Content-Based Visual Search Learned from Social Media

Xirong Li

Printing: Off Page, Amsterdam

Cover: Svetlana Kordumova and Xirong Li

ISBN 978-94-6182-082-2

Content-Based Visual Search Learned from Social Media

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. D.C. van den Boom
ten overstaan van een door het college voor promoties
ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op vrijdag 9 maart 2012, te 12:00 uur

door

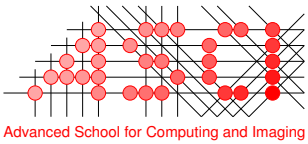
Xirong Li

geboren te Taizhou, China

Promotiecommissie

Promotor:	Prof. dr. ir. A.W.M. Smeulders
Co-promotor:	*Dr. M. Worring* *Dr. C.G.M. Snoek*
Overige Leden:	*Prof. dr. J.Z. Wang* *Prof. dr. M. de Rijke* *Prof. dr. A.T. Schreiber* *Dr. A. Hanjalic*

Faculteit der Natuurwetenschappen, Wiskunde en Informatica



This work was carried out in the ASCI graduate school, at the Intelligent Sensory Information Systems group of the University of Amsterdam. ASCI dissertation series number 253.

Intelligent Sensory Information Systems
University of Amsterdam
The Netherlands

Contents

Chapter 1

Introduction

All of a sudden digital images became social. In just a decade, individual and mostly inactive consumers have transformed into active and connected prosumers, revolutionaries even, who create, share, and comment on massive amounts of image artifacts all over the world wide web. Pronounced manifestations of social images on the Internet include industry initiatives like Facebook, Google, and Flickr, who manage to attract millions of users, daily. In order to make sense of the massive amounts of visual content, online social platforms rely on what people say is in an image, which is known to be ambiguous, overly personalized, and limited [39, 81], see Figure 1.1. Hence, the lack of semantics currently associated with social images is seriously hampering retrieval, repurposing, and usage.

For determining what people have said is factually in the image, and for professional archives which cannot be shared for crowdsourcing, multimedia content analysis is crucial. Despite good progress [108], automated multimedia analysis of visual content is still seriously hampered by the semantic gap, defined as "the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation" [106]. By its definition, the gap has both an objective and subjective aspect. The objective aspect refers to semantics inferred from the visual content, which different people would agree upon. For instance, whether a cow is visible in a picture is a matter of fact. The subjective aspect relates to the personal context of the user in the given situation. Users have personal associations with image subjects. For example, the same cow picture will have quite a different meaning for a farmer. Such information cannot be derived from the visual content alone. We argue that both aspects have to be addressed to fully bridge the semantic gap.

The semantic gap may be bridged using visual classifiers that automatically tag an image with visual concepts including people, objects, scenes, and events, with

Figure 1.1: Examples of socially tagged images. Note that social tags do not necessarily reflect the visual content of the images they are describing, and the majority of social images are untagged.

varying performance [85, 109]. Together with content-based image representations and kernel-based machine learning, labeled training examples play the crucial role in creating automated image taggers. Traditionally, the training examples are labeled by expert annotators. However, expert annotation is labor intensive and time consuming, making well-labeled examples expensive and their availability limited. The lack of high-quality training examples is limiting both the quality and quantity of auto-taggers.

A diverse group of people can act as a wise crowd when making decision, even outperforming experts, given that their options are diverse, independent, and properly aggregated [114]. Extrapolating this observation to the social web context, we hypothesize that despite the subjective tagging of individual users, when tags are properly aggregated they may truly reflect the visual content. In that sense, the deluge of socially tagged images seems the perfect source for next-generation image search. We aim to acquire high-quality training examples by exploiting socially tagged images, and reveal their value for image retrieval. So the fundamental question addressed in this thesis is:

What is the value of socially tagged images for visual search?

To answer the question, we structure the thesis into two parts: offline learning and online use. In Part I, we propose approaches which automatically harvest training examples from the social web. In Part II, we study how to use these examples for two applications: visual search and personalized image tagging, respectively addressing the objective and subjective aspect of the semantic gap.

1.1 Part I: Offline Learning

To learn a visual classifier for a target concept, we need both positive examples and negative examples. Due to the subjective nature of social tagging, images labeled with a target concept are not necessarily good positive examples. We seek positive examples which accurately describe the concept. In contrast to positive examples, negative examples are in the majority on the social web. Consequently, random sampling yields a set of genuine negative examples. But they are often not informative as they do not represent the difficult negatives confusing a classifier. In Part I, we study the value of socially tagged images as positive and negative training examples.

In order to obtain accurate positive examples from a large amount of socially tagged images, we have to determine which tags are relevant to the visual content. So the first question to answer is:

What determines the relevance of a social tag with respect to an image?

Given the large amount of socially tagged images and an equally large amount of tags used in social tagging, a lightweight approach which effectively estimates tag relevance is required. Intuitively, if different persons label visually similar images using the same tags, these tags tend to reflect objective aspects of the visual content. Tags describing feelings can also be objective, but only when those feelings are shared by the crowd rather than caused by personal experience. Starting from this intuition, we propose in **Chapter 2** a neighbor voting algorithm which estimates tag relevance by accumulating votes from visual neighbors.

In practice, visual concepts often vary significantly in terms of their appearance and in terms of their visual context. For instance, a boat can be a canoe, a sailboat, or any other type. Moreover, it may appear in water, on the beach, or even in a museum. It is unlikely that such large variations in the visual content can be described by a specific feature. Consequently, a tag relevance estimator using a single feature tends to be limited. Therefore, as an extension of tag relevance estimation, we consider the fusion of multiple tag relevance estimates. It has been recognized that for generic multimedia analysis, fusing multiple sources of evidence is beneficial [5]. In the new context of tag relevance estimation, given multiple tag relevance estimators driven by various visual features, our second question arises as:

How to fuse tag relevance estimators?

In **Chapter 3**, we study the tag relevance fusion problem. Depending on the level where the fusion is executed, we develop early and late tag relevance fusion schemes for the neighbor voting based estimator introduced in Chapter 2. We systematically study the characteristics and performance of the two fusion schemes.

For obtaining the negative examples, random sampling is the *de facto* standard in the literature. Since a classifier tends to misclassify the negative examples which are visually close to the positive examples, inclusion of such informative examples is important for classifier learning. However, they are unlikely to be hit by random sampling. Hence, the third question to answer is:

Which social images are informative negative examples?

In principle, informative negatives of a given concept will have visual patterns which may overlap the positive examples. Note that the informativeness of negative examples depends on the underlying visual features and on the underlying classifiers. Which negative examples are indeed information are not necessarily consistent with what an observer may expect. Hence, it is difficult to specify informative negative classes by hand-crafted rules. We are interested in approaches which automatically mine the informative negative examples from the social web. In **Chapter 4**, we study a social negative bootstrapping approach for mining informative negatives from socially tagged images.

1.2 Part II: Online Use

To extract rich semantics from the visual content of social images, methods which go beyond single-concept detectors are needed. In addition, since human interpretation of the visual content depends on a specific user, personalized image analysis is essential for solving the subjective part of the semantic gap. Given the accurate positive examples and informative negative examples obtained in Part I, we exploit them in Part II for detecting multiple concepts in unlabeled images, and detecting concepts in a personalized setting.

A user's query is often more complex than a single concept is able to represent, e.g., finding images showing a horse next to a car. Note that a single-concept detector is trained on typical examples of the concept, e.g., cars on a street for the car detector and horses on grass for the horse detector. Images with a horse and a car co-occurring may also have a characteristics appearance, where the individual concepts are not present in their common form. Visual search methods which combine single-concept detectors are mostly ineffective for complex visual searches. So the fourth question to investigate is:

How to exploit socially tagged images for complex visual searches?

In **Chapter 5**, we introduce the notion of bi-concepts as a retrieval method for two concepts directly learned from social data. We define a bi-concept as the co-occurrence of two visual concepts, where its full meaning cannot be inferred from one

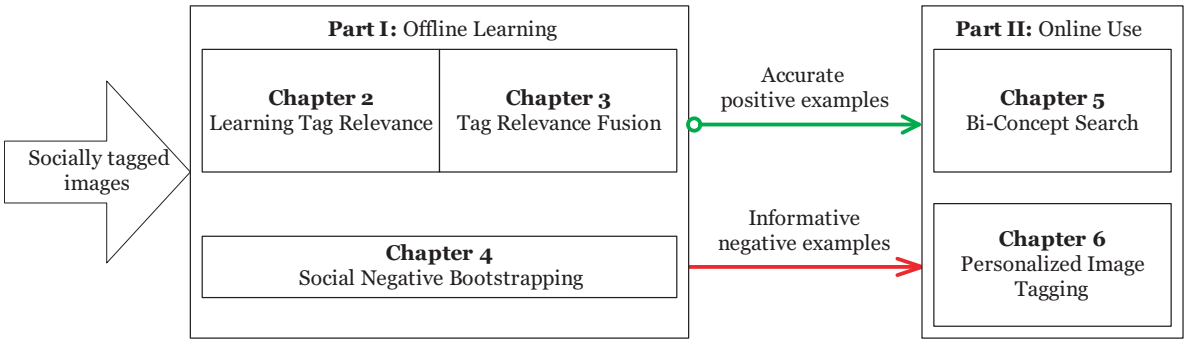


Figure 1.2: Structure of this thesis.

of its component concepts alone. As the number of potential bi-concepts is gigantic, manually collecting training examples is infeasible. Instead, we propose a multi-modal framework which integrates the algorithms proposed in Part I to collect an accurate set of positive examples and informative negative examples from the social web. We learn bi-concept detectors from these examples.

Clearly, personal preferences for image subjects vary from person to person. Some users may collect pictures of flowers, while others may favor photos of sport cars. This real-world phenomenon suggests that personal preferences have to be taken into account when creating an automated image tagger. The performance of present-day taggers is bound by the absence of personal information. More fundamentally, the difficulty is rooted in the subjective aspect of the semantic gap. So the fifth question to study is:

How to personalize automated image tagging with respect to a user's preference?

In **Chapter 6**, we aim for personalizing automated image tagging by jointly exploiting personalized tag statistics and content-driven taggers. We propose a cross-entropy based learning algorithm which personalizes a generic image tagging model by learning from a user's tagging history.

The schematic organization of the thesis is illustrated in Figure 1.2. Due to the dynamic nature of social media, its data distribution changes over time. This requires timely revision of trained models as they may not fit new data. The methods proposed in the thesis suit this nature, as they are automatical, and consequently updating models is just a matter of re-computation. By exploration of socially tagged images, we aim for next-generation image search which looks into the visual content, but without the need of dedicated manual labeling.

Part I: Offline Learning

Chapter 2

Learning Social Tag Relevance by Neighbor Voting

What determines the relevance of a social tag with respect to an image? We propose in this chapter a neighbor voting algorithm which estimates tag relevance by accumulating votes from visual neighbors. Under a set of well defined and realistic assumptions, we prove that our algorithm creates a good tag relevance estimator for both image ranking and tag ranking*.

*Published in *IEEE Transactions on Multimedia*, 11(7):1310-1322, 2009 [64].

2.1 Introduction

The advent of social multimedia tagging – assigning tags or keywords to images, music, or video clips by common users – is significantly reshaping the way people generate, manage, and search multimedia resources. Good examples are Flickr, which hosts more than 2 billion images with around 3 million new uploaded photos per day [6], and YouTube, which serves 100 million videos and 65,000 uploads daily [16]. Apart from their usage for general-purpose search, these rich multimedia databases are triggering many innovative research scenarios in areas as diverse as personalized information delivery [102], landmark recognition [55], concept similarity measurement [136], tag recommendation [12], and automatic image tagging [119, 125]. One would expect user-contributed tags to be a good starting point for all these applications.

Despite the success of social tagging, however, tags contributed by common users are known to be ambiguous, limited in terms of completeness, and overly personalized [39,81]. This is not surprising because of the uncontrolled nature of social tagging and the diversity of knowledge and cultural background of its users. Although the relevance of a tag given the visual content can be subjective for a specific user, an objective criterion is desirable for general-purpose search and visual content understanding. We consider a tag relevant to an image if the tag accurately describes objective aspects of the visual content, or in other words, users with common knowledge relate the tag to the visual content easily and consistently. Other tags are subjective or overly personalized and thus we consider those irrelevant, as illustrated in Figure 2.1. Apart from the fact that tags can be subjective, individual tags are mostly used once per image. This tagging behavior implies that given an image, relevant tags and irrelevant ones are not distinguishable by their occurrence frequency [62]. Hence, a fundamental problem in social image analysis and retrieval is how to accurately and efficiently learn the relevance of a tag with respect to the visual content the tag is describing.

Existing methods to automatically predict tag relevance with respect to the visual content often heavily rely on supervised machine learning methods [8, 18, 57]. In general, the methods boil down to learning a mapping between low-level visual features, e.g., color and local descriptors, and high-level semantic concepts, e.g., airplane and classroom. Since the number of training examples are limited for the supervised methods, the methods are not scalable to cover the potentially unlimited array of concepts existing in social tagging. Moreover, uncontrolled visual content contributed by users creates a broad domain environment having significant diversity in visual appearance, even for the same concept [106]. The scarcity of training examples and the significant diversity in visual appearance might make the learned models unreliable and difficult to generalize. Therefore, in a social tagging environment with large and diverse visual content, a lightweight or unsupervised learning method which effectively and efficiently estimates tag relevance is required.

Intuitively, if different persons label visually similar images using the same tags, these tags are likely to reflect objective aspects of the visual content. The intuition

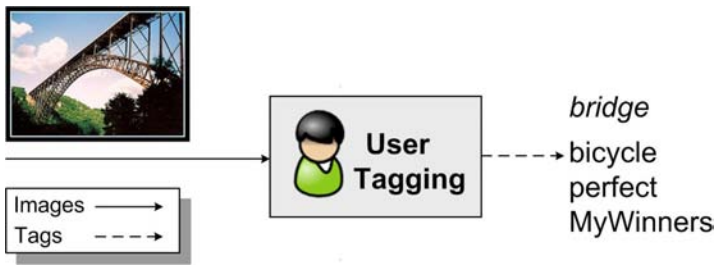


Figure 2.1: Dataflow of user tagging. According to whether a tag is relevant with respect to a given image, we divide user-contributed tags into two types, namely objective and subjective tags. The objective tags are marked by an *italic* font. In this example, tag *bridge* is objective, while the other three tags are subjective. We aim for automated approaches to learning tag relevance.

implies that the relevance of a tag with respect to an image might be inferred from tagging behavior of visual neighbors of that image. Starting from this intuition, we propose a novel neighbor voting algorithm for tag relevance learning. The key idea is, by propagating common tags through visual links introduced by visual similarity, each tag accumulates its relevance credit by receiving neighbor votes. Under a set of well defined and realistic assumptions, we prove that our tag relevance learning algorithm is a good measure for both image ranking and tag ranking. To demonstrate the viability of the proposed algorithm, we provide a systematic evaluation on 3.5 million Flickr images for both social image retrieval and image tag suggestion.

The rest of the chapter is organized as follows. We review related work in Section 2.2. We then describe in detail tag relevance learning in Section 2.3. We setup experiments in Section 2.4. Experimental results are presented in Section 6.5. We conclude the chapter in Section 6.6.

2.2 Related Work

We review work closely related to our motivation for tag relevance learning in the following two directions, that is, improving image tagging and improving image retrieval.

2.2.1 Improving Image Tagging

Depending on whether a target image is labeled, we categorize existing methods into two main scenarios, namely improving image tagging for labeled images and automated image tagging for unlabeled images.

In the first scenario, given an image labeled with some tags, one tries to improve image tagging by removing noisy tags [51], recommending new tags relevant to existing ones [12], or reducing tag ambiguity [133]. In [51] for instance, the authors assume

that the majority of existing tags are relevant with respect to the image. They then measure the relevance of a tag by computing word similarity between the tag and other tags. While in [12], the authors find new tags relevant with respect to the original ones by exploiting tag co-occurrence in a large user-tagged image database. To be precise, by using each of the original tags as a seed, they find a list of candidate tags having the largest co-occurrence with the seed tag. These lists are later aggregated into a single list and the top ranked tags are selected as the final recommendation. Since new tags are suggested purely using the initial tags, images with the same starting tags will end with the same new tags, regardless of the visual content. Hence, methods addressing both textual and visual clues are required.

Methods in the second scenario try to predict relevant tags for unlabeled images. We divide these methods according to their model-dependence into model-based and model-free approaches. The model-based approaches, often conducted in a supervised learning framework, focus on learning a mapping or projection between low-level visual features and high-level semantic concepts given a number of training examples [8, 18, 24, 57, 91]. Due to the expense of manual labeling, however, currently only a limited number of visual concepts can be modeled effectively. Besides, the approaches are often computationally expensive, making them difficult to scale up. Furthermore, the rapid growth of new multimedia data makes the trained models outdated quickly. To tackle these difficulties, a lightweight meta-learning algorithm is proposed in [26]. The gist of the algorithm is to progressively improve tagging accuracy by taking into account both the tags automatically predicted by an existing model and the tags provided by a user as implicit relevance feedback. In contrast to the model-based approaches, the model-free approaches attempt to predict relevant tags for an image by utilizing images on the Internet [63, 119, 125, 130]. These approaches assume there exists a large well-labeled database such that one can find a visual duplicate for the unlabeled image. Then, automatic tagging is done by simply propagating tags from the duplicate to that image. In reality, however, the database is of limited-scale with noisy annotations. Hence, neighbor search is first conducted to find visual neighbors. Disambiguation methods are then used to select relevant tags out of the raw annotations of the neighbors. In [119], for instance, the authors rank tags in terms of their frequency in the neighbor set. However, tags occurring frequently in the entire collection may dominate the results. To restrain such effects, the authors in [125] re-weight the frequency of a tag by multiplying this frequency by its inverse document frequency (idf). The idf value of a tag is inversely and logarithmically proportional to the occurrence frequency of the tag in the entire collection. Nonetheless, the idf scheme tends to over-weight rare tags.

