapping visual features to hash codes. Suppose that visual features are mapped into bit streams, with higher bits representing more important contents of an image, we can speed up the search process by comparing only the value of higher bits of images.

This idea is proposed in [3] which proposes to encode image visual features to so-call hash codes. In [3], images are divided into even blocks and average luminance of each block is extracted as visual features. These features are transformed by a PCA mapping matrix learned beforehand, and then quantized into hash codes. The quantization strategy is that if a feature component is larger than the mean of this vector, it is quantized to 1, otherwise to 0. The efficiency of this approach is tested on a computer with a Dual Intel Pentium 4 Xeon hyper-threaded CPU and 2G memory. And Hamming distance based on the higher 12 bits of the hash codes are measured. It costs about 0.2 second to identify all the duplicate images in a database of 50,000 images, which is very fast.

We use 36-bin color Correlogram [9] as the original visual feature, which is widely used in content-based image retrieval, and map them to 32-dimension hash codes leveraging the algorithm proposed in [3].

3.3.2. Hash code-based Image retrieval. Four distance measures are proposed and compared.

1) *Hash code filtering plus Euclidean distance measure.* As discussed in Section 3.3.1, the higher bits of the hash codes contain the majority of energy of an image. Hence if the higher bits of two hash codes match, possibly they are more similar than only lower bits match. This measure is proposed based on these analyses. Images whose higher *n* bits of hash codes match exactly those of the query image are kept, and then ranked according to Euclidean distances based on Correlogram features. In our experiments, *n* = 20.

2) *Hamming distance.* It measures the number of different bits of two hash codes.

3) *Weighted Hamming distance.* Intuitively, since higher bits are more important, difference in higher bits should be larger-weighted. This measure evenly separates the 32-bit hash codes into 8 bins, and weights the corresponding Hamming distance by $2^{8-i}, 1 \le i \le 8$ for the *i*-th bin.

4) *Euclidean distance on color Correlograms.* We use this measure as a baseline to assess the effectiveness of the hash code based methods.

3.4. Learning annotations by clustering

In this section, we detail the annotation learning process, i.e. the solution to $p(\mathbf{w}_c \mid \mathbf{t}^*) \cdot p(\mathbf{t}^* \mid \mathbf{\Theta}^{**})$ in Eq.3. It is based on a clustering technique proposed by Zeng et al. [15] which suggests salient topics $\mathbf{t}^*$.

3.4.1. The Search Result Clustering algorithm [15]. The Search Result Clustering (SRC) algorithm [15] is an effective clustering technique that can generate clusters with highly readable names (i.e. t^* in Eq.3)· Distinct from previous clustering approaches, it clusters documents by ranking salient phrases. Given a number of documents, it extracts all possible phrases (n-grams) and calculates several properties for each phrase such as phrase frequencies, document frequencies, etc. Then a pre-learned regression model is applied to combine these properties into a single salience score. The top-ranked phrases are taken as the names of the candidate clusters, which are further merged according to their member documents. This method is more suitable for Web applications than other traditional clustering algorithms because it emphasizes the efficiency of identifying relevant clusters. The online demo can be accessed via http://wsm.directtaps.net/default.aspx.

3.4.2. Annotation prediction. We use SRC algorithm to cluster the retrieved semantically and visually relevant images according to their titles, URLs and surrounding texts. However, it is still an open problem to determine the optimal number of clusters for SRC as well as many well-known clustering algorithms, such as k-means. Hence we use Eq.4 to set the number of clusters $| n_{src} |$:

$$| n_{src} | = \max(| \mathbf{\Theta}^* | / 200, 4) \qquad (4)$$

where $| \mathbf{\Theta}^* |$ is the number of retrieved images. 200 is empirically selected because SRC algorithm tends to output highly salient names when the cluster size is large. On the other hand, if $| \mathbf{\Theta}^* |$ is too small, SRC algorithm tends to group all images in one or two clusters and hence images inside one cluster will cover multiple topics such that the learned cluster names are meaningless. This is a trade-off and we force the algorithm to output at least 4 clusters. Moreover, to ensure both the effectiveness and the efficiency, we set $\max|\Theta^*| = 2{,}000$.

We calculate a score for each cluster based on two strategies below respectively, and the names of the clusters whose scores exceed a certain threshold are extracted. After removing the duplicate words and phrases, we output them as the learned annotations.

The two scoring strategies evaluated are:

1) *Maximum cluster size criterion*. A cluster's score equals to the number of its member images. This is just the Maximum a Posteriori estimation (MAP). It assumes that the key concepts are the dominant ones.

