

Image Annotation by Large-Scale Content-based Image Retrieval*

Xirong Li², Le Chen², Lei Zhang¹, Fuzong Lin², Wei-Ying Ma¹

¹Microsoft Research Asia, 49 Zhichun Road, Beijing 100080, China

²Department of Computer Science & Technology, Tsinghua University, Beijing 100084, China

{lxi, chenle02}@mails.tsinghua.edu.cn, linfz@mail.tsinghua.edu.cn

{leizhang, wyma}@microsoft.com

ABSTRACT

Image annotation has been an active research topic in recent years due to its potentially large impact on both image understanding and Web image search. In this paper, we target at solving the automatic image annotation problem in a novel *search* and *mining* framework. Given an uncaptioned image, first in the *search* stage, we perform content-based image retrieval (CBIR) facilitated by high-dimensional indexing to find a set of visually similar images from a large-scale image database. The database consists of images crawled from the World Wide Web with rich annotations, e.g. titles and surrounding text. Then in the *mining* stage, a search result clustering technique is utilized to find most representative keywords from the annotations of the retrieved image subset. These keywords, after salience ranking, are finally used to annotate the uncaptioned image. Based on search technologies, this framework does not impose an explicit training stage, but efficiently leverages large-scale and well-annotated images, and is potentially capable of dealing with unlimited vocabulary. Based on 2.4 million real Web images, comprehensive evaluation of image annotation on Corel and U. Washington image databases show the effectiveness and efficiency of the proposed approach.

Categories and Subject Descriptors

I.4.8 [Image Processing and Computer Vision]: Scene Analysis - Object recognition, H.2.8 [Database Management]: Database Applications – Image databases.

General Terms

Algorithms, Measurement, Experimentation

Keywords

Automatic Image Annotation, Similarity Search, Result Clustering

1. INTRODUCTION

Automatic image annotation has received broad attentions in recent years. Given a set of annotated images as training data, many methods have been proposed in the literature to find most representative keywords to annotate an uncaptioned image.

Besides for object recognition, which try to understand a very limited number of objects in images, most works about image annotation focus on learning a mapping (e.g. translation, joint probability and image classification) between images and words given a number of training images [1,2].

For example, Duygulu et al. [3] proposed a translation model to label images at region level under the assumption that each blob in a visual vocabulary can be interpreted by certain word in a dictionary. The latent Dirichlet allocation model and the hierarchical aspect model were investigated in [1]. Jeon et al. [4] proposed cross-media relevance model to predict the probability of generating a word given the blobs in an image. In the scenario that each word is treated as a distinct class, image annotation can be viewed as multi-class classification problem. Yang et al. [6] use multiple-instance learning to identify particular keywords from image data using labeled bags of examples. The basic intuition is to learn the most representative image region for a given keyword.

However, compared with the potentially unlimited vocabulary existing in the Web-scale image databases, only a very limited number of concepts can be modeled on a small-scale image database by learning projections or correlations between images and keywords. By leveraging numerous Web pages, search technology oriented approaches seem to be a promising way to solve this problem. Wang et al. [5] proposed a search-based annotation system – AnnoSearch. This system requires an initial keyword as a seed to speed up the search by leveraging text-based search technologies. However, the initial keyword might not always be available in real environment. In the case there is no initial keyword available for the query image, the system will encounter a serious efficiency problem. Furthermore, the system tends to be biased by the quality of initial keywords. If the initial keywords are not accurate, the annotation performance will degenerate.

In this paper, we target at building a novel image annotation system which can efficiently and effectively leverage the large-scale Web images based on two key techniques: the proposed Multi-Index algorithm for indexing high-dimensional visual features, and the search result clustering (SRC) [7] technique.

The rest of the paper is organized as follows. In section 2, we describe the whole annotation system in detail. Experimental results are then reported in section 3. We conclude this paper and discuss some future works in section 4.

*This work was performed at Microsoft Research Asia.