To summarize, the existing methods for image tagging try to rank relevant tags ahead of irrelevant ones in terms of the tags' relevance value with respect to an image. However, since the tag ranking criterion is not directly related to the performance of image retrieval using the tagging results, optimizing image tagging does not necessarily yield good image rankings [40].

2.2.2 Improving Image Retrieval

Given unsatisfactory image tagging results, one might expect to improve image retrieval directly. Quite a few methods follow this research line, either by reranking search results in light of visual consistency [33, 45, 46, 52, 90, 139] or by expanding the original queries [10, 21, 72, 86]. We briefly review these methods in the following two paragraphs. For a more comprehensive survey, we refer to [27, 106].

Reranking methods assume that the majority of search results are relevant with respect to the query and relevant examples tend to have similar visual patterns such as color and texture. To find the dominant visual patterns, density estimation methods are often used, typically in the form of clustering [46,90] and random walk [52]. In [52] for instance, the authors leverage a random walk model to find visually representative images in a search result list obtained by text-based retrieval. To be precise, first an adjacent graph is constructed wherein each node corresponds to a result image and the edge between two nodes are weighted in terms of the visual similarity between the two corresponding images. A random walk is then simulated on the graph to estimate the probability that each node is visited. Since images in dense regions are more likely to be visited, the above probability is used to measure the representativeness of an image in the visual feature space and accordingly rerank the search results. However, density estimation is inaccurate when feature dimensionality is high and samples are insufficient for computing the density [101]. Besides, density estimation is computationally expensive. In [46] for example, the authors report an execution time of 18 seconds per search round, while a study on web users [84] shows the tolerable waiting time for web information retrieval is only 2 seconds, approximately. The difficulty in density estimation and the associated computational expense put the utility of reranking methods for social image retrieval into question.

Query expansion methods augment the original query by automatically adding relevant terms [10, 21, 72]. In [21], for instance, the authors use synonyms from a dictionary, whereas in [72] the authors select strongly related terms from text snippets returned by web search engines. Another example is [10], where the authors use clustering methods to find correlated tags. Though adding more query terms may retrieve more relevant results, how to choose appropriate expansion terms requires further research [11].

In summary, the reranking and query expansion methods try to rank images relevant with respect to a query ahead of irrelevant images. However, the methods leave the fundamental problem of subjective user tagging unaddressed.

Though we have witnessed great efforts devoted into improving both image tagging and image retrieval, the efforts are almost disconnected. Recent research, e.g., [15, 25, 43, 107], investigates the potential of leveraging automatic tagging results for image and video retrieval. To the best of our knowledge, however, up till now the solutions to the two problems are still separated, including our previous work [62, 63] which deal with social image tagging and social image retrieval, respectively. This work is an attempt to solve image ranking and tag ranking in a unified tag rele-

vance learning framework. In contrast to approaches for image ranking which are query-dependent, e.g., [52, 90], our algorithm is query-independent. This advantage allows us to run the algorithm offline without imposing extra waiting time on users. Further, by updating tag frequency with the learned tag frequency, we seamlessly embed visual information into current tag-based social image retrieval paradigms. For automatic image tagging, our algorithm shares similarities with the model-free approaches, e.g., [119, 125, 130], since they can be regarded as propagating tags between neighbor images. Note however that our algorithm is more general as it is applicable to both image retrieval and tagging. Moreover, we provide a formal analysis which is missing in previous studies.

2.3 Learning Tag Relevance by Neighbor Voting

In order to fulfill image retrieval, we seek a tag relevance measurement such that images relevant with respect to a tag are ranked ahead of images irrelevant with respect to the tag. Meanwhile, to fulfill image tagging, the measurement should rank tags relevant with respect to an image ahead of tags irrelevant with respect to the image. Recall the intuition that if different persons label visually similar images using the same tags, these tags are likely to reflect objective aspects of the visual content. This intuition suggests that the relevance of a tag given an image might be inferred from how visual neighbors of that image are tagged: the more frequent the tag occurs in the neighbor set, the more relevant it might be, as illustrated in Figure 2.2. However, some frequently occurring tags, such as '2007' and '2008', are unlikely to be relevant to the majority of images. Hence, a good tag relevance measurement should take into account the distribution of a tag in the neighbor set and in the entire collection, simultaneously. Motivated by the informal analysis above, we propose a neighbor voting algorithm for learning tag relevance, as depicted in Figure 2.2. Though the proposed algorithm is simple, we deem it important to gain insight into the rationale for the algorithm. The following two subsections serve for this purpose. Concretely, we first define in Section 2.3.1 two criteria to describe the general objective of tag relevance learning. Then, in Section 2.3.2 we provide a formal analysis of user tagging and content-based nearest neighbor search. We see how our algorithm is naturally derived from the analysis. Finally, we describe in detail the algorithm in Section 2.3.3.

2.3.1 The Objective of Tag Relevance Learning

We first introduce some notation for the ease of explanation. We denote a collection of user-tagged images as Φ and a vocabulary of tags used in Φ as W. For an image $I \in \Phi$ and a tag $w \in W$, let $r^*(w, I) : \{W, \Phi\} \mapsto \mathbf{R}$ be a tag relevance measurement. We call $r^*(w, I)$ an ideal measurement for image and tag ranking if it satisfies the following two criteria:

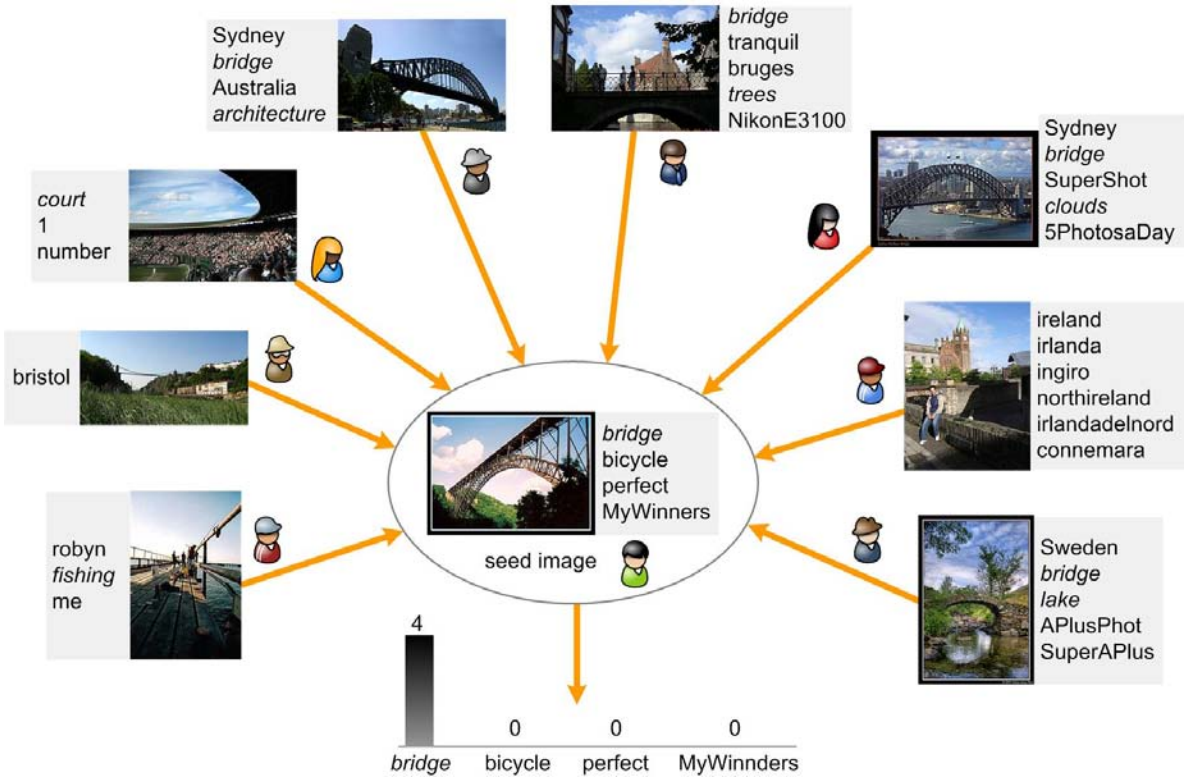


Figure 2.2: Learning tag relevance by neighbor voting. The tag relevance value of each tag is estimated by accumulating the neighbor votes it receives from visually similar images of the seed image. In this example, since four neighbor images are labeled with *bridge*, the tag relevance value of *bridge* with respect to the seed image is 4. Hence, we update the tag frequency of *bridge* from 1 to 4.

Criterion 1: Image ranking. *Given two images* $I_1, I_2 \in \Phi$ *and tag* $w \in W$*, if* w *is relevant to* I_1 *but irrelevant to* I_2*, then*

$$r^*(w, I_1) > r^*(w, I_2) \tag{2.1}$$

Criterion 2: Tag ranking. *Given two tags* $w_1, w_2 \in W$ *and image* $I \in \Phi$*, if* I *is relevant to* w_1 *but irrelevant to* w_2*, then*

$$r^*(w_1, I) > r^*(w_2, I) \tag{2.2}$$

Our goal is to find a tag relevance measurement satisfying the two criteria.

2.3.2 Learning Tag Relevance from Visual Neighbors

As aforementioned, given an image I labeled with a tag w, the occurrence frequency of w in visual neighbors of I to some extent reflects the relevance of w with respect

Table 2.1: Main notations defined in Chapter 2.

Notation	Definition
Φ	a collection of user-tagged images.
L_w	$L_w \subset \Phi$, all images labeled with tag w in the collection.
R_w	$R_w \subset \Phi$, all images relevant with respect to tag w in the collection.
R_w^c	$R_w^c = \Phi \backslash R_w$, all images irrelevant with respect to tag w in the collection.
$P(w\|R_w)$	probability of correct tagging, i.e., an image randomly selected from R_w is labeled with tag w.
$P(w\|R_w^c)$	probability of incorrect tagging, i.e., an image randomly selected from R_w^c is labeled with tag w.
$P(R_w)$	probability that an image randomly selected from the entire collection is relevant to tag w.
$P(R_w^c)$	probability that an image randomly selected from the entire collection is irrelevant to tag w.
f	a similarity function between two images, measured on low-level visual features.
$N_f(I,k)$	$N_f(I,k) \subset \Phi$, k nearest neighbors (k-nn) of an image I found in the collection by f.
$N_{rand}(k)$	$N_{rand}(k) \subset \Phi$, k images randomly selected from the collection.
$n_w[\cdot]$	an operator counting the number of tag w in any subset of the collection.

to I. Note that the neighbors can be decomposed into two parts according to their relevance to w, i.e., images relevant and irrelevant to w. If we know how relevant and irrelevant images are labeled with w and how they are distributed in the neighbor set, we can estimate the tag's distribution in the neighbors.

To formalize the above notions, we first define a few notations as listed in Table 2.1. We now study how images relevant and irrelevant to a tag are labeled with that tag. In a large user-tagged image database, it is plausible that for a specific tag w, the number of images irrelevant to the tag is significantly larger than the number of relevant images, i.e., $|R_w^c| \gg |R_w|$, where $|\cdot|$ is the cardinality operator on image sets. Moreover, one might expect that user tagging is better than tagging at random such that relevant images are more likely to be labeled, meaning $|L_w \cap R_w| > |L_w \cap R_w^c|$.

By approximating the probability of correct tagging $P(w|R_w)$ using $|L_w \cap R_w|/|R_w|$ and the the probability of incorrect tagging $P(w|R_w^c)$ using $|L_w \cap R_w^c|/|R_w^c|$, we have $P(w|R_w) > P(w|R_w^c)$. Hence, we make an assumption on user tagging behavior, that is,

> **Assumption 1: User tagging**. *In a large user-tagged image database, the probability of correct tagging is larger than the probability of incorrect tagging.*

Next, we analyze the distribution of images relevant and irrelevant with respect to tag w in the k nearest neighbor set of image I. Compared to random sampling,

a content-based visual search defined by a similarity function f can be viewed as a sampling process biased by the query image. We consider two situations with respect to the visual search accuracy, that is, equal to and better than random sampling. In the first situation where the visual search is equal to random sampling, the number of relevant images in the neighbor set is the same as the number of relevant images in a set of k images randomly selected from the collection. While in the second situation where the visual search is better than random sampling, given two images I_1 relevant to tag w and I_2 irrelevant to w, we expect to have

$$|N_f(I_1, k) \cap R_w| > |N_{rand}(k) \cap R_w| > |N_f(I_2, k) \cap R_w|.$$

For instance, consider w to be 'bridge', I_1 a bridge image and I_2 a non-bridge image. In this example, $N_f(I_1, k)$ should contain more bridge images than $N_{rand}(k)$, while $N_f(I_2, k)$ should contain less bridge images than $N_{rand}(k)$. Viewing random sampling as a baseline, we introduce an offset variable $\varepsilon_{I,w}$ to indicate the visual search accuracy. In particular, we use $(P(R_w) + \varepsilon_{I,w})$ to represent the probability that an image randomly selected from the neighbor set $N_f(I, k)$ is relevant with respect to w. Since an image is either relevant or irrelevant to w, we use $(1 - (P(R_w) + \varepsilon_{I,w}))$, namely $(P(R_w^c) - \varepsilon_{I,w})$, to represent the probability that an image randomly selected from $N_f(I, k)$ is irrelevant with respect to w. Then, the number of relevant images in the neighbor set is expressed as

$$|N_f(I, k) \cap R_w| = k \cdot (P(R_w) + \varepsilon_{I,w}), \tag{2.3}$$

and the number of irrelevant images in the neighbor set as

$$|N_f(I, k) \cap R_w^c| = k \cdot (P(R_w^c) - \varepsilon_{I,w}). \tag{2.4}$$

It is worth mentioning that the variable $\varepsilon_{I,w}$ is introduced to help us derive important properties of the proposed algorithm. We do not rely on $\varepsilon_{I,w}$ for implementing the algorithm.

Based on the above discussion, if the visual search is equal to random sampling, we have $\varepsilon_{I,w} = 0$. If the visual is better than random sampling, we have

$$\varepsilon_{I_1,w} > 0 > \varepsilon_{I_2,w}, \text{ for } I_1 \in R_w \text{ and } I_2 \in R_w^c. \tag{2.5}$$

We then make our second assumption as

> **Assumption 2: Visual search**. *A content-based visual search is better than random sampling.*

Bearing the analysis of user tagging and visual search in mind, we now consider the distribution of tag w within the neighbor set of image I. Since we can divide the

neighbor set into two distinct subsets $N_f(I,k) \cap R_w$ and $N_f(I,k) \cap R_w^c$, we count the number of w in the two subsets, separately. That is,

$$\begin{aligned} n_w[N_f(I,k)] &= n_w[N_f(I,k) \cap R_w] + n_w[N_f(I,k) \cap R_w^c] \\ &= k \cdot (P(R_w) + \varepsilon_{I,w}) P(w|R_w) + k \cdot (P(R_w^c) - \varepsilon_{I,w}) P(w|R_w^c). \end{aligned} \tag{2.6}$$

In a similar fashion we derive

$$n_w[N_{rand}(k)] = k \cdot (P(R_w)P(w|R_w) + P(R_w^c)P(w|R_w^c)) . \tag{2.7}$$

Since $n_w[N_{rand}(k)]$ reflects the occurrence frequency of w in the entire collection, we denote it as $Prior(w,k)$. By substituting Eq. 2.7 into Eq. 2.6, we obtain

$$n_w[N_f(I,k)] - Prior(w,k) = k \cdot (P(w|R_w) - P(w|R_w^c))\, \varepsilon_{I,w}. \tag{2.8}$$

Further, by defining

$$tagRelevance(w,I,k) := n_w[N_f(I,k)] - Prior(w,k), \tag{2.9}$$

we arrive at the following two theorems:

Theorem 1: Image ranking. *Given assumption 1 and assumption 2, tagRelevance yields an ideal image ranking for tag w, that is, for $I_1 \in R_w$ and $I_2 \in R_w^c$, we have $tagRelevance(w,I_1) > tagRelevance(w,I_2)$.*

Theorem 2: Tag ranking. *Given assumption 1 and assumption 2, tagRelevance yields an ideal tag ranking for image I, that is, for two tags w_1 and w_2, if $I \in R_{w_1}$ and $I \in \bar{R}_{w_2}$, we have $tagRelevance(w_1,I) > tagRelevance(w_2,I)$.*

We refer to the Appendix for detailed proofs of the two theorems. Note that in the proof of theorem 1, assumption 2 (Eq. 2.5) can be relaxed as ($\varepsilon_{I_1,w} > \varepsilon_{I_2,w}$) which we call relaxed assumption 2. Since the relaxed assumption is more likely to hold than its origin, this observation indicates that image ranking is relatively easier than tag ranking.

Our tag relevance function in Eq. 2.9 consists of two components which represents the distribution of the tag in the local neighborhood and in the entire collection, respectively. This observation confirms our conjecture made in the beginning of Section 2.3 that a good tag relevance measurement should take both distribution into account.

2.3.3 A Neighbor Voting Algorithm

We have argued in Section 2.3.2 that learning tag relevance boils down to computing $(n_w[N_f(I,k)] - Prior(w,k))$, i.e., the count of tag w in the k nearest neighbors of image I minus the prior frequency of w. Consider that each neighbor votes on w if it is labeled with w itself, $n_w[N_f(I,k)]$ is then the count of neighbor votes on w. Thereby, we introduce a neighbor voting algorithm: given a user-tagged image, we first perform content-based k-nn search to find its visual neighbors, and then for each neighbor image, we use its tags to vote on tags of the given image. We approximate the prior frequency of tag w as

$$Prior(w,k) \approx k\frac{|L_w|}{|\Phi|}, \tag{2.10}$$

where k is the number of visual neighbors, $|L_w|$ the number of images labeled with w, and $|\Phi|$ the size of the entire collection. Note that the function *tagRelevance* in Eq. 2.9 does not necessarily obtain positive results. We set the minimum value of *tagRelevance* to 1. In other words, if the learned tag relevance value of a user-contriubted tag is less than its original frequency in an image, we reject the tag relevance learning result for that image. In addition, we observe that the voting result might be biased by individual users who have a number of visually similar images, as shown in Figure 2.3(a). In order to make the voting decision more objective (which we target at), we introduce a unique-user constraint on the neighbor set. That is, each user has at most one image in the neighbor set per voting round. As shown in Figure 2.3(b), with the unique-user constraint we effectively reduce the voting bias. We finally summarize the procedure for learning tag relevance by neighbor voting in Algorithm 1.