2) *Average member image score criterion*. The average similarity of the member images to the query image is used as the score of the corresponding cluster. The reason is obvious: the more relevant the member images of a cluster are to the query, the more probably the concepts learned from this cluster represents the content of the query image.

4. Evaluation

A series of experiments were conducted to evaluate the effectiveness and efficiency of the AnnoSearch system.

We extracted 2.4 million photos from several online photo forums. They are of high quality and have rich descriptions, such as title, category and comments from the photographers. Though these descriptions are noisy, they cover to a certain degree the concepts of the corresponding photos, as shown in Figure 1. These photos make up of the database, from which the relevant images are retrieved to annotate the query image.

Two query datasets are used to evaluate the system performance. The first one is 30 Google images [8] of 15 categories (see Table 1) randomly selected. To evaluate the effectiveness of our approach, we deliberately used a few vague query keywords, e.g. we use "Paris" as the query keyword to annotate a photo of "Sacre Coeur". We manually assessed the retrieval results on this dataset.

Table 1: Queries from Goolge

Apple, Beach, Beijing, Bird, Butterfly, Clouds, Clownfish, Japan, Liberty, Lighthouse, Louvre, Paris, Sunset, Tiger, Tree

Table 2: Queries from U.Washington

Australia, Campus, Cannon beach, Cherries, Football, Geneva, Green lake, Indonesia, Iran, Italy, Japan, San juan, Spring flower, Swiss mountain, Yellowstone

The second dataset is a content-based image retrieval database downloadable from the University of Washington (UW) (http://www.cs.washington.edu/research/imagedatabase/groundtruth/). Images in this dataset have about 5 on average manually labeled ground truth annotations. And for many images, not all objects are annotated. In our evaluation, we stick to the UW ground truth annotations, i.e. synonyms or correct annotations that do not appear in UW annotations are assumed incorrect. The UW folder names are the query keywords (see Table 2).

4.1. Experiments on Google images

4.1.1. Evaluation criterion. Since no ground truth is available, we propose a strict evaluation criterion for this dataset.

For annotation systems, generally there are three types of predicted annotations: "perfect", "correct", and "wrong". "Perfect" annotations hit the main contents of an image, e.g. "rose" for the first image in Figure 1. "Correct" annotations are right ones but not perfect, e.g. "France" for an image of Eiffel tower. And "wrong" annotations are false positives. We believe that a comprehensive and effective evaluation criterion should differentiate these three types of results and thus proposed the criterion shown in Eq.5. It extends the *normalized score* measure [1], which only categorizes predictions into "right" or "wrong".

$$E = (p + 0.5\times r - w)/n \qquad (5)$$

n denotes the number of annotations predicted. p, r, w are the number of "perfect", "correct", and "wrong" annotations respectively. To emphasize the preference for "perfect" annotations, we punish the "correct" ones by lower weighting it (i.e. 0.5). In Eq.5, if all the predictions are "perfect", $E = 1$, while if all are wrong, $E = -1$.

4.1.2. System effectiveness. Figure 3 shows how the precision E varies vs. the similarity weight. This weight weights the average similarity of images retrieved after content-based search, and the resulted score is the threshold used to filter the irrelevant images. It determines $|\boldsymbol{\Theta}^*|$ in Eq.4 and directly affects the learned clusters and the predicted annotations. The reason of proposing such a threshold strategy is that, since the similarity of images varies greatly, it is very difficult to select a fixed threshold which promises satisfactory results for any queries.

The green square curves in Figure 3 represent the text-based method that no visual features are available. It serves as the baseline method and uses the maximum cluster size criterion to predict annotations.

Figure 3 (a) shows the performance with maximum cluster size criterion. The weighted Hamming distance measure performs the best. This is reasonable because it emphasizes the feature components that capture the important features of an image and de-emphasize the rest ones. It is interesting that Euclidean distance on Correlogram measure performs nearly the same of the Hamming distance measure. This shows that the information-loss due to PCA can be ignored on this dataset.

Another interesting result is that the hash code filtering plus Euclidean distance method performs badly. The reason is that the hash code generation method is too coarse thus the higher 20 bits of the hash codes for many irrelevant images are the same as those of the query image.

All the distance measures perform much better than the baseline method. This shows that requesting the visual similarity of voting images is valuable.