2. ANNOTATION SYSTEM

2.1 Problem Formulation

The intention of image annotation is to find a keyword set w^* that maximizes the conditional probability $P(w|I_q)$, where w is a keyword in the vocabulary and I_q is an uncaptioned image, as indicated by (*) in Eq. 1. We reformulate this optimization problem from a search and mining perspective, that is,

$$w^* = \arg \max_w P(w|I_q) \quad (*) \quad (1)$$

$$= \arg \max_w \sum_{I_i \in \Phi} P(w|I_i)P(I_i|I_q) \quad (**)$$

where I_i denotes an image in a database Φ , in which each image has some textual descriptions, $P(I_i|I_q)$ denotes the probability that I_i is relevant (i.e. similar) to I_q , and $P(w|I_i)$ represents the likelihood that I_i can be interpreted by w .

Our algorithm is composed of two basic stages:

1. **Searching similar images.** For an uncaptioned image I_q , we first find a set of visually similar images Φ_s from a large-scale image database.
2. **Mining representative keywords.** Given the image set Φ_s , we further cluster the descriptive texts of Φ_s (i.e., image title, surrounding text, etc.) to find the most representative keywords as the annotations to I_q .

The system framework is shown in Figure 1.

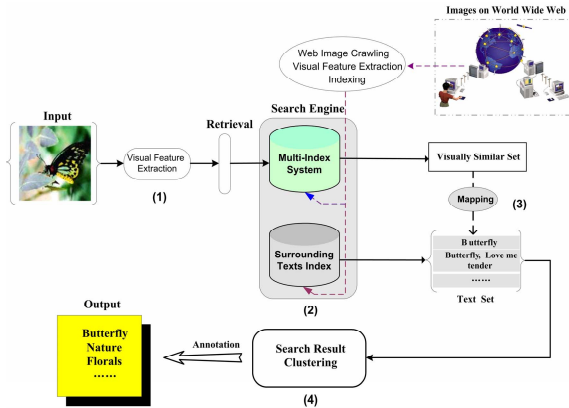


Figure 1. Framework of automatic image annotation system

2.2 System Implementation

2.2.1 High-Dimensional Indexing

Typically, an image can be represented by global feature (one point in a global feature space) or local feature (multiple points in a local feature space). In a small database, a simple sequential scan is usually employed for K nearest-neighbor (K -NN) search. However, if we want to exploit Web-scale, say millions or billions images, efficient indexing algorithms are imperative.

Unfortunately, high-dimensional indexing still remains a challenging problem in the database field. Existing methods for exact K -NN search are outperformed on average by sequential scanning if the number of dimensions increases [9], which is known as the curse of dimensionality. Fortunately, in many cases, such as multimedia applications, the precision of search results is

much more important than the recall rate when the database is large enough, e.g. millions or billions. In this case, an approximate yet efficient indexing technique can be employed.

Most existing algorithms for high-dimensional indexing can be viewed as certain forms of data-space partitioning, i.e., points are classified into blocks, which will be further indexed. However, all these methods suffer from the boundary problem. That is, when a query locates near a boundary, blocks sharing this boundary need to load to get better performance. Intuitively, heterogeneous indexing methods having different boundaries might be mutually beneficial to overcome the boundary problem to some extent and be able to achieve more accurate K -NN search with less disk I/Os. Based on the above observation, we propose the Multi-Index, a practical solution to index high-dimensional data (see Alg. 1).

```

Input
 $q$  : a query point,
 $K$  : number of approximate nearest neighbors of  $q$ ,
 $N$  : number of indexing systems,
 $\{\lambda_i | i=1, \dots, N, \lambda_i > 0, \sum \lambda_i = 1\}$  : disk I/O proportion parameters,
 $N_{IO}$  : number of disk I/O.
begin
   $S_c \leftarrow \phi$ 
  foreach  $i=1, \dots, N$  do
     $S_c^i \leftarrow$  {points retrieved from the  $i$ th index system
                by  $\lambda_i N_{IO}$  disk I/Os}
   $S_c \leftarrow S_c \cup S_c^i$ 
end
return  $K$  approximate nearest neighbors in  $S_c$ .

```

Algorithm 1. The Multi-Index Algorithm.