Algorithm 1 Learning tag relevance by neighbor voting

Input: A user-tagged image I.
Output: $tagRelevance(w,I,k)$, i.e., the tag relevance value of each tag w in I.

Find k nearest visual neighbors of I from the collection with the unique-user constraint, i.e., a user has at most one image in the neighbor set.
for tag w in tags of I **do**
  $tagRelevance(w,I,k) = 0$
end for
for image J in the neighbor set of I **do**
  for tag w in (tags_of_J $\cap$ tags_of_I) **do**
    $tagRelevance(w,I,k) = tagRelevance(w,I,k) + 1$
  end for
end for
$tagRelevance(w,I,k) = tagRelevance(w,I,k) - Prior(w,k)$
$tagRelevance(w,I,k) = max(tagRelevance(w,I,k), 1)$

(a) (b)

Figure 2.3: Tag relevance learning with the unique-user constraint. The query example is the biggest image in the center of (a) and (b). The query is labeled with tag 'tiger' by a user. Figure (a) shows visual neighbors without the unique-user constraint, namely standard content-based search. Since the neighbor set is dominated by images from few users, the tag relevance value of 'tiger' voted by 1000 neighbors is 557. While in Figure (b), with the unique-user constraint, each user has at most one image in the neighbor set per voting round. The tag relevance value of 'tiger' voted by 1000 neighbors is thus reduced to 6. The unique-user constraint makes the voting result more objective.

2.4 Experimental Setup

2.4.1 Experiments

We evaluate our tag relevance learning algorithm in both an image ranking scenario and a tag ranking scenario. For image ranking, we compare three tag-based image retrieval methods with and without tag relevance learning. For tag ranking, we demonstrate the potential of our algorithm in helping user tagging in two settings, namely, tag suggestion for labeled images and tag suggestion for unlabeled images. Specifically, we design the following three experiments.

• **Experiment 1: Tag-based image retrieval**. We employ a general tag-based retrieval framework widely used in existing systems such as Flickr and YouTube. We adopt OKAPI-BM25, a well founded ranking function for text retrieval [53] as a baseline. Given a query q containing keywords $\{w_1, \ldots, w_n\}$, the relevance score of

an image I is computed as

$$score(q, I) = \sum_{w \in q} qtf(w) idf(w) \frac{tf(w) \cdot (k_1 + 1)}{tf(w) + k_1 \cdot (1 - b + b \frac{l_I}{l_{avg}})},$$

where $qtf(w)$ is the frequency of tag w in q, $tf(w)$ the frequency of w in the tags of I, l_I the total number of tags of I, and l_{avg} the average value of l_I over the entire collection. The function $idf(w)$ is calculated as $\log \frac{N - |L_w| + 0.5}{|L_w| + 0.5}$, where N is the number of images in the collection and $|L_w|$ is the number of images labeled with w. By using learned tag relevance value as updated tag frequency in the ranking function, we investigate how our algorithm improves upon the baseline. We study the performance of the baseline method and our method, given various combinations of parameters. In total, there are three parameters to optimize. One is k, the number of neighbors for learning tag relevance. We choose k from {100; 200; 500; 1000; 2000; 5000; 10,000; 15,000; 20,000}. The other two are b and k_1 in OKAPI-BM25. The parameter b ($0 \leq b \leq 1$) controls the normalization effect of document length. Here, document length is the number of tags in a labeled image. We let b range from 0 to 1 with interval 0.1. The variable k_1 is a positive parameter for regularizing the impact of tag frequency. Since k_1 does not affect ranking for single-word queries, we set k_1 to 2, a common choice in text retrieval [53].

Considering that the OKAPI-BM25 ranking function originally aims for text retrieval and hence might not be optimal for tag-based image retrieval, we further compare with a recent achievement in web image retrieval by Jing and Baluja [52] (see details in Section 2.2.2). As depicted in [52], there are two parameters to optimize: a dump factor d ($d > 0.8$) controlling the restart probability of random walk and m the number of top ranked results in an initial list to calculate the prior probability. We try various parameter combinations, i.e., $d \in \{0.85; 0.90; 0.95\}$ and $m \in \{5; 10; 20; 100; 1000\}$.

• **Experiment 2: Tag suggestion for labeled images**. Given an image labeled with some tags, we aim for automated methods that accurately suggest new tags relevant to the image. We investigate how our algorithm improves upon a recent method by Sigurbjörnsson and Van Zwol [12] by introducing visual content information into the tag suggestion process. Similar to [12], we first find x candidate tags having the highest co-occurrence with the initial tags. For each candidate tag, we then compute its relevance score with respect to the image as follows,

$$score(c, I) = score(c, \mathbf{w}_I) \cdot \frac{\lambda}{\lambda + (rank_c - 1)}, \tag{2.11}$$

where c is the candidate tag, I the image, and $\mathbf{w}_I$ the set of initial tags. The function $score(c, \mathbf{w}_I)$ computes a relevance score between the candidate tag and the initial tags. We adopt $Vote^+$, the best method in [12], as an implementation of the *score* function. The input $rank_c$ is the position of tag c in the candidate tag list ranked

by *tagRelevance* in descending order. The variable λ is a positive parameter for regularizing the effect of tag relevance learning. By optimizing the algorithm on the same training set as used in [12], we determine the optimized setting of the two parameters x and λ as 17 and 20, respectively.

• **Experiment 3: Tag suggestion for unlabeled images**. We compare with two model-free approaches: a tag frequency (tf) approach by Torralba *et al.* [119] and an approach by Wang *et al.* [125] which re-weights the frequency of a tag by its inverse document frequency (tf-idf). For our algorithm, since no user-defined tags are available, we consider all tags in the vocabulary as candidates. We estimate *tagRelevance* for each candidate tag with respect to the unlabeled image, and then rank the tags in descending order by *tagRelevance*. We take care to make the comparison fair. First, since the baselines do not consider user information, we remove the unique-user constraint from our algorithm. Second, for all methods we fix the number of the visual neighbors to 500, as suggested in [125]. Finally, for each method, we select the top 5 tags as a final suggestion for each test image.

In all the three experiments, we use *baseline* to represent the baseline methods, and *tagRelevance* for our method.

2.4.2 Data Collections

We choose Flickr as a test case of user tagging. We downloaded images from Flickr by randomly generating photo ids as query seeds. By removing images having no tags and those failed to extract visual features, we obtain 3.5 million labeled images in total. The images are of medium size with maximum width or height fixed to 500 pixels. After Porter stemming, the number of distinct tags per image varies from 1 to 1230, with an average value of 5.4. The collection has 573,115 unique tags and 272,368 user ids.

Note that the image retrieval experiment studies how well images are ranked, while the two tag suggestion experiments focus on how well tags are ranked. Different targets result in two different evaluation sets, one for image retrieval and the other for tag suggestion.

• **Evaluation set for image retrieval**. We create a ground truth set as follows. We select 20 diverse visual concepts as queries. The queries are listed in Table 2.2 with visual examples in Figure 2.4. As defined earlier, we consider a query concept and an image relevant if the concept is clearly visible in the image and we shall relate the concept to the visual content easily and consistently with common knowledge. Therefore, toys, cartoons, painting, and statues of the concept are treated as irrelevant. For each query, we randomly select 1000 examples from images labeled with the query in our 3.5 million Flickr collection, and relabel them according to our labeling criterion. We report user tagging accuracy of all 20 queries in Table 2.2. For each query, we score its 1000 test images with the two baseline methods and the proposed algorithm, respectively. The images are then ranked in light of their relevance

Table 2.2: Ground truth statistics for our image retrieval experiment. Each query has 1000 manually labeled examples. User tagging accuracy is the number of relevant images divided by 1000.

	3.5 million user-tagged images	
Query	*Tag frequency*	*User tagging accuracy*
airplane	15,231	0.447
beach	64,348	0.331
boat	25,385	0.424
bridge	25,197	0.762
bus	14,296	0.641
butterfly	8,476	0.701
car	37,614	0.548
cityscape	11,063	0.657
classroom	7,763	0.388
dog	52,981	0.764
flower	71,699	0.829
harbor	8,420	0.503
horse	27,008	0.736
kitchen	11,464	0.389
lion	8,509	0.326
mountain	36,844	0.502
rhino	4,929	0.346
sheep	3,603	0.525
street	40,772	0.426
tiger	8,214	0.224

scores. If two images have the same score, they are ranked according to photo ids in descending order so that latest uploaded images are ranked ahead.

• **Evaluation set for tag suggestion**. To evaluate the performance of tag suggestion for labeled and unlabeled images, we adopt a ground truth set from [12], which is created by manually assessing the relevance of tags with respect to images. The set consists of 331 Flickr images, having no overlap with the 3.5 million collection. Since the relevance of tags '2005', '2006', and '2007' with respect to an image is quite subjective, we remove the three tags from the ground truth beforehand. Note that these tags might be predicted by tag suggestion methods. In that case, we consider the tags irrelevant. The number of tags per image in the evaluation set varies from 1 to 14, with an average value of 5.5. Examples of the ground truth are shown in Figure 2.5. For experiment 2, we follow the same data partition as [12], that is, 131 images for training and the remaining 200 for testing. Since no training is required for all the three methods in experiment 3, we take the entire ground truth set (331 images in total) for testing.

Figure 2.4: Visual examples of 20 queries in our image retrieval experiment.

Figure 2.5: Multimedia examples of the ground truth for our tag suggestion experiments.

2.4.3 Evaluation Criteria

For image retrieval, images relevant with respect to user queries should be ranked as high as possible. Meanwhile, ranking quality of the whole list is important not only for user browsing, but also for applications using search results as a starting point. For tag suggestion, tags relevant with respect to user images should be ranked as high as possible. Moreover, the candidate tag list should be short such that users pick out relevant tags easily and efficiently. Therefore, we adopt the following two standard criteria to measure the different aspects of the performance. Given a ranked list of l instances where an instance is an image for image retrieval and a tag for tag suggestion, we measure

• **precision at n (P@n)**: The proportion of relevant instances in the top n retrieved results, where $n \leq l$. For image retrieval, we report P@10, P@20, and P@100 for each query. For tag suggestion, we report P@1 and P@5, averaged over all

test images, as used in [12]. We consider a predicted tag relevant with respect to a test image if the tag is from the ground truth tags of the image. The Porter stemming is done before tag matching. Since we always predict 5 tags for each image, for those images having less than 5 ground truth tags, their P@5 will be smaller than 1.

• **average precision (AP)**: AP measures ranking quality of the whole list. Since it is an approximation of the area under the precision-recall curve [150], AP is commonly considered as a good combination of precision and recall, e.g., [40, 46, 86]. The AP value is calculated as $\frac{1}{R}\sum_{i=1}^{l}\frac{R_i}{i}\delta_i$, where R is the number of relevant instances in the list, R_i the number of relevant instances in the top i ranked instances, δ_i=1 if the i-th instance is relevant and 0 otherwise. To evaluate the overall performance, we use mean average precision (MAP), a common measurement in information retrieval. MAP is the mean value of the AP over all queries in the image retrieval experiment and all test images in the tag suggestion experiments.

2.4.4 Large-scale Content-based Visual Search

To implement the neighbor voting algorithm, we need to define visual similarity between images and then search visual neighbors in our 3.5 million Flickr photo database. Visual similarity between two images is measured using corresponding visual features. Since we need features relatively stable for search and efficient to compute to cope with millions of images, we adopt a combined 64-dimensional global feature as a tradeoff between effectiveness and efficiency. The feature is calculated as follows. For each image, we extract 44-d color correlgoram [47], 14-d color texture moment [145], and 6-d RGB color moment. We separately normalize the three features into unit length and concatenate them into a single vector. We use the Euclidean distance as a dissimilarity measurement. The feature is used throughout all the three experiments.

To search millions of images by content, efficient indexing methods are imperative for speed up. We adopt a K-means clustering based method for its empirical success in large-scale content-based image retrieval [59]. First for indexing, we divide the whole dataset into smaller subsets by the K-means clustering. Each subset is indexed by a cluster center. Then for a query image, we find neighbors within fewer subsets whose centers are the closest to the query. The search space is thus reduced. Since the search operation in individual subsets can be executed in parallel, we execute neighbor search in a distributed super computer.

2.5 Results

2.5.1 Experiment 1: Tag-based Image Retrieval

As shown in Figure 2.6, our *tagRelevance* substantially outperforms the *tag baseline* for all parameter settings. Recall that the OKAPI-BM25 parameter b controls the

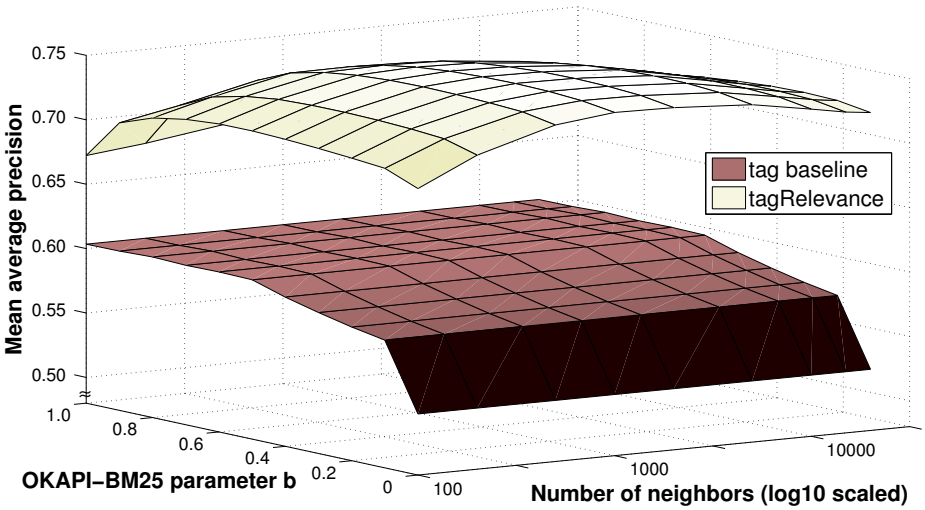


Figure 2.6: Experiment 1: An overall comparison between image retrieval methods with and without tag relevance learning. The *tag baseline* method uses original tags, while our *tagRelevance* method uses learned tag relevance as updated tag frequency. We study the retrieval performance given various combinations of the OKAPI-BM25 parameter b and the number of neighbors for tag relevance learning. We measure the overall performance using mean average precision of the 20 queries from Figure 2.4. The *tagRelevance* consistently outperforms the *tag baseline* for all parameter settings.

impact of normalizing scores by the total number of tags within an image. Hence, we observe different behavior of b in the two methods: the *tag baseline* tends to perform well when b approaches 1; in contrast, the *tagRelevance* improves as b approaches 0. Since tag frequency is not discriminative in original tagging, the baseline method heavily relies on the normalization factor. While in the new method, tag frequency becomes more discriminative after tag relevance learning.

The proposed algorithm is also robust to the number of neighbors used for voting. To show this property, we first run leave-one-out cross validation on the 20 queries to determine the optimized OKAPI-BM25 parameter b for *tag baseline* and our method, which is 0.8 and 0.3, respectively. As shown in Figure 2.7, *tagRelevance* consistently outperforms *tag baseline*. More precisely, we reach at least 20% relative improvement in terms of MAP when the number of neighbors is between 200 and 20,000.

We conclude experiment 1 with a per-query comparison between three methods, namely *tag baseline*, *baseline* [52], and our *tagRelevance*. We again use the optimized parameters for *tag baseline* and *tagRelevance*. The number of neighbors is 1000. For *baseline* [52], we take the best output of *tag baseline* as initial search results and run leave-one-out cross validation to obtain an optimized parameter setting, i.e., d=0.85 and m=100. As shown in Table 2.3, for some queries *baseline* [52] is on a

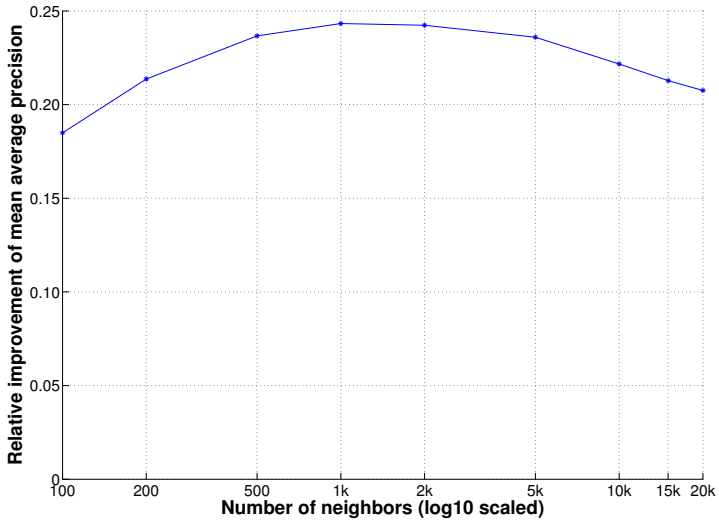


Figure 2.7: Experiment 1: Relative improvement in terms of mean average precision (MAP) over the best tag baseline with respect to the number of neighbors for learning tag relevance. The best baseline is reached at b=0.8 with MAP 0.605. By using learned tag relevance value as updated tag frequency for retrieval, we obtain at least 20% relative improvement in terms of MAP when the number of neighbors is between 200 and 20,000.

par with our *tagRelevance*, especially for the top ranked results. Nevertheless, for the majority of the queries and the evaluation metrics, the proposed algorithm compares favorably with the two baselines. On average, compared with the tag baseline, we obtain a relative improvement in terms of P@20 by 28.8% and 24.3% in terms of MAP. Compared with the baseline [52], we obtain a relative improvement in terms of P@20 by 15.3% and 19.9% in terms of MAP.