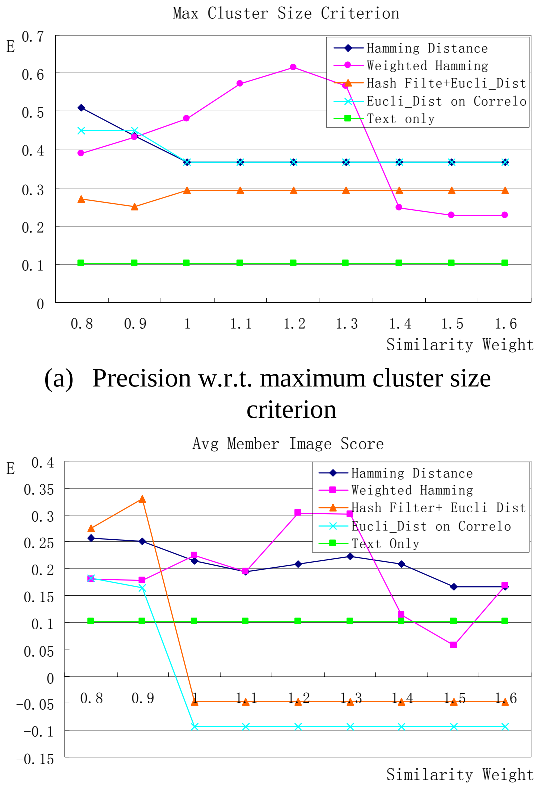
Figure 3 (b) shows the performance with maximum average member image score criterion. It is generally worse than that with the maximum cluster size criterion. A possible reason is the semantic gap. Recall that SRC algorithm clusters images based on their surrounding texts. Thus images in a cluster may have very different visual features even if they belong to the same category. This fact will not affect the maximum cluster size criterion but the average member image score criterion, because the latter uses visual similarity to score the clusters.

Note that the system performance drops rapidly when the threshold is too large so that $|\boldsymbol{\Theta}^*|$ is too small to ensure good clustering performance.

The first four rows in Figure 5 show a few examples of the AnnoSearch system's outputs. The boldfaced keywords are queries.

4.1.3. System efficiency. The efficiency of the four distance measures (image ranking procedure is included) was tested based on the 30 queries and 24,000 retrieved images on average. The hardware environment is a Dual Intel Pentium 4 Xeon hyper-threaded CPU and 2G memory.

The time cost is 0.034, 0.072, 0.051 and 0.122 seconds for Hamming distance measure, weighted Hamming distance measure, hash code filtering plus Euclidean distance measure, and Euclidean distance on Correlograms measure respectively. We can see that calculating Euclidean distances is nearly 4 times lower of calculating Hamming distance. The hash code filtering plus Euclidean distance measure is the second efficient. The reason is that hash code in this measure is as the inverted index which is very efficient to pick up similar images. The time is mainly cost by the Euclidean distance calculation afterwards.



(a) Precision w.r.t. maximum cluster size criterion

(b) Precision w.r.t. average member image score criterion

Figure 3. Annotation precision of the four distance measures vs. similarity weight.

Note that the above evaluations were conducted with all features loaded into memory. If disk access is required, we can imagine that hash code-based measures will be even faster than the original visual feature-based ones.

4.2. Experiments on UW dataset

Because ground truth is available and this dataset is not very large, we use precision and recall as the evaluation criteria. Figure 4 shows the maximum precision and recall of the two ranking strategies vs. the four distance measures. Again the weighted Hamming distance measure performs the best. An interesting