In current system, we adopt two indexing algorithms: K-means based indexing algorithm [11] and VA⁺-File [9]. To evaluate the effectiveness of the Multi-Index algorithm, we use the relative approximation error metric E [11] for a set of queries Q as

$$E = \frac{1}{|Q|} \sum_{q \in Q} \frac{\sum_{i=1}^K \|q - a_i\|^2}{\sum_{j=1}^K \|q - r_j\|^2}$$

where (r_1, \dots, r_k) are the ground truth K -NN results while (a_1, \dots, a_k) are the approximate ones. The evaluation is shown in Figure 2, where K denotes K-means based indexing, V denotes VA⁺-File and K+V denotes using both indexing algorithms, which is the Multi-Index algorithm. The proportion of No. disk I/Os² is 9(K-mean):1(VA⁺-File), which is determined empirically. The experiment was conducted on the 2.4M image dataset (see Section 3).

It can be seen that, keeping the same disk I/O expense, the Multi-Index (K+V) outperforms both K and V. The Multi-Index solution somewhat resembles LSH [10] as they both use many indices. However, any kind of indexing methods can be encompassed in the Multi-Index system, instead of a specific family of hashing functions. Specifically, when heterogeneous indexing strategies are adopted (e.g. K-means plus VA⁺-File), we get better performances, while keeping a relatively low storage cost. It is worth noting that the high-dimensional indexing is disk-based, thus the disk I/O number rather than the CPU time is used to evaluate the efficiency. Our annotation system, indexing about 2.4M images, is implemented on a computer with a Dual Intel

² The number of disk I/Os is counted as the number of disk blocks to load. In current experiment, each query point is represented by 128 bytes and each disk block is 8 KB. Then each block can store $(8*1024)/128=64$ points.

Pentium 4 Xeon hyper-threaded CPU and 2G memory. Experimental results show that the indexing scheme is very efficient. On average, it takes 0.5 second to annotate one image.

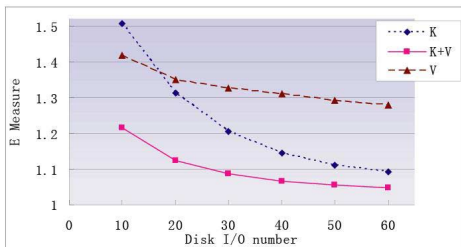


Figure 2. Performance Evaluation of Multi-Index

2.2.2 Annotation Prediction

We cluster images based on their surrounding text using SRC³ [7]. Different from traditional clustering approaches, SRC clusters documents by ranking salient phrases. It first extracts salient phrases and calculates several properties, such as phrase frequencies, and combines the properties into a salience score based on a pre-learned regression model. As SRC is capable of generating highly readable cluster names, these cluster names could be used as candidate annotation keywords. By further merging and pruning candidate clusters that are largely overlapped, top-ranked phrases are used as the final annotations.

In SRC, the number of clusters $|c|$ is automatically calculated by $|c| = \max(|\Phi_s|/200, 5)$. Since Φ_s is collected by similar image retrieval, we propose the following control factor to ensure the visual consistency in Φ_s while avoiding bringing in much noise:

$$\lambda = \frac{\text{Similarity (Bottom Ranked Image, Query Image)}}{\text{Similarity (Top Ranked Image, Query Image)}}$$

λ can be viewed as a tradeoff parameter between image quantity and quality, which is empirically set to 0.8 in this system.

3. EXPERIMENTS

We conducted two experiments to evaluate the proposed algorithm. First, to get a fair and objective evaluation, we only indexed 4500 Corel⁴ images and compared our algorithm with several previous works [3,4,6] on this dataset. Since our goal is to leverage large-scale image database and annotate images with unlimited vocabulary, we further indexed 2.4M images and performed more comprehensive experiments on the U. Washington image database⁵ (UW) to verify the effectiveness of the system.

3.1 Model Comparison

We compared the proposed search algorithm with three models, i.e., the machine translation model (MT) [3], the multi-instance learning model (MIL) [6] and the cross-media relevance model (CMRM) [4]. The dataset used by the three models is the common Corel database, which consists of 5,000 images, with 371 keywords and 500 blobs overall. We directly adopted the features used in [3,4,6] so that the four algorithms have the same input. In

the search system, blobs in each image are treated as visual words, thus similar images can be obtained through visual document retrieval. The ranking function adopted here is BM25 [12].

We used the same dataset partitioning as [3], i.e. 4500 images for training and the remaining 500 images for testing. The partitioning is also consistent with [4,6]. The average per-word precision and recall is reported in Table 1 for the best 49 keywords as [4,6] did. Table 1 shows clearly that the search approach outperforms both MT and MIL, and is comparable with CMRM, though it is not specifically proposed for this dataset.