2.5.2 Experiment 2: Tag Suggestion for Labeled Images

We report the performance of the two tag suggestion methods in Table 2.4. For all evaluation metrics, the *tagRelevance* improves upon the *baseline*. More precisely, we obtain an improvement of 6.3% in terms of P@1 and 6.6% in terms of MAP. While the improvement in terms of P@5 is 4.5%, which is relatively small. The reasons are two-fold. First, by measuring the relevance of a candidate tag with respect to an image at both textual and visual aspects, the *tagRelevance* is more likely to rank relevant tags ahead of irrelevant ones. Second, since we use the *baseline* as a starting point, if the method fails to retrieve relevant tags, it is unlikely to create a better ranked list. As shown in Table 2.5, compared to the *baseline*, our method finds more relevant tags which describe visual aspects of an image.

Table 2.3: Experiment 1: Per-query comparison between image retrieval methods with and without tag relevance learning. **Bold numbers** indicate the top performers. For most of the 20 queries, we improve upon the baseline methods by using learned tag frequency as updated tag frequency. On average, compared with the tag baseline, we obtain a relative improvement in terms of P@20 by 28.8% and 24.3% in terms of MAP. Compared with the baseline [52], we obtain a relative improvement in terms of P@20 by 15.3% and 19.9% in terms of MAP..

	Precision at 5			**Precision at 20**			**Precision at 100**			**Average precision**		
Query	*tag baseline*	*baseline* [52]	*tagRelevance*	*tag baseline*	*baseline* [52]	*tagRelevance*	*tag baseline*	*baseline* [52]	*tagRelevance*	*tag baseline*	*baseline* [52]	*tagRelevance*
airplane	0.400	**0.800**	0.600	0.500	**0.750**	0.400	**0.520**	**0.520**	0.510	0.446	0.513	**0.531**
beach	0.400	0.400	**1.000**	0.500	0.350	**0.900**	0.370	0.370	**0.710**	0.383	0.356	**0.666**
boat	0.400	0.200	**1.000**	0.600	0.550	**0.950**	0.520	0.520	**0.720**	0.477	0.487	**0.619**
bridge	**1.000**	0.800	0.800	**0.950**	0.900	0.900	0.880	0.880	**0.900**	0.802	0.806	**0.830**
bus	**1.000**	**1.000**	0.600	0.700	**0.850**	**0.850**	0.740	0.740	**0.870**	0.684	0.792	**0.836**
butterfly	0.800	0.800	**1.000**	0.800	0.900	**0.950**	0.940	0.940	**0.990**	0.816	0.838	**0.932**
car	**1.000**	**1.000**	0.800	0.650	0.800	**0.900**	0.660	0.660	**0.800**	0.610	0.674	**0.730**
cityscape	0.000	**1.000**	**1.000**	0.500	**0.950**	**0.950**	0.690	0.690	**0.980**	0.698	0.683	**0.907**
classroom	0.800	**1.000**	0.600	0.500	**0.900**	0.750	0.500	0.500	**0.600**	0.482	**0.551**	0.532
dog	**1.000**	0.800	**1.000**	0.950	0.950	0.950	0.830	0.830	**0.930**	0.806	0.820	**0.869**
flower	1.000	1.000	1.000	0.900	0.950	**1.000**	0.910	0.910	**0.980**	0.889	0.891	**0.963**
harbor	**0.800**	0.600	**0.800**	0.700	0.650	**0.950**	0.600	0.600	**0.900**	0.582	0.614	**0.768**
horse	0.800	**1.000**	**1.000**	0.550	0.950	**1.000**	0.700	0.700	**0.890**	0.718	0.774	**0.829**
kitchen	0.800	**1.000**	**1.000**	0.800	**0.900**	**0.900**	0.600	0.600	**0.900**	0.518	0.642	**0.742**
lion	0.800	0.800	**1.000**	0.950	0.450	**1.000**	0.420	0.420	**0.930**	0.476	0.393	**0.774**
mountain	0.600	0.400	**1.000**	0.500	0.650	**0.900**	0.500	0.500	**0.840**	0.517	0.550	**0.769**
rhino	1.000	1.000	1.000	0.950	**1.000**	0.950	0.820	0.820	**0.860**	0.697	0.659	**0.746**
sheep	**1.000**	**1.000**	0.800	0.850	0.900	**0.950**	0.790	0.790	**0.890**	0.638	0.677	**0.748**
street	0.400	0.400	**0.600**	0.300	0.500	**0.600**	0.390	0.390	**0.680**	0.412	0.477	**0.578**
tiger	0.400	0.800	**1.000**	0.550	0.450	**0.900**	0.610	0.610	**0.780**	0.442	0.338	**0.673**
average	0.720	0.790	**0.880**	0.685	0.765	**0.882**	0.649	0.649	**0.833**	0.605	0.627	**0.752**

Table 2.4: Experiment 2: Tag suggestion for labeled images. For each image, we choose the top 5 ranked tags. **Bold numbers** indicate the top performers.

	Tag suggestion methods	
Evaluation criteria	*baseline* [12]	*tagRelevance*
Precision at 1	0.522	**0.555**
Precision at 5	0.359	**0.375**
Mean average precision	0.622	**0.663**

2.5.3 Experiment 3: Tag Suggestion for Unlabeled Images

As shown in Table 2.6, our *tagRelevance* method outperforms the two baseline methods for all evaluation criteria. Since the *tf* baseline [119] ranks tags in terms of tag frequency, it tends to suggest tags occurring frequently in the entire collection such as '2006'. By re-weighting tag frequency using the idf value, the *tf-idf* baseline [125] may restrain such effects to some extent. However, it risks over-weighting rare tags like 'campcourtney'. By contrast, our *tagRelevance* uses the frequency of a tag minus its prior frequency to restrain high frequent tags. Meanwhile, since the prior frequency of the rare tags are small, these tags are not over-weighted. Hence, our method is more effective and robust.

Since all the three methods rely on the effectiveness of the visual search, we further study how the methods behave when the accuracy of the visual search is low (P@n$<$ 0.05), medium (0.05$\leq$P@n$\leq$0.20), and high (P@n$>$0.20). As illustrated in Table 2.7, we select three test images, where the manually assessed accuracy of the 30 nearest neighbors is 0.77, 0.00, and 0.10, respectively. We observe that all methods succeed when the visual search is good. Obviously, all methods fail when no relevant images exist in the neighbor set. Interestingly, in an intermediate situation when the visual search is unsatisfactory with only a few relevant examples in the neighbor set, our method predicts more relevant tags than the two baseline methods. We make a further investigation on the entire test set. Since manually assessing the visual search accuracy for the test set is laborious, we estimate the accuracy as follows. For each test image with a number of ground truth tags, we consider a neighbor image relevant if the tags of the neighbor image and the tags of the test image have at least one tag in common. It is in this way that we count relevant neighbors and subsequently compute the visual search accuracy. As shown in Figure 2.8, our algorithm outperforms the baselines, given different visual search accuracy. In particular, our algorithm performs especially better when the visual search accuracy is medium or low. The evidence from both Table 2.7 and Figure 2.8 demonstrates the potential of our tag relevance learning algorithm. In addition, note that the majority of the test images have unsatisfactory visual search results (61.9% low and 35.5% medium), resulting in

Table 2.5: Experiment 2: Examples of tag suggestion for labeled images by different methods. The *italic* font indicates relevant tags and the **bold** font indicates unique relevant tags produced by our method. We improve upon the *baseline* by addressing tag relevance with respect to the visual content. Compared to the *baseline*, our method finds more relevant tags which describe visual aspects of the images.

User-labeled images		**New suggested tags**	
Image	*Tags*	*baseline* [12]	*tagRelevance*
	lighthouse	beach sea ocean harbor 2005	sea beach ocean harbor ***sunset***
	loch scotland lake waves	*water* castle *beach* katrine edinburgh	*water* ***mountain*** *beach* castle sea
	d40 london stonehenge uk bath	england sister nikon nikond40 stone	england sister water ***street*** stone
	mexico	2006 vacation new oaxaca honeymoon	2006 vacation ***beach*** new honeymoon

a relatively low performance for automatic image tagging. This observation implies that tag suggestion for unlabeled images can be improved further by including more advanced visual features.

2.5.4 Discussion

So far, we have verified the effectiveness of the proposed algorithm for tag-based image retrieval and automatic tag suggestion for labeled and unlabeled images. As discussed in Section 2.3.2, since image ranking imposes a relatively looser requirement

Table 2.6: Experiment 3: Tag suggestion for unlabeled images. For each image, we choose the top 5 ranked tags. **Bold numbers** indicate the top performers.

	Tag suggestion methods		
Evaluation criteria	*baseline* [119]	*baseline* [125]	*tagRelevance*
Precision at 1	0.061	0.068	**0.097**
Precision at 5	0.068	0.059	**0.074**
Mean average precision	0.126	0.120	**0.153**

Table 2.7: Experiment 3: Examples of tag suggestion for unlabeled images by different methods. The *italic* font indicates relevant tags and the **bold** font indicates unique relevant tags produced by our method. We illustrate how the three methods perform when the accuracy of the visual search is high (for the image at the top), low (for the image in the middle), or medium (for the image at the bottom). Compared to the two baseline methods, our method predicts more relevant tags even when the visual search is unsatisfactory.

Visual Search		Suggested tags by different methods		
Image	*Accuracy*	*baseline* [119]	*baseline* [125]	*tagRelevance*
	0.77	*flower* *red* macro nature garden	*flower* *red* macro *rose* garden	*flower* *red* macro *rose* garden
	0.00	2006 family japan beach vacation	2006 cat family campcourtney august12006	icehockety hockey family hurricane cat
	0.10	2006 wedding japan park vacation	2006 pepperell wedding japan park	japan ***bike*** hiking park texas

on content-based visual search than tag ranking, the former is easier than the latter. The empirical evidences from the three experiments confirm this conclusion. To better understand how our assumptions on visual search hold in practice, we introduce a validation experiment as follows. For each of the 20 queries used in the image retrieval experiment, we count the proportion of $<$ *relevant image*, *irrelevant image* $>$ pairs that satisfy assumption 2 and relaxed assumption 2, respectively. Note that for other visual similarity functions in the literature, we can use this method to estimate how

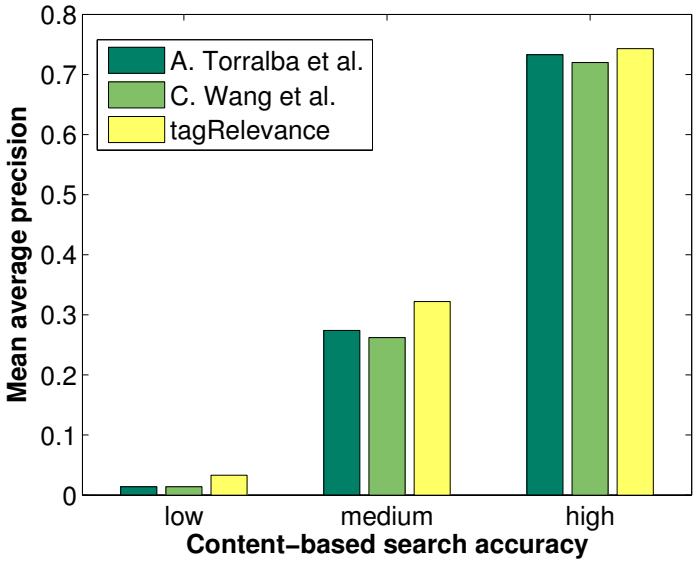


Figure 2.8: Experiment 3: The effect of content-based visual search on tag suggestion for unlabeled images. We categorize the accuracy of a content-based visual search into three levels, that is, low (precision$<$0.05), medium (0.05$\leq$precision$\leq$0.20), and high (precision$\geq$0.20). The proposed algorithm outperforms the baselines, given different levels of visual search accuracy. In particular, our algorithm performs especially better when the visual search accuracy is medium or low.

a particular visual similarity measurement meets the assumptions and consequently select proper features based on the estimation. As shown in the boxplot in Figure 2.9, on average, 37.8% pairs satisfy assumption 2 and 73.4% pairs satisfy relaxed assumption 2. The results again verify our conclusions that learned tag relevance is a good criterion for image ranking and it can be improved further for tag ranking by leveraging more advanced visual features.

Up to now, we have successfully managed 3.5 million user-tagged images by executing our algorithm in parallel. Considering the heavy computation effort, however, it would be interesting to investigate in the future how to regularize the learning process, say from a Hill-climbing set, to ease the computation for new user-submitted images. Though our evaluations are conducted on Flickr, the proposed algorithm is general. Hence, it is also applicable to other social photo sharing websites. Finally, we present in Figure 2.10 some of the tag relevance learning results with updated tag frequency.

2.6 Conclusions

Since user tagging is known to be subjective and overly personalized, a fundamental problem in social image analysis and retrieval is how to accurately interpret the

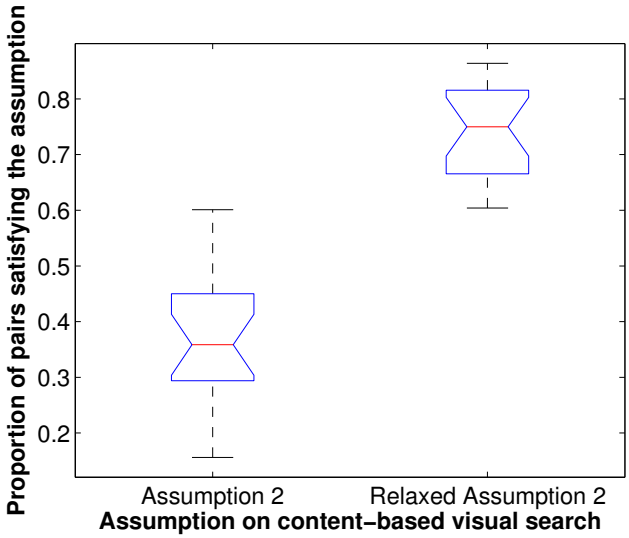


Figure 2.9: Validation of the two assumptions on content-based visual search. We refer to Section 2.3.2 for the definitions of assumption 2 and relaxed assumption 2. For each of the 20 queries used in the image retrieval experiment, we count the proportion of $< relevant\ image,\ irrelevant\ image >$ pairs that satisfy assumption 2 and relaxed assumption 2, respectively. We use boxplot to visualize the results. On average, 37.8% pairs satisfy assumption 2 and 73.4% pairs satisfy relaxed assumption 2.

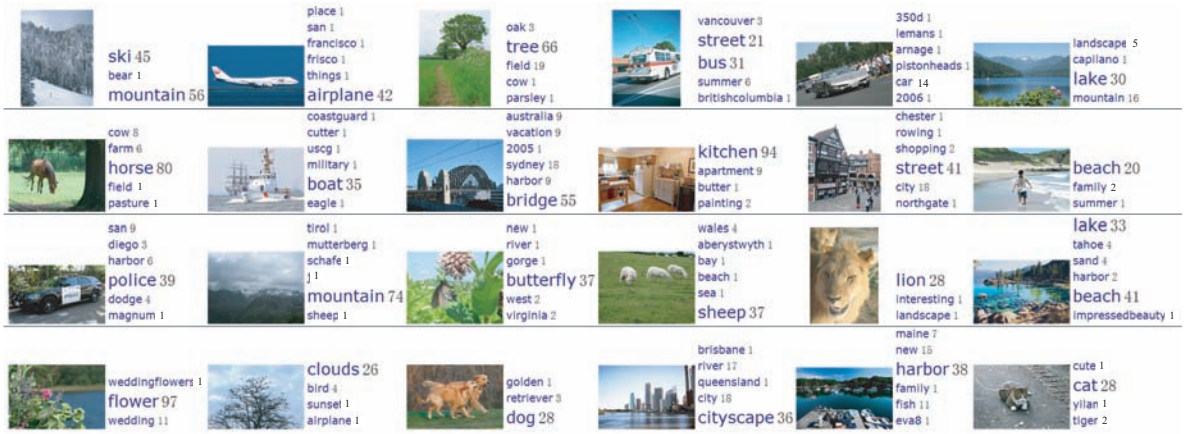


Figure 2.10: Results of learning tag relevance by neighbor voting. The images are user-tagged photos from our 3.5 million Flickr collection. The texts on the right side of each image are user-contributed tags followed by estimated tag relevance value. The number of neighbors for tag relevance learning is 1000.

relevance of a tag with respect to the visual content the tag is describing. In this chapter, we propose a neighbor voting algorithm as an initial step towards conquering

the problem. Our key idea is to learn the relevance of a tag with respect to an image from tagging behaviors of visual neighbors of that image. In particular, our algorithm estimates tag relevance by counting neighbor votes on tags. We show that when 1) the probability of correct user tagging is larger than the probability of incorrect user tagging and 2) content-based visual search is better than random sampling, our algorithm produces a good tag relevance measurement for both image ranking and tag ranking. Moreover, since the proposed algorithm does not require any model training for any visual concept, it is efficient in handling large-scale image data sets.

To verify our algorithm, we conduct three experiments on 3.5 million Flickr photos: one image ranking experiment and two tag ranking experiments. For the image ranking experiment, we improve social image retrieval by using learned tag relevance as updated tag frequency in a general tag-based retrieval framework. Retrieval with tag relevance learning obtains a 24.3% relative improvement in terms of mean average precision, when compared to a tag-based retrieval baseline. For the tag ranking experiments, we consider two settings, i.e., tag suggestion for labeled images and tag suggestion for unlabeled images. In the tag suggestion experiment for labeled images, our algorithm finds more tags which describe visual aspects of an image, leading to a relative improvement of 6.3% in terms of mean average precision when compared to a text baseline. In the tag suggestion experiment for unlabeled images, our algorithm compares favorably against two baselines. Specifically, we effectively restrain high frequency tags without over-weighting rare tags. Our study demonstrates that the proposed algorithm predicts more relevant tags even when the visual search is unsatisfactory. In summary, all the three experiments show the general applicability of tag relevance learning for both image ranking and tag ranking. The results suggest a large potential of our algorithm for real-world applications.