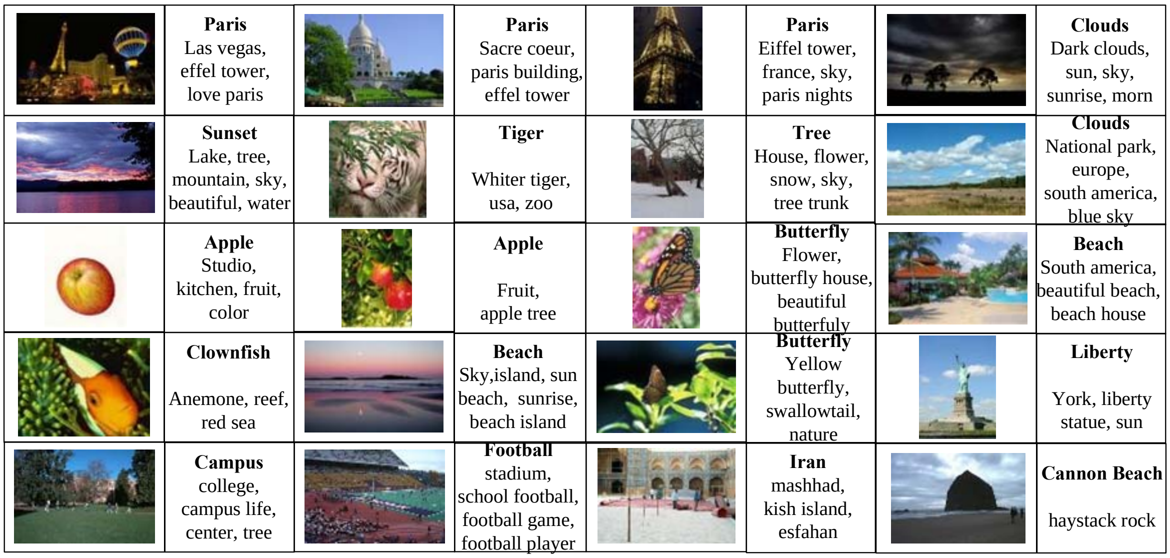
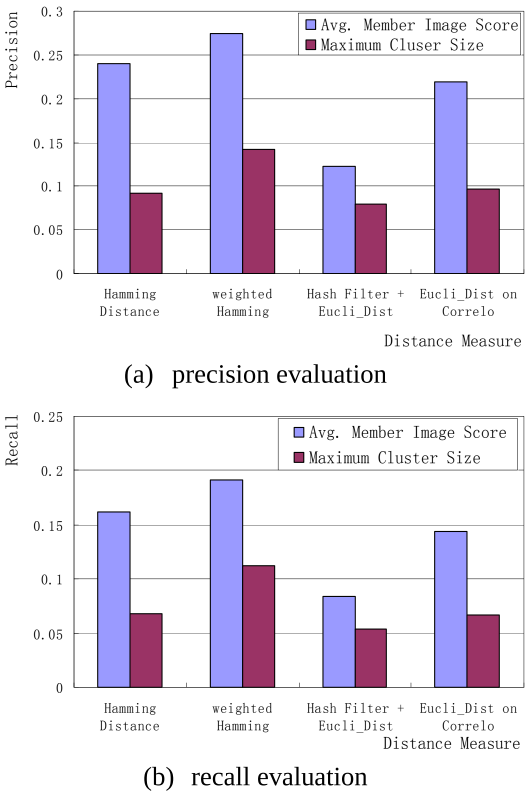

Paris
Las vegas, effel tower, love paris
Paris
Sacre coeur, paris building, effel tower
Paris
Eiffel tower, france, sky, paris nights
Clouds
Dark clouds, sun, sky, sunrise, morn
Sunset
Lake, tree, mountain, sky, beautiful, water
Tiger
Whiter tiger, usa, zoo
Tree
House, flower, snow, sky, tree trunk
Clouds
National park, europe, south america, blue sky
Apple
Studio, kitchen, fruit, color
Apple
Fruit, apple tree
Butterfly
Flower, butterfly house, beautiful butterfuly
Beach
South america, beautiful beach, beach house
Clownfish
Anemone, reef, red sea
Beach
Sky,island, sun beach, sunrise, beach island
Butterfly
Yellow butterfly, swallowtail, nature
Liberty
York, liberty statue, sun
Campus
college, campus life, center, tree
Football
stadium, school football, football game, football player
Iran
mashhad, kish island, esfahan
Cannon Beach
haystack rock

Figure 5. Examples Output by the AnnoSearch System.

point is that the average member image score criterion now works better. This is because few images in our 2.4 million photograph database matches the UW images. And the average image score strategy helps to rank higher the clusters whose member images are more relevant to the query, thus is less biased by the irrelevant member image descriptions.



(a) precision evaluation

(b) recall evaluation

Figure 4. Performance on UW dataset

It is worth mentioning that the real performance of our system will be much better than shown in Figure 4. As aforementioned, the current evaluation did not take synonyms, e.g. “beach” and “coast”, and semantically relevant keywords, e.g. “Geneva” and “Switzerland”, into consideration. Moreover, UW ground truth annotations may ignore some contents of an image, and our current evaluation assumes them as “wrong” even if the predicted annotation are correct but just do not appear in UW ground truth. To prove this, we examined the predicted annotations of 100 randomly selected queries. The corrected precision and recall are 38.14% and 22.95% respectively, nearly 12% precision improvement, with weighted Hamming distance measure, the average member image score criterion, and the similarity weight 1.2. The bottom four examples in Figure 5 show examples of “no hit” by the strict evaluation.