Table 1. Performance comparisons of the four models

Models	MT	MIL	CMRM	Search
Avg. precision	0.20	0.31	0.41	0.39
Avg. recall	0.34	0.46	0.49	0.49

3.2 Experiments on UW Database

To obtain a well-annotated image database, we crawled 2.4M images from several photo forum sites, because images in photo forums have rich and accurate descriptions provided by photographers. In this work, a 64-dimensional global feature [8] are extracted and then indexed by the Multi-Index system. It is worth noting that global features are helpful for image-level concept annotation, but are ineffective in object-level annotation. We will investigate how to index local features and define local similarity in our future work. We used the UW database as the test dataset, which consists of 1109 images and 351 unique words.

To evaluate the performance in an intuitive way, we employed two criteria, i.e. precision (p) and recall (r).

$$p = \frac{1}{n} \sum_{k=1}^n \frac{\text{number of correctly annotated words in a test image } I_k}{\text{number of annotated words in } I_k}$$

$$r = \frac{1}{n} \sum_{k=1}^n \frac{\text{number of correctly annotated words in a test image } I_k}{\text{number of ground truth words in } I_k}$$

Then the performance on Corel dataset using these new criteria is $p=0.19$ and $r=0.44$.

Figure 3 shows the performances with different $|\Phi_s|$. It can be seen that the performance improves when $|\Phi_s|$ increases from 500 to 2000. But when $|\Phi_s|$ exceeds 2000, the performance can hardly improve. This implies that more images may bring more noises, and thus may degenerate the performance. It also shows that a better similarity measure (e.g. local similarity measure) is needed to obtain a more semantically relevant image set.

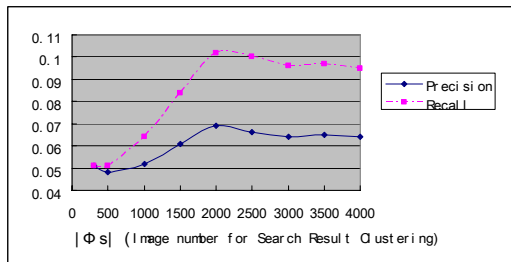


Figure 3. Performances with different $|\Phi_s|$

Note that the performance on the whole UW database is not satisfactory. One possible reason is that the evaluation is done by exact keyword matching, hence keywords outside the ground truth yet meaningful are simply excluded. Moreover, predicting annotations with an unlimited vocabulary, which is a significant

³Please refer the online system: <http://wsm.directtaps.net/default.aspx>

⁴http://www.cs.arizona.edu/people/kobus/research/data/eccv_2002/

⁵<http://www.cs.washington.edu/research/imagedatabase/groundtruth/>

advantage of this annotation system benefited from Web-scale data, can hardly be evaluated in this rigid way. Therefore, we further conducted the following experiments with evaluation by human check, where $|\Phi_s|$ is set no larger than 2000.

Due to the expense of human check, 50 images are randomly selected from the UW dataset. Note that this subset is selected without bias because the performance on it (AE-Web in Figure 4) is consistent with that on the whole dataset ($p=0.07$, $r=0.10$).

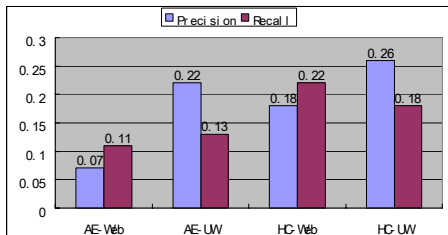


Figure 4. Performance evaluation on 50-image subset

Using auto-evaluation result (AE-Web) as a baseline, human check shows that the actual precision of the annotation system (HC-Web) is 0.18 and the actual recall is 0.22, which are 157.1% and 100% improvements. We further restricted the annotation vocabulary to UW dataset, that is, only words included in the UW vocabulary are allowed to annotate. It can be seen that both precision and recall for auto-evaluation (AE-UW) and human-check (HC-UW) are significantly improved with this constraint. This implies that higher performance is easily achieved in a closed dataset with a fixed vocabulary. Among all these results, HC-Web has the highest recall, which is benefited from the large-scale data. Meanwhile, noisy or irrelevant words may also be predicted as annotations, resulting in some drop in precision.