Chapter 3

Tag Relevance Fusion for Social Image Search

How to fuse tag relevance estimates? In this chapter we develop early and late fusion schemes from generic multimedia analysis in the new context of tag relevance estimation. We systematically study the characteristics and performance of early and late tag relevance fusion. Experiments on a large present-day benchmark show that tag relevance fusion leads to better image search. Moreover, we find that the unsupervised fusion methods are practically as effective as the supervised alternatives, but without the need of *any* training efforts*.

*A preliminary version of this work received the Best Paper Award from *the ACM International Conference on Image and Video Retrieval* 2010 [65]. Submitted [67].

3.1 Introduction

Searching for the increasing amounts of varied and dynamically changing images on the social web is important. A number of applications are grounded on social image search, such as landmark visualization [55], visual query suggestion [147], training data acquisition [115], photo-based question answering [144], and photo-based advertisements [70], to name a few. Given that social images are often described by user-contributed tags, one might expect tag-based retrieval to be a natural and good starting point for image search. Compared to content-based search [27], tag-based search potentially bypasses the semantic gap problem, and its scalability has been verified by decades of text retrieval research [7]. However, due to varied reasons, including batch tagging and the diversity of user knowledge, social tagging is known to be ambiguous, subjective, and inaccurate [81]. Moreover, since individual tags are used only once per image in the social tagging paradigm, relevant and irrelevant tags for a specific image are not separable by tag statistics alone. Estimating the relevance of social tags with respect to the images they are describing is essential for generic image search.

For tag relevance estimation, quite a few methods have been proposed. For example, Liu et al. [75] utilize a random walk model to rank tags in terms of their relevance to the visual content. Chen et al. [20] train a Support Vector Machine classifier per tag. Given an image and its social tags, Zhu et al. [151] measure the relevance of a specific tag in terms of its semantic similarity to the other tags. We proposed a neighbor voting algorithm which exploits tagging redundancies among multiple users [64]. By using learned tag relevance value as a new ranking criterion, better image search results are obtained, when compared to image search using original tags.

Positioned in a deluge of social data, however, tag relevance estimation is challenging. Visual concepts for example "boat" or "garden" vary significantly in terms of their visual appearance and visual context. A single measurement of tag relevance as proposed in previous work is limited to tackle such large variations, resulting in suboptimal image search. For large-scale data, we consider fusing multiple tag relevance estimates as an important extension to methods for tag relevance estimation.

In a broad sense of tag relevance fusion, we shall consider multiple sources of evidence such as visual content [20, 64, 75], tag co-occurrence [12, 151], social notes [89, 98], and personal tagging history [60]. Among them, visual content is the only objective piece of evidence. Hence, in this work we focus on fusing tag relevance derived from visual content analysis.

It is becoming increasingly clear that no single feature can represent the visual content completely [38, 78, 126]. Global features are suited for capturing the gist of scenes [88], while local features better depict properties of objects [122]. As shown previously in content-based image search [117, 127], image annotation [41, 80], and video concept detection [128, 137], fusing multiple visual features is beneficial. The question arises what fusion methods are suited in the new context of tag relevance estimation.

Before answering the question, we think over the main characteristics of social data: large-scale, miscellaneous, and dynamic. These characteristics imply that a fused tag relevance measurement should favorably exploit and aggregate the large amount of miscellaneous information scattered on the social web. Moreover, lightweight fusion methods are preferred to keep pace with the rapid development of social data. To establish our study, we develop the notion of early and late fusion from Snoek et al. [110] in the new context, and define

Definition 1 (Early Tag Relevance Fusion). *Fusion schemes that integrate individual features before estimating social tag relevance scores.*

Definition 2 (Late Tag Relevance Fusion). *Fusion schemes that first use individual features to estimate social tag relevance scores separately, and then integrate the scores.*

The main contributions of this chapter are as follows.

- We propose tag relevance fusion as an extension of tag relevance estimation.
- Using the neighbor voting algorithm as a base tag relevance estimator [64], we present a systematic study on early and late tag relevance fusion. We extend the base estimator for both early and late fusion. Our previous work [65], which discusses late tag relevance fusion only, is a special case of this work.
- Experiments on a large benchmark show that tag relevance fusion leads to better image search.
- This study provides a practical mechanism to exploit diverse visual features in interpreting tag relevance for social image search.

The problem we study lies at the crossroads of social tag relevance estimation and visual fusion. So next we present a literature review of both areas.

3.2 Related Work

3.2.1 Social Tag Relevance Estimation

A number of methods have been proposed to attack the tag relevance estimation problem [20, 56, 64, 74, 75, 112, 138, 149, 151]. We structure these methods in terms of the main rationale they use to estimate tag relevance. We summarize the rationale into the following three aspects: visual consistency [56, 64, 75, 112], semantic consistency [20, 151], and visual-semantic consistency [74, 149]. Given two images labeled with the same tag, the visual consistency based methods conjecture that if one image is visually closer to images labeled with the tag than the other image, then the former image is more relevant to the tag. Liu et al. [75] use a random walk model to find such visually close images. Sun and Bhowmick [112] exploit visual consistency to quantify

a tag's relevance to the visual content. We proposed a neighbor voting algorithm which infers the relevance of a tag with respect to an image by counting its visual neighbors labeled with that tag [64]. Lee et al. [56] first identify tags which are suited for describing the visual content by a dictionary lookup. Later, they apply the neighbor voting algorithm to the identified tags. Zhu et al. [151] investigate semantic consistency, measuring the relevance of a tag to an image in terms of its semantic similarity to the other tags assigned to the image, ignoring the visual content of the image itself. Chen et al. [20] assume that photos uploaded by the same user within a short time span form a semantically related group. They train SVM models for individual tags, and use the models to estimate image tag relevance within the photo group. To jointly exploit visual and semantic consistency, Liu et al. [74] perceive tag relevance estimation as a semi-supervised multi-label learning problem, while Zhu et al. [149] formulate the problem as decomposing an image tag co-occurrence matrix. In all the above methods, only a single feature is considered. We hypothesize that fusing multiple tag relevance estimates driven by diverse features could improve such methods.

3.2.2 Visual Fusion

Snoek et al. [110] classify fusion methods into two groups: early fusion and late fusion. We follow their taxonomy to organize our literature review on visual fusion. In early fusion, a straightforward method is to concatenate individual features to form a new single feature [3, 110]. As feature dimensionality increases, the method suffers from the curse of dimensionality [92]. Another disadvantage of the method is the difficulty to combine features into a common representation [110]. Instead of feature concatenation, another method is to combine visual similarities of the individual features [41, 80, 128]. In the context of image annotation, Makadia et al. [80] and Guillaumin et al. [41] linearly combine multiple visual similarities. In a video concept detection context, Wang et al. [128] also choose linear fusion to combine similarity graphs defined by different features. In late fusion, models are obtained separately on the individual features and their output is later combined [126, 137]. Wu et al. [137] first train base classifiers using distinct features. Then, they view the output of the base classifiers as a new feature to obtain a final classifier. Wang et al. [126] adaptively combine the base classifiers in a boosting framework. To the best of our knowledge, visual fusion in the tag relevance estimation context has not been well explored.

3.3 Base Tag Relevance Estimators

For a valid comparison between early and late fusion, we should choose the same base tag relevance estimators for both fusion schemes. Thus, before delving into the discussion about tag relevance fusion and its solutions, we first make our choice of base estimators. For the ease of consistent description, we use x to denote an image,

and w for a social tag. Let $g(x, w)$ be a base tag relevance function whose output is a confidence score of a tag being relevant to an image. Further, let S be a large set of social-tagged images, and S_w the set of images labeled with w, $S_w \subset S$.

A base estimator should be data-driven and favorably exploit the large amount of social data. Moreover, it should be generic enough to adapt to both early and late fusion. In that regard, we choose the neighbor voting algorithm proposed in our previous work [64]. A recent study by Sun et al. [113] indicates that this algorithm is the most effective for tag relevance estimation. In order to find visual neighbors from S for a given image x, we use $f(x)$ to represent a specific visual feature vector, and $S_{x,f,k}$ as the k nearest visual neighbors of x, measured in terms of f. We also have to specify a distance function for each given feature. The optimal distance varies in terms of tasks [41, 94]. Here for all features, we choose the Euclidean distance for its widespread use in the literature. In the neighbor voting algorithm, $g(x, w)$ is computed as

$$g(x, w) = \frac{|S_{x,f,k} \cap S_w|}{k} - \frac{|S_w|}{|S|}, \tag{3.1}$$

where $|\cdot|$ is the cardinality of a set. We see from (3.1) that more neighbor images labeled with the tag induces larger tag relevance scores, while common tags which have high frequency and thus low descriptive power are suppressed by the second term. We conceptualize early and late tag relevance fusion with diagrams in Fig. 6.1, and elucidate them next.

3.4 Tag Relevance Fusion

3.4.1 Problem Formalization

From an information fusion perspective [14], diversity in base tag relevance estimators is important for effective fusion. We can generate multiple tag relevance estimators by varying the visual feature f, the number of neighbors k, or both. For a given feature, as the larger set of visual neighbors always includes the smaller set of visual neighbors, the parameter k has a limited impact on the diversity. Hence, we fix k and diversify the base estimators by using diverse visual features. Let $F = \{f_1, \ldots, f_m\}$ be a set of such features. We use $g_i(x, w)$ to denote a base estimator specified by feature f_i, $i = 1, \ldots, m$. For the two fusion schemes defined in Section 5.1, we use $G^e(x, w)$ to denote a fused tag relevance estimator obtained by early fusion, and $G^l(x, w)$ to denote a late fused estimator.

Since linear fusion is a well accepted choice for visual fusion as described in Section 3.2.2, we follow this convention for tag relevance fusion. For early fusion, we aim for a better neighbor set by combining visual similarities defined by the m features. Concretely, given two images x and x', let $sim_i(x, x')$ be a visual similarity defined

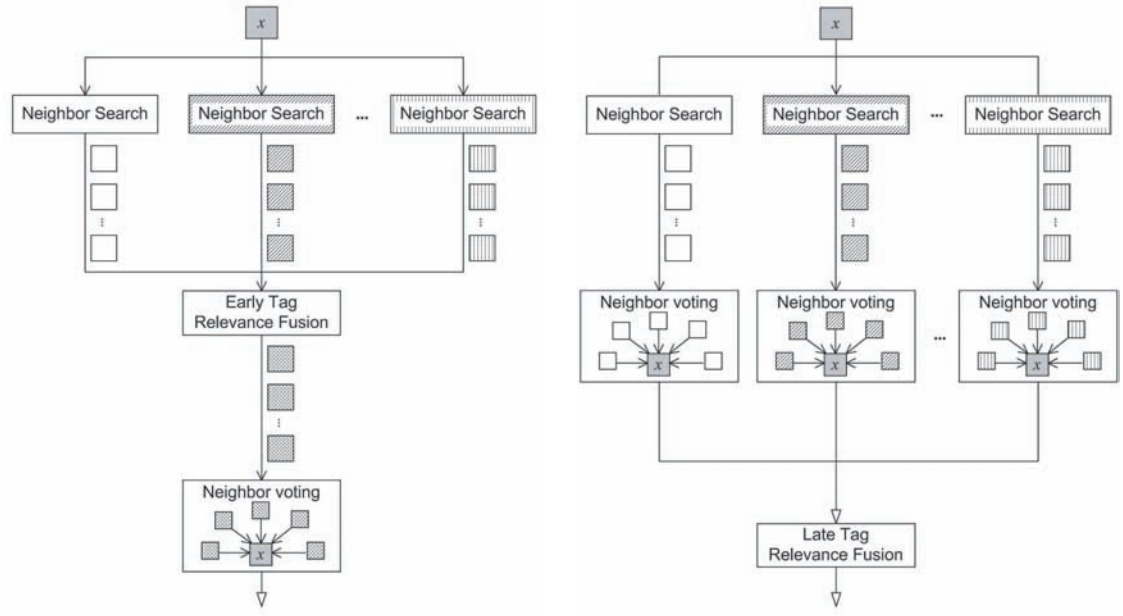


Figure 3.1: The two proposed tag relevance fusion schemes. We use the neighbor voting algorithm [64] to realize base tag relevance estimators. Given an image x, different textured backgrounds indicate its visual neighbors obtained by diverse features. In Early Tag Relevance Fusion (a), we fuse multiple visual neighbor sets (denoted by $\downarrow$) to obtain a better neighbor set for tag relevance estimation. In Late Tag Relevance Fusion (b), we fuse multiple tag relevance estimates (denoted by $\triangledown$).

by feature f_i. We define the fused visual similarity as

$$sim_\Lambda(x, x') = \sum_{i=1}^{m} \lambda_i \cdot sim_i(x, x'), \tag{3.2}$$

where λ_i is a weight indicating the importance of feature f_i in the fusion process. The subscript Λ makes the dependence of the fused similarity on $\{\lambda_i\}$ explicit. We choose features which are intellectually devised, so we assume that they are better than random guessing, meaning adding them is helpful for measuring the visual similarity. Hence, we constrain our solution with $\lambda_i \geq 0$. Since normalizing weights by dividing by their sum does not affect image ranking, any linear fusion with nonnegative weights can be transformed to a convex combination. So we enforce $\sum_{i=1}^{m} \lambda_i = 1$. Replacing the similarity used in (3.1) by the fused similarity (3.2) leads to the early fused tag relevance function:

$$G^e_\Lambda(x, w) = \frac{|S_{x,\Lambda,k} \cap S_w|}{k} - \frac{|S_w|}{|S|}, \tag{3.3}$$

where $S_{x,\Lambda,k}$ denotes the k nearest neighbors obtained by $sim_\Lambda(x, x')$.

In a similar fashion, we define our linear late fused tag relevance function:

$$G^l_\Lambda(x,w) = \sum_{i=1}^{m} \lambda_i \cdot g_i(x,w). \tag{3.4}$$

In order to study the statistical properties of $G^e_\Lambda(x,w)$ and $G^l_\Lambda(x,w)$, we extend the derivation of a single estimator in our earlier study [64] to the new context. The effectiveness of an estimator $g(x,w)$ depends on the accuracy of social tagging and visual neighbor search [64]. To describe the social tagging accuracy, we divide the social image set S into two disjoint subsets, S_{w+} and S_{w-}, where images in S_{w+} are relevant to w and images in S_{w-} are irrelevant to w. Let Q_{w+} be the probability of correct tagging, i.e., an image randomly sampled from S_{w+} is labeled with w, and Q_{w-} be the probability of incorrect tagging, i.e., an image randomly sampled from S_{w-} is labeled with w. We have established in our previous work [64] that in the social web the probability of correct tagging is larger than the probability of incorrect tagging, thus

$$Q_{w+} - Q_{w-} > 0. \tag{3.5}$$

Given an image relevant to w, the accuracy of neighbor search by a specific feature f is the percentage of neighbors which are also relevant to w. To describe the neighbor search accuracy, we consider random sampling as a baseline. Let P_{w+} be the probability that an image randomly sampled from S is relevant to w. We introduce an offset variable $\varepsilon_{x,w,f}$ to reflect the relative accuracy compared to random sampling. In particular, we use $(P_{w+} + \varepsilon_{x,w,f})$ to represent the probability that an image randomly sampled from the neighbor set $S_{x,f,k}$ is relevant to w. We can re-express $g(x,w)$ as

$$g(x,w) = \varepsilon_{x,w,f} \cdot (Q_{w+} - Q_{w-}). \tag{3.6}$$

For the derivation of (3.6), we refer to the paper [64]. Consequently, for any image x_{w+} relevant to w and any image x_{w-} irrelevant to w, if

$$\varepsilon_{x_{w+},w,f} - \varepsilon_{x_{w-},w,f} > 0, \tag{3.7}$$

we will have $g(x_{w+},w) - g(x_{w-},w) > 0$, meaning an ideal tag relevance estimator which ranks all relevant images in front of all irrelevant images.

To describe the neighbor search accuracy of the fused similarity (3.2), we use $\varepsilon_{x,w,\Lambda}$ to denote the corresponding offset variable. Substituting $\varepsilon_{x_{w+},w,\Lambda}$ for $\varepsilon_{x_{w+},w,f}$ in (3.6), we re-write $G^e_\Lambda(x,w)$ as

$$G^e_\Lambda(x,w) = \varepsilon_{x,w,\Lambda} \cdot (Q_{w+} - Q_{w-}). \tag{3.8}$$

For images x_{w+} and x_{w-}, the difference between their tag relevance values is

$$G^e_\Lambda(x_{w+},w) - G^e_\Lambda(x_{w-},w) = (\varepsilon_{x_{w+},w,\Lambda} - \varepsilon_{x_{w-},w,\Lambda}) \cdot (Q_{w+} - Q_{w-}). \tag{3.9}$$

In the context of image annotation, Makadia et al. [80] and Guillaumin et al. [41] report that combining diverse features improves the neighbor search accuracy. According to their studies, we make our assumption about early fusion:

Assumption 1 (Early fusion). *A fused visual similarity is better than visual similarities using individual features, i.e.,* $\varepsilon_{x_{w+},w,\Lambda} - \varepsilon_{x_{w-},w,\Lambda} > 0$ *is more likely to be valid than* $\varepsilon_{x_{w+},w,f} - \varepsilon_{x_{w-},w,f} > 0$.

Due to the limitation of individual features for representing the visual content and finding the correct visual neighbors, inequality (3.7) might be violated. Given the diverse feature set, the inequality of the fused similarity is more likely to hold. Having inequality (3.5) and assumption 1, we can conclude that $G^e_\Lambda(x_{w+}, w) - G^e_\Lambda(x_{w-}, w) > 0$ is more likely to be valid than $g(x_{w+}, w) - g(x_{w-}, w) > 0$. This means the early fused estimator is better than the base estimators for ranking relevant images ahead of irrelevant images.