Moreover, because no supervised learning stage is included in our approach and the UW images do not match concepts of our database and hence few relevant images can be found, the task is much tougher for us than for the previous approaches. In most of the previous works, training data and testing data are selected from the same dataset and the training dataset is usually much larger than the testing dataset, e.g. [1][2] use 4,500 Corel images for training and 500 images for testing, and the performance is still around 20~30%. This shows that our system is more effective in predicting annotations, and is robust in handling outsiders.

5. Discussions

There are three major disadvantages in traditional computer vision or machine learning approaches for image auto-annotation: 1) they require a supervised learning process to learn the prediction models; 2) the vocabularies are generally small; 3) most of them use the Corel Stock-style database that images are well organized under semantic concepts with clean descriptions. Models learned on such data lacks generalization capability. The AnnoSearch system avoids all the disadvantages. It handles highly scalable vocabulary and is entirely unsupervised. It is also very robust to outsider queries.

6. Conclusions and future works

In this paper, we proposed a novel approach which reformulates the image auto-annotation problem as searching for semantically and visually similar images on the Web and mining annotations from their descriptions. It has three steps: 1) given a query keyword, a text-based search is performed to retrieve semantically similar images; 2) then given the query image, a content-based search is used to identify the images that are also visually similar; 3) at last, the selected images are clustered and salient phrases are extracted from their descriptions, from which the annotations are learned and assigned to the query image. To make it an online system, a hash coding algorithm is adopted to speed up the content-based search. Experiments are conducted on 2.4 million photo forum images that proved the effectiveness and efficiency of this proposed approach.

There is still a long way to go. The ultimate goal of image annotation is to process tens of thousands of images of various concepts precisely and efficiently, not a single query case. Hence in the future, we will work on reinforcing the labels of images inside a large-scale database. Moreover, we are interested to tackle the problem of how to annotate query images without any associated keywords.

7. References

[1] K. Barnard, P. Duygulu, N. Freitas, D. Forsyth, D. Blei, and M. Jordan, “Matching Words and Pictures”, *Journal of Machine Learning Research*, 2003, pp.(3):1107-1135.

[2] D. Blei, and M. I. Jordan, “Modeling Annotated Data”, *SIGIR*, 2003.

[3] B. Wang, Z.W. Li, M.J. Li, “Efficient Duplicate Image Detection Algorithm for Web Images and Large-scale Database”, *Technical Report of Microsoft Research*, 2005

[4] G. Carneiro, and N. Vasconcelos, “A Database Centric View of Semantic Image Annotation and Retrieval”, *SIGIR*, 2005.

[5] P. Duygulu, K. Barnard, N. Freitas, and D. Forsyth, “Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary”, *ECCV*, 2002, pp. 97-112

[6] X. Fan, X. Xie, Z. Li, M. Li, and W.-Y. Ma, “Photo-to-Search: Using Multimodal Queries to Search the Web from Mobile Devices”, *7th ACM SIGMM Workshop on MIR*, 2005.

[7] A. Ghoshal, P. Ircing, and S. Khudanpur, “Hidden Markov Models for Automatic Annotation and Content-Based Retrieval of Images and Video”, *SIGIR*, 2005.

[8] Google image. images.google.com. 2005.

[9] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, “Image Indexing Using Color Correlograms”, *CVPR*, 1997

[10] J. Jeon, and R. Manmatha, “Automatic Image Annotation of News Images with Large Vocabularies and Low Quality Training Data”, *ACM Multimedia*, 2004

[11] V. Lavrenko, R. Manmatha, and J. Jeon, “A Model for Learning the Semantics of Pictures”, *NIPS*, 2003.

[12] B.T. Li, K. Goh, E. Chang. “Confidence-based Dynamic Ensemble for Image Annotation and Semantics Discovery”, *ACM Multimedia*, 2003, pp. 195-206.

[13] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu, “GCap: Graph-based Automatic Image Captioning”, *International Workshop on Multimedia Data and Document Engineering*, 2004.

[14] T.Yeh, K.Tollmar,T.Darrell, “Searching the Web with Mobile Images for Location Recognition”, *CVPR*, 2004, pp. 76-81

[15] H.J. Zeng, Q.C. He, Z. Chen, and W.-Y. Ma, “Learning To Cluster Web Search Results”. *SIGIR*, 2004, pp. 210-217.

[16] W.Y. Liu, S. Dumais, Y. Sun, H. Zhang, M. Czerwinski, and B. Field. “Semi-Automatic Image Annotation”, *HCI*, 2001, pp. 326-333