An interesting result is that there are 9 new words which are examined as correct annotations yet outside the UW vocabulary, such as city, summer and rose. In contrast, the randomly selected 50 images have a total of only 73 unique words in the ground truth. This result further shows the strength of the annotation system, that is, higher recall and larger vocabulary in annotation.

Four illustrative annotation results are given in Figure 5. For more results, please refer to the online demonstration⁶.

3.3 Discussion: Closed dataset vs. Open dataset

Observe that the performance on Corel dataset ($p=0.19$, $r=0.44$) is much better than that on the UW dataset ($p=0.18$, $r=0.22$). It is worth noting that these two experiments are conducted in totally different scenarios. The Corel experiment is based on a closed dataset, as 90% of the whole set are used for training and the remaining 10% for testing. The strong correlation between training set and testing set leads to good performances but this is less practical in real systems. The experiments in Section 3.2 show that the reported performances are usually higher on a fixed annotation vocabulary than that on an open one. While for UW data, we treat it as a black box in the system. Though the precision and recall is relatively low, this experiment is much closer to the real scenarios. Annotating an open dataset without any prior knowledge is obviously a very challenging problem.

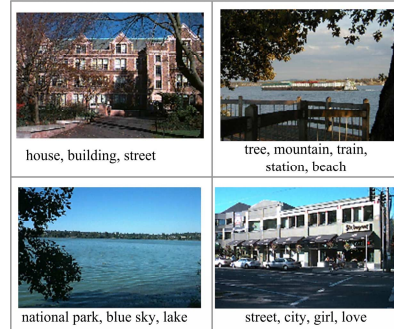


Figure 5. Some illustrative annotations

4. CONCLUSIONS & FUTURE WORKS

In this paper, we have presented a practical and effective image annotation system. We formulate the image annotation as searching for similar images and mining key phrases from the descriptions of the resultant images, based on two key techniques: Multi-Index – a practical solution to index 2.4M Web image database and SRC – the search result clustering technique.

To the best of our knowledge, this work is the first endeavor to annotate an open dataset without any prior knowledge by leveraging large-scale content-based image retrieval. Comprehensive experiments have demonstrated the effectiveness of the proposed approach. Due to the limitations of the global features in the current system, the performance for images containing complex objects are not satisfactory. In the future, we will investigate how to leverage local features to improve the annotation quality.

5. REFERENCES

- [1] K. Barnard, P. Duygulu, D. Forsyth, N. Freitas, D. Blei, and M. Jordan. Matching words and pictures. *JMLR*, 2003.
- [2] G. Carneiro and N. Vasconcelos. A Database Centric View of Semantic Image Annotation and Retrieval. *SIGIR*, 2005.
- [3] P. Duygulu, K. Barnard, N. Freitas and D. Forsyth. Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. *ECCV*, 2002.
- [4] J. Jeon, V. Lavrenko and R. Manmatha. Automatic Image Annotation and Retrieval using Cross-Media Relevance Models. *SIGIR*, 2003.
- [5] X. Wang, L. Zhang, F. Jing and W. Ma. AnnoSearch: Image Auto-Annotation by Search. *CVPR*, 2006.
- [6] C. Yang, M. Dong and F. Fotouhi. Region Based Image Annotation Through Multiple-Instance Learning. *ACM MM*, 2004.
- [7] H. Zeng, Q. He, Z. Chen and W. Ma. Learning to cluster web search results. *SIGIR*, 2004.
- [8] L. Zhang, Y. Hu, M. Li, W. Ma and H. Zhang. Efficient Propagation for Face Annotation in Family Albums. *ACM MM*, 2002.
- [9] H. Ferhatosmanoglu, E. Tuncel, D. Agrawal and A. Abbadi. Vector Approximation Based Indexing for Non-uniform High Dimensional Data Sets. *CIKM*, 2000.
- [10] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. *VLDB*, 1999.
- [11] H. Ferhatosmanoglu, E. Tuncel, D. Agrawal and A. Abbadi. Approximate neighbor searching in multimedia databases. *ICDE*, 2001.
- [12] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu and M. Gattford. Okapi at TREC-3. *TREC-3*, 1995.

⁶ <http://202.108.85.220/MultiIndex/index.htm>