For late tag relevance fusion, by putting (3.6) into (3.4), we re-write $G^l_\Lambda(x, w)$ as

$$G^l_\Lambda(x, w) = \left(\sum_{i=1}^{m} \lambda_i \cdot \varepsilon_{x,w,f_i} \right) \cdot (Q_{w+} - Q_{w-}). \tag{3.10}$$

For images x_{w+} and x_{w-}, the difference between their tag relevance values is

$$G^l_\Lambda(x_{w+}, w) - G^l_\Lambda(x_{w-}, w) = (Q_{w+} - Q_{w-}) \sum_{i=1}^{m} \lambda_i \cdot (\varepsilon_{x_{w+},w,f_i} - \varepsilon_{x_{w-},w,f_i}). \tag{3.11}$$

According to (3.11), as long as inequality (3.7) is valid for the majority of the features, we will have $G^l_\Lambda(x_{w+}, w) - G^l_\Lambda(x_{w-}, w) > 0$. We describe this by our assumption on late fusion:

Assumption 2 (Late fusion). *For a diverse range of concepts, on average,* $\sum_{i=1}^{m} \lambda_i \cdot (\varepsilon_{x_{w+},w,f_i} - \varepsilon_{x_{w-},w,f_i}) > 0$ *is more likely to be valid than* $\varepsilon_{x_{w+},w,f_i} - \varepsilon_{x_{w-},w,f_i} > 0$ *for any individual feature* f_i.

Using diverse features results in base tag relevance estimators complementary to each other. Combining (3.5) and assumption 2, we can conclude that $G^l_\Lambda(x_{w+}, w) - G^l_\Lambda(x_{w-}, w) > 0$ is more likely to be valid than $g(x_{w+}, w) - g(x_{w-}, w) > 0$. This means the late fused estimator is better than the base estimators for ranking relevant images in front of irrelevant images.

From assumption 1 and assumption 2 we see that the effectiveness of tag relevance estimation no longer counts on a specific feature, but on the majority of the features used. This provides a theoretical motivation for fusing multiple tag relevance estimates driven by diverse features.

As shown in (3.3) and (3.4), for both early and late fusion, we have expressed their corresponding tag relevance functions in a convex combination. It is thus possible to leverage the same optimization algorithms for determining the fusion parameters. In order to evaluate a specific Λ, we need a performance measure such as average

precision to assess image rankings obtained by $G_\Lambda(x, w)$, where $G_\Lambda(x, w)$ is $G^e_\Lambda(x, w)$ in early fusion and $G^l_\Lambda(x, w)$ in late fusion. To formalize the optimization process, let D_w be a set of training images for tag w, where each image is labeled with the tag by social tagging, with ground truth created by manual verification. Let $rank(D_w; G_\Lambda)$ be a ranking of D_w, obtained by sorting D_w in descending order by $G_\Lambda(x, w)$. We use $\pi(rank(D_w; G_\Lambda))$ to represent a performance measure function which evaluates the ranking accuracy in terms of the ground truth. The goal of supervised fusion is to find a Λ such that π is maximized on D_w, formally

$$\Lambda^* = \underset{\Lambda}{\operatorname{argmax}}\, \pi(rank(D_w; G_\Lambda)), \tag{3.12}$$

with the convexity constraint,

$$\Lambda = \{\lambda_1, \ldots, \lambda_m\}, \lambda_i \geq 0, \sum_{i=1}^{m} \lambda_i = 1.$$

3.4.2 Fusion Methods

It is apparent that for early (late) fusion, better features (estimators) should have larger weights. The intuition helps us devise learning algorithms which exploit the performance measure as feedback to exclude many suboptimal weights, without the need to evaluate them. Moreover, performance measures widely used in the literature such as average precision are not differentiable. So we have to choose an algorithm which does not require gradient computation. Bearing these in mind, we choose the coordinate ascent based optimization technique, exploited by Metzler and Croft [82] in the domain of document retrieval. Here, we apply the technique to supervised early and late tag relevance fusion.

Supervised Fusion: Coord-Ascent. The Coordinate Ascent method iteratively solves (3.12) by optimizing merely one parameter in a learning round, with the remaining parameters fixed. Concretely, suppose λ_i is the parameter being optimized. Since π is not differentiable, we obtain the optimal value of λ_i by grid search: we evaluate λ_i with its values ranging over $\{0, 1/L, 2/L, \ldots, 1\}$, where L is an integer parameter controlling the quantization granularity. The value which maximizes π is selected as the new value of λ_i. Then, λ_{i+1} is activated and the same procedure is repeated. The loop continues until π no longer increases. To ensure the convexity constraint, we re-normalize the parameters after each iteration.

Given the large array of tags in the social web, where well-labeled training data are often unavailable, it is worthwhile to consider unsupervised fusion algorithms. Given no prior information concerning $\{\lambda_i\}$, we should choose uniform weights, meaning we make the least assumptions about things we do not know [50]. Following this thought, we consider the following two simple fusion algorithms: Average and Borda Count.

Unsupervised Fusion I: Average. For early fusion, the visual similarity is the average of $\{sim_i(x, x')\}$. To cope with the potential scale issue, we re-scale the

similarity scores of individual features by min-max normalization as used by Makadia et al. [80]. For late fusion, the corresponding $G^l_\Lambda(x, w)$ is simply the average of $\{g_i(x, w)\}$.

Unsupervised Fusion II: Borda. The Borda Count algorithm is well recognized as a solid choice for combining rankings generated by multiple sources of evidence [**?**, 89]. The only difference between Borda and Average is that Borda quantizes (continuous) scores (which are visual similarities in early fusion or tag relevance values in late fusion) into discrete ranks. It is thus more robust to outliers when compared to Average. To apply the Borda algorithm, for early fusion we replace $sim_i(x, x')$ in (3.2) by the corresponding rank-based scores. For late fusion, $g_i(x, w)$ in (3.4) is replaced.

The unsupervised fusion methods do not need a training step. To train a supervised fusion model, a learning step is required. For late fusion, when $\{g_i(x, w)\}$ have been pre-computed, evaluating one of the m parameters in Λ in an iteration involves computing $G^l_\Lambda(x, w)$ for each image in D_w, with a time complexity of $O(|D_w| \cdot m)$. Subsequently, sorting D_w requires $O(|D_w| \cdot \log |D_w|)$. As there are L values to evaluate in an iteration, the complexity per iteration is $O(L \cdot |D_w| \cdot (m + \log |D_w|))$. The overall complexity for optimizing the late fusion model is $O(T \cdot L \cdot |D_w| \cdot (m + \log |D_w|))$, where T is the number of iterations. For early fusion, suppose that the neighbor sets $\{S_{x,f_i,k}\}$ have been pre-computed. Evaluating a specific parameter in an iteration involves computing (3.3) for each image in D_w. The most computationally intensive part of (3.3) is obtaining $S_{x,\Lambda,k}$. Computing the fused similarity (3.2) has an order of $O(|S| \cdot m)$, and selecting the k neighbors from S by partial sort needs $O(|S| \cdot \log k)$. So getting $S_{x,\Lambda,k}$ for each image in D_w has a time complexity of $O(|S| \cdot (m + \log k))$. Evaluating a given parameter has an order of $O(|D_w| \cdot |S| \cdot (m + \log k))$ plus $O(|D_w| \cdot \log |D_w|)$ for sorting D_w. Note that $\log |D_w| \ll |S|$ in general, meaning we can ignore the sorting cost. So the complexity per iteration is $O(L \cdot |D_w| \cdot |S| \cdot (m + \log k))$, and the overall complexity for optimizing the early fusion model is $O(T \cdot L \cdot |D_w| \cdot |S| \cdot (m + \log k))$. Hence, training a supervised late fusion model is far more efficient than its early fusion counterpart.

3.5 Experimental Setup

Given the large-scale nature of our problem, it is impractical to explicitly verify the assumptions on early and late fusion, as this would require ground truth on every image from the neighbor sets of all visual features. More importantly, our goal is social image search. Therefore, we evaluate the performance of the entire search systems.

3.5.1 Data sets

Social Images for Tag Relevance Estimation S. We use a publicly available set of 3.5M social-tagged images[†], collected from Flickr in a random manner in our previous work [64]. Since batch-tagged images tend to be visually redundant, we remove such images. Also, we remove images having no tags corresponding to WordNet. After this preprocessing step, we obtain a compact set of 80K images as an instantiation of S.

Ground-truth data. We choose NUS-WIDE contributed by Chua et al. [22]. This widely used ground-truth data consists of Flickr images with manually verified annotations for 81 diverse visual concepts. We again remove batch-tagged images, and preserve images whose social tags contain at least one of the 81 concepts. For social image search, supervised fusion models shall be trained using existing data, and applied to data which will be uploaded to the social web in the future. To simulate such a scenario, we divide images in NUS-WIDE into two subsets in terms of their DateUploaded property: NUS-past and NUS-future. We use NUS-past for training (64,048 images), and NUS-future for testing (64,049 images). The statistics of the data are given in Table 3.1.

3.5.2 Social Image Search Experiments

For each of the 81 concepts, we conduct tag-based image search. We sort images labeled with the concept in descending order by (fused) tag relevance values. The two fusion schemes, *early* and *late*, and the three fusion algorithms, *Coord-Ascent*, *Average*, and *Borda*, result in six fusion methods. We use *EarlyFusion-Average* to represent the combination of early fusion and the Average algorithm. In a similar manner, we name the other five methods *EarlyFusion-Borda*, *EarlyFusion-Coord-Ascent*, *LateFusion-Average*, *LateFusion-Borda*, and *LateFusion-Coord-Ascent*. For a more comprehensive comparison, we also report image search performance using three simple metadata features: DateUploaded, Views, and TagNum. Given an image, Views indicates how many times the image has been viewed. TagNum is the number of social tags assigned to the image. For DateUploaded and Views, we rank images in descending order to favor freshness and popularity. For TagNum, we sort images in ascending order to penalize over-tagged images.

Evaluation Criteria. We adopt average precision (AP), a common choice for evaluating visual search engines [105]. For an overall assessment, we report mean average precision (MAP), the average of AP scores of all concepts. Because we aim to answer whether tag relevance estimation can benefit from fusion, we consider the performance difference between single measurements of tag relevance and different fusion methods more informative to draw conclusions.

[†]Data available at `http://staff.science.uva.nl/~xirong/tagrel/`

Table 3.1: Statistics of the ground-truth data [22] used in our experiments. The large variance in the tagging accuracy of diverse concepts implies the importance of tag relevance estimation for social image search.

	Training data				Test data			
Per concept	min	max	mean	stdev	min	max	mean	stdev
No. images	125	7,749	1,480	1,584	117	9,198	1,595	1,857
No. relevant images	7	7,212	879	1,278	5	8,540	939	1,473
Tagging accuracy	0.06	0.93	0.52	0.21	0.02	0.93	0.51	0.21

The number of images for a given concept is the number of images labeled with the concept by social tagging. The number of relevant images is the number of images labeled with and relevant to the concept. (Social) Tagging accuracy is the number of relevant images divided by the number of images. .

3.5.3 Implementation

Base tag relevance estimators $\{g_i(x, w)\}$. The base estimators are specified by the visual features f and the number of neighbors k. As we have discussed in Section 3.4, k does not contribute significantly for diversifying the base estimators. So we empirically fix k to be 500. Concerning f, we choose the following four visual features which describe image content from different perspectives: COLOR, CSLBP, GIST, and DSIFT. COLOR is a 64-d global feature, combining the 44-d color correlogram [47], the 14-d texture moments [145], and the 6-d RGB color moments. CSLBP is a 80-d center-symmetric local binary pattern histogram [44], capturing local texture distributions. GIST is a 960-d feature describing dominant spatial structures of a scene by a set of perceptual measures such as naturalness, openness, and roughness [88]. DSIFT is a 1024-d bag-of-keypoints histogram depicting local information of the visual content. We adopt dense sampling for keypoint localization and the SIFT descriptor for keypoint description [122]. As aforementioned, for all features we use the Euclidian distance to measure visual similarity. We denote the four tag relevance estimators by the corresponding feature names.

Parameters for Supervised Fusion. As we discussed in Section 3.4.2, optimizing the early fusion model is computationally much more expensive than optimizing the late fusion model. To keep them comparable, for each concept we randomly sample at maximum 500 training examples to form the training data D_w. We empirically set $L = 100$. On a dual-quad-core compute node with 2.4 GHz CPUs and 24 GB memory, training $G^e_\Lambda(x, w)$ takes about 49 minutes per iteration. Training $G^l_\Lambda(x, w)$ is far more efficient: approximately 0.3 seconds per iteration. Concerning factors affecting the trained models, we observe that the choice of L and the amount of training data are less dominant compared to the divergence between training and test data.

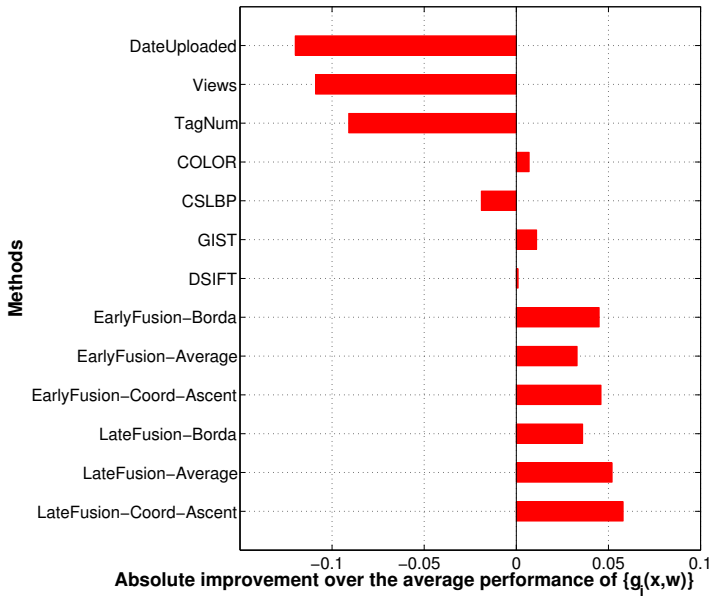


Figure 3.2: Comparing six variants of tag relevance fusion. The average performance of the four base estimators $\{g_i(x, w)\}$ is 0.637. Compared to this reference point, tag relevance fusion with various configurations obtains absolute improvements ranging from 0.033 to 0.058.

3.6 Results

Single Tag Relevance versus Metadata Features. To better show the performance difference between different methods, we use the average performance of the four base tag relevance estimators as a reference point, which has an MAP of 0.637. As shown in Fig. 3.2, we find among the three metadata features that TagNum with an MAP of 0.546 is the best, followed by Views and DateUploaded. They are all inferior to the base estimators. This result confirms that image search using learned tag relevance is superior to image search using original tags [64]. Concerning the base estimators, as they use four distinct features, their performance varies with a standard deviation of 0.013. Notice that the deviation is approximately 2.1% of the average performance. Thus, we conclude that the neighbor voting algorithm is robust with respect to the visual features used.

Tag Relevance Fusion versus Single Tag Relevance. As shown in Fig. 3.2, tag relevance fusion leads to better image search performance. For instance, EarlyFusion-Borda and LateFusion-Average obtain an absolute improvement of 0.045 and 0.052, respectively, which is 7.1% and 8.2% better than the average performance of the single tag relevance methods. For a comprehensive understanding, we make a per-concept comparison, as illustrated in Fig. 3.3. For 78 out of the 81 concepts, the search performance is improved by both EarlyFusion-Borda and LateFusion-Average.

Further, for each concept we check the best performer among the four base esti-

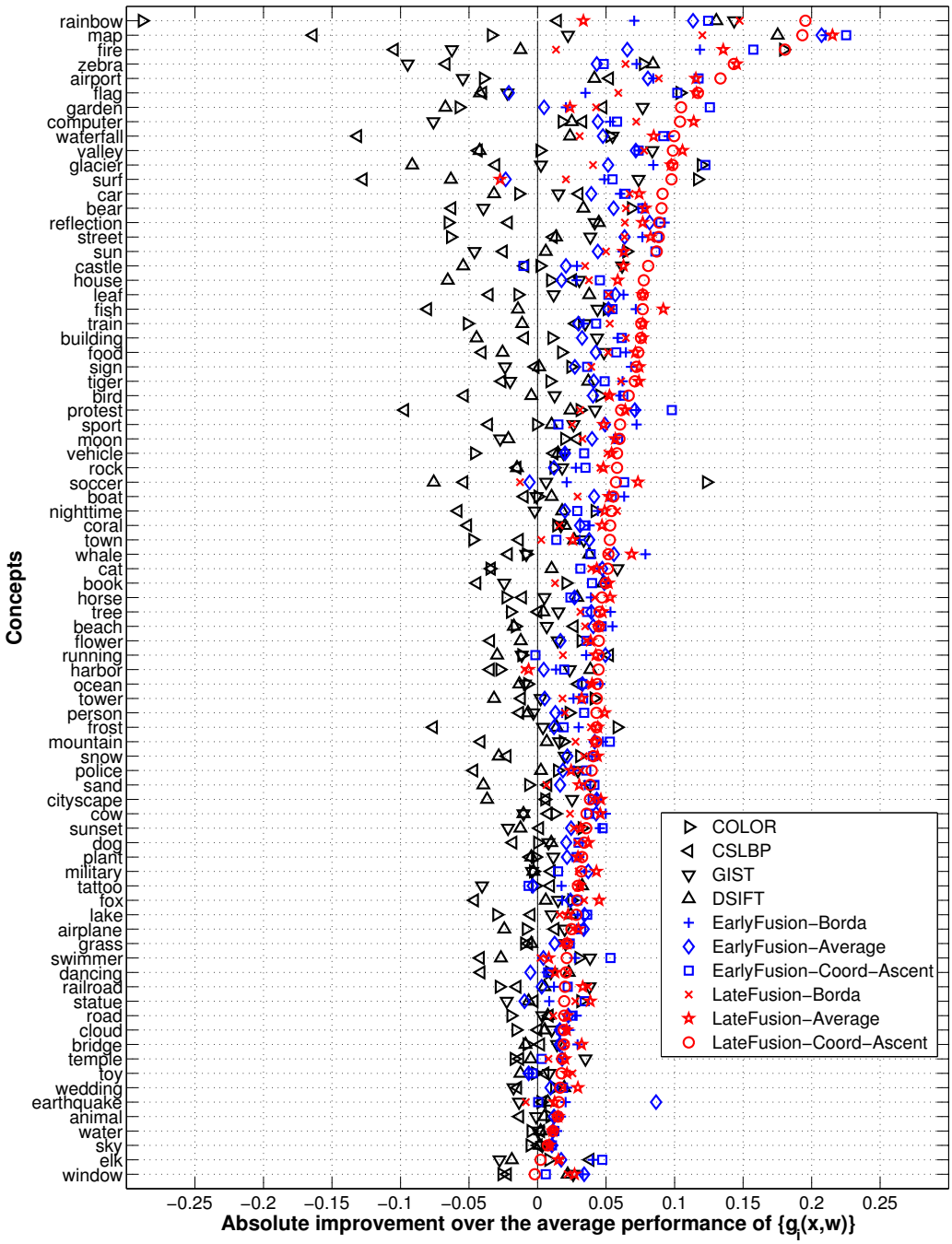


Figure 3.3: Tag Relevance Fusion versus Single Tag Relevance: A per-concept comparison. The concepts are sorted in descending order by the absolute improvement of the LateFusion-Coord-Ascent method. Best viewed in color.

mators. We find that for 22 concepts COLOR is the best, for 10 concepts CSLBP, for 31 concepts GIST, and for 18 concepts DSIFT. For every concept, we compare

EarlyFusion-Coord-Ascent and LateFusion-Coord-Ascent with the concept's best performer. For 55 out of the 81 concepts, EarlyFusion-Coord-Ascent outperforms the best performers, which vary per concept. For 65 concepts, LateFusion-Coord-Ascent is better than the best performers. These results show the effectiveness of tag relevance fusion for social image search.

Comparing Unsupervised Fusion Methods. The overall results of the six fusion methods are given in Table 3.2. Concerning the two unsupervised fusion algorithms, we observe that Borda is better than Average for early fusion, while Average is more suited for late fusion. As shown in Fig. 3.4, for 70 concepts EarlyFusion-Borda is better than EarlyFusion-Average. The result shows that the rank-based normalization is more effective than the min-max normalization for unsupervised fusion of visual similarities. In contrast, LateFusion-Average outperforms LateFusion-Borda for 68 concepts. This result is mainly due to the fact that the base estimators already include an effect of smoothing by quantizing the visual neighborhood via neighbor voting. Further quantization by the Borda algorithm makes tag relevance estimates less discriminative. Only when some base estimators yield large yet inaccurate values such as COLOR for "rainbow" as illustrated in Fig. 3.6(b), Borda is preferred. We also observe the limitations of late fusion for addressing concepts which are rarely tagged. Consider "earthquake" for instance. There are only 113 images labeled with the concept in our social collection S. The base estimators mostly yield zero scores for the concept. Late Fusion does not add much in this case. In contrast, by directly manipulating the neighbor sets, EarlyFusion-average yields the best result for "earthquake". In general, we consider LateFusion-Average the best choice for unsupervised fusion, for its competitive performance and its flexibility in adding new base estimators.

Comparing Supervised Fusion Methods. As shown in Table 3.2, the supervised methods achieve the best performance for both early and late fusion. In theory, given enough training data which well represents (unseen) test data, EarlyFusion-Coord-Ascent should be better than LateFusion-Coord-Ascent, as the latter implies quantization at every decision point and thus a possible loss of information. However, given the dynamic nature of social data, training data from the past is sometimes inadequate to represent future data. In such a cross-data scenario, late fusion with

Table 3.2: An overall comparison between the six tag relevance fusion methods.

	Fusion Schemes	
Fusion Algorithms	Early	Late
Coord-Ascent	0.683	0.695
Average	0.670	0.689
Borda	0.682	0.673

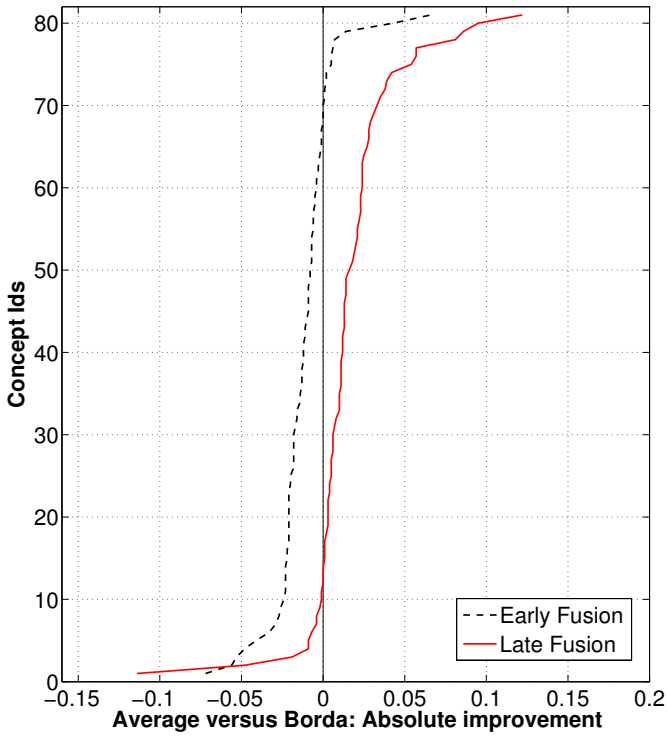


Figure 3.4: Comparing the two unsupervised fusion algorithms: Average versus Borda. Borda corresponds to the reference line x=0. Falling at the left side of the line means Borda is better, while falling at the right side means Average is better. For a better view of the data, we sort concepts for early and late fusion separately. While Borda is better than Average for early fusion, Average is more suited for late fusion.

quantization tends to be more robust than early fusion. Indeed, our experiment indicates that early and late fusion perform similarly, with even a small increase for the latter.

Supervised versus Unsupervised Fusion Methods. Comparing the supervised methods to their unsupervised alternatives, for some concepts such as "surf" and "rainbow" where there is large variance in the performance of the base estimators, the relative improvement is up to 37.3% and 27.4%. However, when averaging over all concepts, the improvement is marginal. The EarlyFusion-Coord-Ascent method beats EarlyFusion-Average with a relative gain of 1.9%, and LateFusion-Coord-Ascent improves LateFusion-Average with a relative gain of 0.9% only. This result seems counter-intuitive, as one would expect a larger improvement from supervised learning. To reveal whether the result is caused by a few outlier concepts, we also look into individual concepts. As shown in Fig. 3.5, although for 41 out of the 81 concepts LateFusion-Coord-Ascent improves over LateFusion-Average, there are only 6 concepts which have a relative improvement of more than 5%.

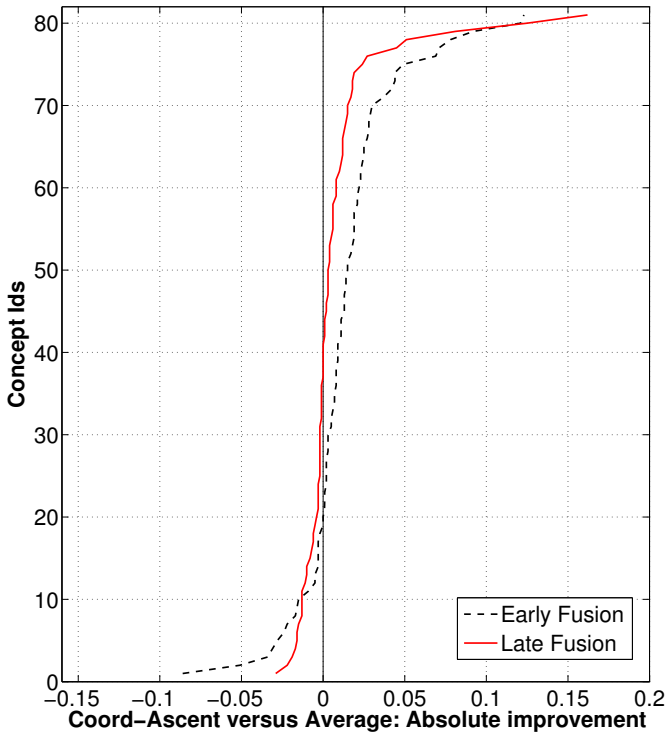
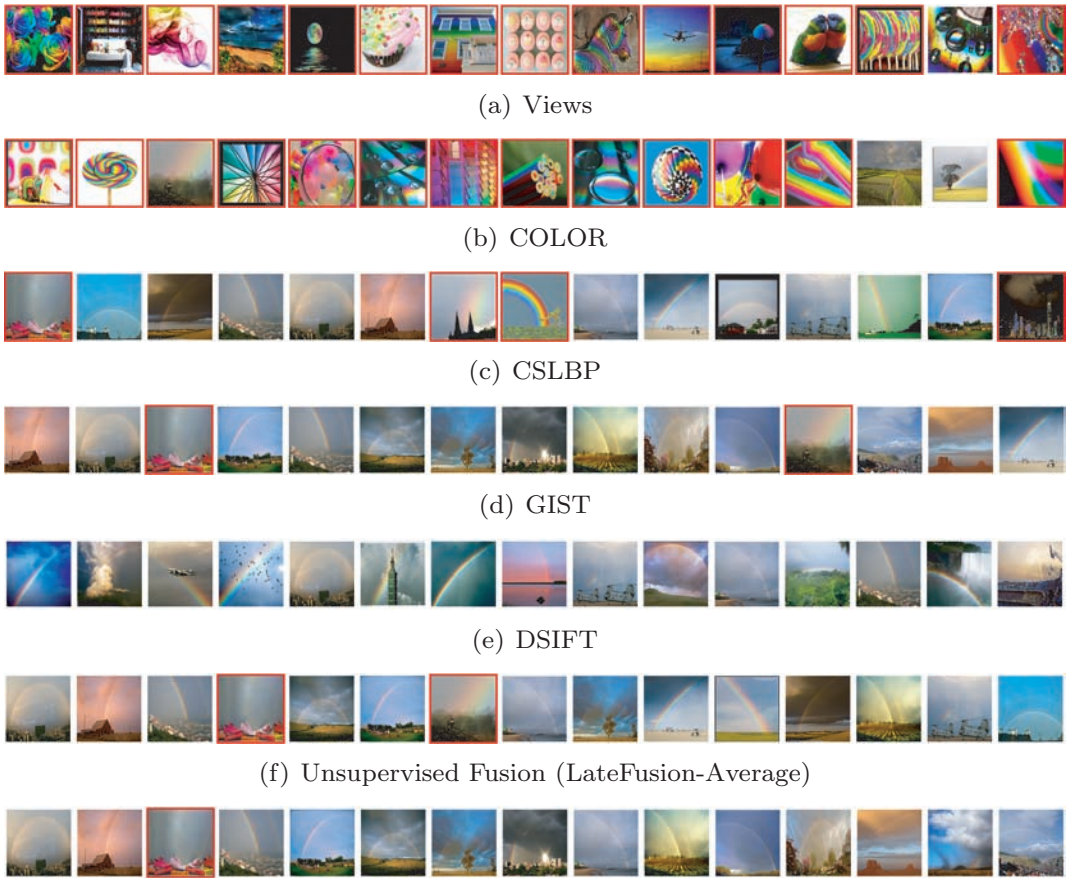


Figure 3.5: Comparing supervised and unsupervised fusion algorithms: Coord-Ascent versus Average. Average corresponds to the reference line x=0. Falling at the left side of the line means Average is better, while falling at the right side means Coord-Ascent is better. For a better view of the data, we sort concepts for early and late fusion separately. For both early and late fusion, the coordinate ascent learning algorithm obtains parameters better than the uniform weights. Because late fusion is more robust than early fusion, the improvement in late fusion is smaller than the improvement in early fusion.

We attribute such counter-intuitive results to the following two reasons. First, due to vagaries of social data, the models optimized on the past data tend to be suboptimal for the future data. Consider the concept "soccer" for instance. The CSLBP and DSIFT estimators have AP scores of 0.263 and 0.389 on the training data respectively, while their performance on the test data is much lower with AP scores of 0.152 and 0.130. In contrast, the performance of the other two base estimators is relatively stable when crossing datasets. As a consequence, the optimized model indeed makes the performance degenerate when compared to LateFusion-Average. Second, different from traditional learning-to-fuse scenarios where features or rankers might be just better than random guess [35, 117], the features employed in this study were intellectually designed and shown to be effective. As shown in Fig. 3.2, the base estimators already provide a strong starting point. Moreover, distinct features result in complementary neighbor sets for early fusion and complementary tag relevance



(a) Views

(b) COLOR

(c) CSLBP

(d) GIST

(e) DSIFT

(f) Unsupervised Fusion (LateFusion-Average)

(g) Supervised Fusion (LateFusion-Coord-Ascent)

Figure 3.6: Image search results of the concept "rainbow". The top 15 results of each method are shown. A red border indicates a false positive result. Best viewed in color.

estimates for late fusion. Therefore, though unsupervised fusion is suboptimal for some concepts, it is practically as effective as supervised fusion in general.

3.7 Discussions and Conclusions

Estimating the relevance of social tags with respect to the visual content is essential for social image search but challenging. In this chapter we introduce tag relevance fusion as an extension to methods for tag relevance estimation. Using the neighbor voting algorithm to instantiate base tag relevance estimators [64], we have conducted a systematic study on early and late tag relevance fusion schemes. Image search experiments on a large benchmark show that compared to a single measurement of tag relevance, fusing multiple tag relevance driven by diverse features improves the image search accuracy. By a coordinate ascent based supervised fusion, we obtain

a relative improvement of 8.2% in terms of mean average precision. As revealed by overall and concept-by-concept analysis, tag relevance fusion robustifies tag relevance estimation for social image search.

The two fusion schemes each have their merit. Early fusion which directly manipulates the neighbor sets is more effective for addressing concepts rarely tagged. Late fusion is more robust to differences between the data on which the method is trained and the data to which it is applied. As the latter is what is mostly encountered when exploiting social data, it is the method of choice.

An exciting observation is that the simple unsupervised fusion algorithms perform comparably to their supervised counterparts. LateFusion-Average, which simply averages multiple tag relevance estimates, is effective, with a loss of 1.9% only, when compared to the best supervised fusion method. Moreover, it is more flexible than early fusion methods for incorporating novel tag relevance estimators. We recommend LateFusion-Average as a practical mechanism to exploit diverse tag relevance estimates for social image search.

Chapter 4

Social Negative Bootstrapping for Visual Categorization

Which social images are informative negative examples? We propose in this chapter a simple yet effective approach: *social negative bootstrapping*. Compared to creating negative examples by random sampling, which is the de facto standard in literature, the proposed approach iteratively and adaptively selects informative samples from social-tagged images. We achieve more accurate visual categorization without the need of manually labeling any negatives*.

*Published in *the Proceedings of the ACM International Conference on Multimedia Retrieval* 2011 [68].

4.1 Introduction

Labeled examples are crucial to learn classifiers for visual categorization. To be precise, we need positive and negative examples with respect to a specific visual category. When the number of categories is large, obtaining labeled examples in an efficient way is essential. To that end, current research focuses on obtaining positive examples [28, 100]. In [28], for instance, the authors investigate online collaborative annotation. The authors in [100] gather positive examples by re-ranking web image search results. While intensive effort has been devoted to the positive examples, random sampling is the de facto standard for achieving the negatives [28, 54, 61, 87, 100, 121, 151].

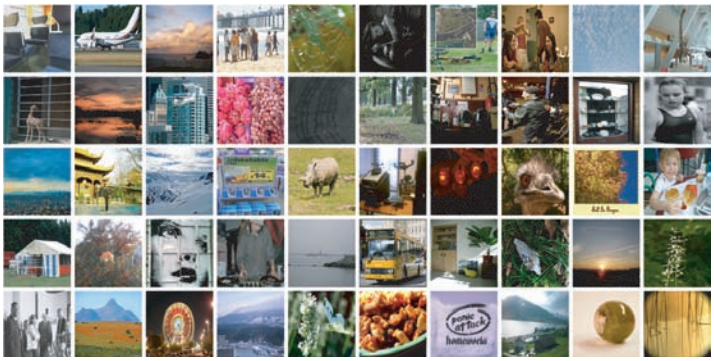
In practice, a classifier tends to misclassify negative examples which are visually similar to positive examples. As Fig. 4.1 shows, to derive an accurate classifier for a category, say 'aeroplane', confusing negatives such as images of birds or sky should be included when training classifiers. However, such informative negatives are unlikely to be hit by random sampling.

In this chapter we go beyond random sampling by proposing a novel, yet technically simple, *social negative bootstrapping* approach. Our approach conceptually bears some resemblance to active learning [118] and AdaBoost [36], as all of them seek informative examples for learning a new classifier. However, there are two notable differences between our approach and traditional active learning. First, in contrast to active learning which requires human interaction to label examples selected in each round, our approach selects informative negatives without human interaction. Second, our definition of informative examples differs from its counterpart in a typical active learning setting. There, one assumes that the input data consists of both positive and negative examples. Thus, examples the system is most uncertain about, namely closest to the decision boundary [118], are considered informative. Our approach, by contrast, selects negatives falling on the positive side and far away from the boundary. Compared to AdaBoost which works on fully labeled data, our approach is grounded on social-tagged data, without the need of manually labeling any negative examples. Since most annotation efforts to create fully labeled data are consumed by annotating the negatives [61], social negative bootstrapping is suited for exploiting large-scale datasets.

4.2 Related Work

This study is about sampling negative examples for visual categorization. But, it cannot stand alone without positive examples. So we first review recent progress in obtaining positive examples, and then discuss work on the negatives.

(a) Positive examples of 'aeroplane'



(b) Randomly sampled negative examples of 'aeroplane'

(c) Automatically generated negative examples (this chapter)

Figure 4.1: A positive set and two negative sets of the visual category 'aeroplane'. The negative set (b) is obtained by random sampling, while the negative set (c) is automatically generated by our approach. Note that, compared to (b), our negatives are visually more similar to the positive set (a). Hence, they are more informative, yielding more accurate visual classifiers. Such negatives are found *without the need of actually labeling them*.

4.2.1 Obtaining Positive Examples

Much research has been conducted towards devicing efficient solutions to acquire positive examples. E.g., by data-driven learning from web image search results [100, 119, 130, 142] or social-tagged data [64, 112, 121, 151], or by online collaborative annotation [28, 85, 95]. In [100], for instance, the authors train a visual classifier on web image search results of a given category, and re-rank the search results by the classifier. By estimating the relevance of user tags to image content [64], social-tagged data can be cleaned up. The authors in [95] develop an online annotation tool, letting web users label images as volunteers. Though the automated approaches are not

comparable to human annotation [54, 121], their output already gives a good starting point for manual labeling. Therefore, a recent trend is to combine data-driven learning and online annotation. For instance, the authors in [28] build an ImageNet wherein positive examples of a WordNet category [32] are obtained by labeling web image search results of the category using the Amazon Mechanical Turk service. In this service, web annotators are paid by micro payments. In a recent release of ImageNet, for almost 9,000 categories, there are at least five hundred positive examples per category. Compared to traditional expert labeling, the new labeling mechanism yields positive examples for many categories with lower cost. In this chpater we assume that positive examples are obtained by (one of) the approaches described above, and focus on obtaining negative examples.

4.2.2 Obtaining Negative Examples

Despite the achievement of gathering positive examples, the problem of how to effectively obtain the negatives remains unclear and its importance underestimated. One might consider bypassing the negative labeling problem by one-class learning, which creates classifiers using positive examples only [116]. However, as in principle learning from more information will lead to better results, visual classifiers trained by one-class learning are inferior to classifiers trained by two-class learning with randomly sampled negatives [61].

To automatically create a negative training set for a given category, the mainstream approach is to randomly sample a relatively small subset from a large pool of (social-tagged) examples [28,54,61,87,100,121,151]. Apart from the obvious fact that random sampling is simple and easy to use, we attribute its popularity to the following two reasons. First, as the possible negatives significantly outnumber the positive training set, down-sampling the negatives bypasses class imbalance which is known to affect classifier learning [49]. Second, except for some over-frequent categories such as 'sky' and 'person', the chance of finding genuine positive examples in a random fraction of the pool is low. If the pool is sufficiently large, one might end with a set of reliable negatives, but not necessarily the most informative ones.

Since negative examples are selected at random, the performance of individual classifiers may vary. According to the bootstrap aggregation theory [13], such variance can be reduced by model averaging. Hence, the authors in [87] perform random sampling multiple times to create multiple classifiers, and combine them uniformly. Although the robustness of the final classifier might be improved by classifier aggregation, such a "random+aggregation" approach seems not strategically better than random sampling.

Negative bootstrapping has also been studied in the context of text categorization, e.g., [73]. There, unlabeled examples are inserted into the negative set, if they are most dissimilar to the positives, or predicted as negatives with high confidence by the current classifier. A similar idea is reported in [140] for video retrieval, where negatives are selected at the bottom when ranking unlabeled examples by their scores of being

positives in descending order. Though sampling at the bottom probably yields reliable negatives, an intrinsic drawback is that those negatives are already correctly classified, adding them to the training process is not so useful by definition. Indeed, empirical evidence from [87] indicates that such conservative sampling is inferior to random sampling.

In this chapter, we strive to reveal the true value of social-tagged images as negative training examples. As a reward, we obtain visual classifiers which are more accurate than classifiers trained on randomly sampled negatives or their aggregated version.

4.3 Social Negative Bootstrapping

4.3.1 Problem Statement

Let x be a target image which we want to categorize, and V a large set of visual categories. Let S be a large set of images, where each image is labeled with at least one category from V by social tagging. Given a specific category $w \in V$, let B_{w+} be a positive training set, which are obtained, for instance, by the approaches described in Section 4.2.1. In a classical two-class learning setting, one has access to B_{w+} and a set of manually labeled negative examples. In random negative bootstrapping, manually labeled negatives are replaced by randomly sampled pseudo-negatives. Social negative bootstrapping derives a visual classifier $G(x, w)$ from B_{w+} and from B_{w-} which contains negative examples obtained from S. The output of $G(x, w)$ is a likelihood score of the image x being positive with respect to the category w. We aim for negative examples most informative to train classifiers, but without actually labeling any negatives.

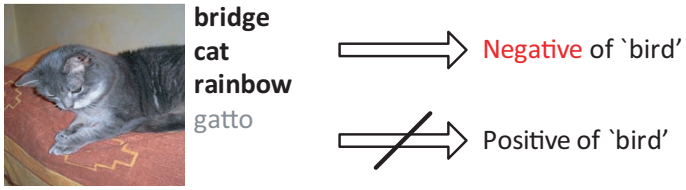
4.3.2 The Algorithm

For a given category w and a positive set B_{w+}, we adaptively and iteratively select informative negatives B_{w-} from S. In particular, we select the informative examples from those negatives having the highest probability of being misclassified. We detail our proposal as follows.

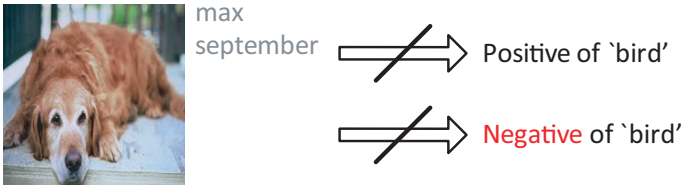
Virtual Labeling

For social-tagged data, even though user tags are often unreliable for identifying positive examples, we argue that they are reliable for determining negative examples, allowing us to construct an effective virtual labeling procedure exploiting tag statistics and semantics.

We base the virtual labeling procedure on our observation about social image tagging. User tags of an image may contain some visual categories, or they may contain no visual categories, as shown in Fig. 4.2. In both cases, determining the



(a) User tags contain visual categories



(b) User tags do not contain visual categories

Figure 4.2: Inferring negative examples of a specific visual category by tag reasoning. Tags corresponding to visual categories are marked by a bold font. If an image is labeled with visual categories, but not with a target category, say 'bird', or its semantically related tags, the image is likely to be a negative example of the category. Images in all other cases are not taken as positives, nor as negatives.

positiveness of the image to a given category w is difficult, due to the subjectiveness of social tagging. However, on a set of randomly sampled images we observe that if an image is labeled with visual categories, but not labeled with w or its semantically related tags, the image is likely to be a negative example of w. We illustrate this observation in Fig. 4.2(a).

To obtain a set of reliable negatives, we need to determine V_w, a tagging vocabulary the average user uses to depict the category w, where $V_w \subset V$. By simply excluding images labeled with tags from V_w, we will obtain a set of reliable negative examples. We use S_{w-} to represent the virtually labeled negative set,

$$S_{w-} \leftarrow virtual\ labeling(S, w). \tag{4.1}$$

Concerning general criteria for creating V_w, to cope with the diversity of user tags, we construct V_w as a set of tags semantically correlated to the category. Semantic correlation between tags can be measured, say, by tag co-occurrence in a large corpus [23] or by human knowledge [32]. Note that our virtual labeling is conducted in the tag space, rather than in the visual feature space wherein visual categorization is performed. As a consequence, we obtain reliable negatives, among which we expect sufficient samples which are informative for training classifiers.

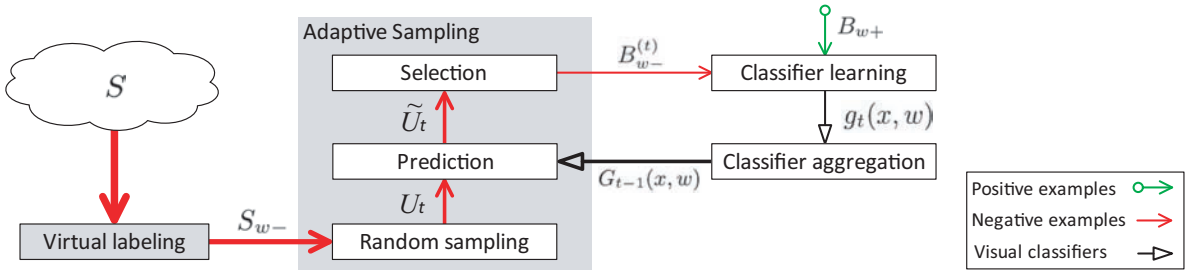


Figure 4.3: The proposed *social negative bootstrapping* approach. Given a specific visual category w and a positive set B_{w+}, we obtain a series of informative negative sets $\{B^{(t)}_{w-}\}$ from a large set of virtually labeled negative examples S_{w-} by multi-round adaptive sampling. In round t, we use $G_{t-1}(x,w)$ to classify a candidate set U_t, and select the most misclassified negatives to form $B^{(t)}_{w-}$. To initialize the bootstrapping process, $B^{(1)}_{w-}$ is randomly sampled from S_{w-}. By iteratively exploiting the informative negatives, we obtain visual classifiers with better discrimination ability, but without the cost of manually labeling any negatives.

Adaptive Sampling

The virtually labeled set S_{w-} is large, say having millions of images. Directly training classifiers on B_{w+} and S_{w-} is not only computationally challenging, but also suffering from extreme class imbalance. To make learning feasible, we iteratively exploit S_{w-} by performing multi-round learning. We use T to denote the number of learning rounds, and $t = 1, \ldots, T$ to index the rounds. In each round t, we adaptively select the most informative negative examples based on classifiers trained in previous rounds. To that end, we propose a two-stage adaptive sampling strategy. Suppose we have a classifier $G_t(x,w)$ obtained in the round t. In the first stage, we randomly sample n_u samples from S_{w-} to form a candidate set U_t,

$$U_t \leftarrow random\ sampling(S_{w-}, n_u). \tag{4.2}$$

To scale down the computational cost of selecting informative negatives from U_t and to reduce the chance of having genuine positives in U_t, we make $n_u \ll |S_{w-}|$. In the second stage, we use $G_{t-1}(x,w)$ to predict labels for each example in U_t, and obtain $\widetilde{U}_t$ in which each example is associated with a likelihood score of being positive to w,

$$\widetilde{U}_t \leftarrow predction(U_t, G_{t-1}(x,w)). \tag{4.3}$$

We consider examples which are *most misclassified*, i.e., predicted as positive with the largest scores, the *most informative* negatives. We rank examples in $\widetilde{U}$ by their scores in descending order and select the top ranked examples as the informative negative set found in the round t. We denote this negative set as $B^{(t)}_{w-}$. To bypass class imbalance, we enforce the number of the selected negatives to be equal to $|B_{w+}|$, namely

$$B^{(t)}_{w-} \leftarrow selection(\widetilde{U}_t, |B_{w+}|), \tag{4.4}$$

Table 4.1: The proposed social negative bootstrapping algorithm.

INPUT: visual concept w, expert-labeled positive examples B_{w+}, social-tagged examples S, and the number of learning rounds T.
OUTPUT: visual classifier $G_T(x, w)$.

1. **Creating negative example pool**:
   $S_{w-} \leftarrow \texttt{virtual-labeling}(S, w)$.
2. **Creating an initial classifier**:
   (a) $B_{w-}^{(1)} \leftarrow \texttt{random-sampling}(S_{w-}, |B_{w+}|)$.
   (b) $g_1(x, w) \leftarrow \texttt{classifier-learning}(B_{w+}, B_{w-}^{(1)})$.
   (c) $G_1(x, w) = g_1(x, w)$.
3. **For** $t = 2, \ldots, T$ **do**
   3.1 **Adaptive sampling**:
      (a) $U_t \leftarrow \texttt{random-sampling}(S_{w-}, n_u)$.
      (b) $\widetilde{U}_t \leftarrow \texttt{prediction}(U_t, G_{t-1}(x, w))$.
      (c) $B_{w-}^{(t)} \leftarrow \texttt{selection}(\widetilde{U}_t, |B_{w+}|)$.
   3.2 **Classifier learning**:
      $g_t(x, w) \leftarrow \texttt{classifier-learning}(B_{w+}, B_{w-}^{(t)})$.
   3.3 **Classifier aggregation**:
      $G_t(x, w) = \frac{t-1}{t} G_{t-1}(x, w) + \frac{1}{t} g_t(x, w)$.

where $|\cdot|$ denotes set cardinality. By repeating the adaptive sampling procedure, we iteratively select informative negatives from S_{w-} in an adaptive manner.

Classifier Learning and Aggregation

In each round t, we learn a new classifier $g_t(x, w)$ from B_{w+} and $B_{w-}^{(t)}$. As $B_{w-}^{(t)}$ is composed of negatives which are most misclassified by previous classifiers, we suppose that the new classifier is complementary to its ancestors. Therefore, we choose classifier aggregation to obtain the final classifier. Let $G_t(x, w)$ be an aggregated classifier which uniformly combines $g_t(x, w)$ and the previous t-1 classifiers:

$$G_t(x, w) = \frac{t-1}{t} G_{t-1}(x, w) + \frac{1}{t} g_t(x, w). \tag{4.5}$$

To trigger the bootstrapping process, we train an initial classifier $g_1(x, w)$ on B_{w+} and $B_{w-}^{(1)}$, which consists of examples randomly sampled from S_{w-}, with $|B_{w-}^{(1)}| = |B_{w+}|$.

We illustrate the entire framework in Fig. 4.3, with the algorithm given in Table 4.1. By adaptively selecting informative negative sets, social negative bootstrapping enables us to derive visual classifiers with better discrimination ability.

4.4 Experimental Setup

We compare the proposed approach with the following two types of baselines, both of which rely on random sampling to obtain negative training data: 1) "random sampling" [54, 61, 100, 121, 151], and 2) "random+aggregation" [87]. For a fair comparison, whenever applicable, we will make our approach and the baselines share the same input and parameters.

4.4.1 Data sets

Positive training set B_{w+}. We choose the PASCAL VOC 2008 training set [31], collected from Flickr, with expert-labeled ground truth for 20 visual categories. For each category, we randomly sample 50 positive examples as B_{w+}.

Social-tagged image set S. We construct S as follows. We create the visual category vocabulary V by taking the intersection between the ImageNet vocabulary [28] and a social tagging vocabulary in which each tag is used by at least 100 distinct users in a set of 10 million Flickr images. The size of V is 5,009. Next, we go through 3.5 million Flickr images[†] created in our previous work [64], and remove images batch-tagged or having no tags from V. We end with S consisting of 650K images.

Two test sets. To evaluate classifiers derived from the same training set but by different approaches, we adopt the following two test sets, which were created independently by manually labeling different subsets of Flickr images. For within-dataset visual categorization, we adopt the VOC2008 validation set [31]. To test the robustness of the proposed approach in a cross-dataset setting, we choose the NUS-OBJECT test set [**?**]. We present in Table 4.2 data statistics of the training and test sets.

4.4.2 Implementation

Image representation. Since vector-quantized keypoint descriptors are effective features for visual categorization, we follow this convention. In particular, we adopt dense sampling for keypoint localization and SURF [9] for keypoint description, using a fast implementation of dense-SURF [120]. With the SURF descriptors quantized by a codebook of 4000 bins, an image is represented by a 4000-dimensional feature which describes dominant structural patterns of that image.

Base classifiers. The proposed approach does not rely on specific classification models. Here we instantiate $g_t(x, w)$ using Support Vector Machine (SVM) for its good performance [123]. Since we do not aim for the best possible performance, but rather focus on the performance gain, we train two-class SVM classifiers using LIBSVM's default cost parameter [17], and the χ^2 kernel.

[†]Data available at `http://staff.science.uva.nl/~xirong/tagrel/`

Table 4.2: Statistics of the training and test sets used in our experiments. For each category w, we train classifiers on a small number of positive set B_{w+} and a large amount of social-tagged negative set S_{w-}.

	Training set		**Positives in test sets (%)**	
Category w	$\lvert B_{w+}\rvert$	$\lvert S_{w-}\rvert$	VOC08-val	NUS-OBJECT
aeroplane	50	521,010	5.3	4.8
bicycle	50	484,144	4.5	–
bird	50	395,079	6.3	5.8
boat	50	438,637	4.3	7.1
bottle	50	390,601	5.2	–
bus	48	511,708	2.3	–
car	50	383,319	10.1	3.5
cat	50	482,091	7.6	3.5
chair	50	327,967	8.0	–
cow	37	521,429	1.7	1.3
diningtable	50	484,960	2.4	–
dog	50	489,730	9.1	4.0
horse	50	525,110	4.6	2.6
motorbike	50	513,191	4.6	–
person	50	190,541	48.9	–
pottedplant	50	520,920	4.3	–
sheep	32	508,885	1.4	–
sofa	50	401,056	3.0	–
train	50	515,572	3.3	2.1
tv/monitor	50	228,876	4.9	–

Parameters of social negative bootstrapping. To create the social-tagged negative pool for a given category w, we compute the Normalized Google Distance (NGD) [23] between tags and w on the 10 million set. Tags whose distance to w is smaller than 1 are considered as semantically correlated to w. Notice that the tag 'face' is strongly correlated to 'person' related concepts, but it tends to be under-used in social tagging. Thus, if an image has faces detected by the Viola-Jones detector [124], we add 'face' to the tags of that image. We combine the correlated tags and childnodes of w in WordNet to form the correlated tag set,

$$V_w = \{w' \in V | NGD(w, w') < 1 \text{ or } w' \text{ is a WordNet childnode of } w\}. \quad (4.6)$$

For the size of the candidate set in each learning round, namely n_u in Eq. 4.2, we strike a balance between effectiveness and efficiency, with $n_u = 1000$ as our choice. We observe that the overall performance becomes stable after 50 learning rounds, therefore we set $T = 50$.

Parameters of the two negative sampling baselines. We use the same negative pool S_{w-} as used in the proposed approach for the two baseline approaches. In each learning round t, "random sampling" randomly selects $|B_{w+}|$ negative examples

from S_{w-} to train a classifier, while "random+aggregation" uniformly aggregates this classifier and the previous t-1 classifiers.

Given the parameter setting above, we train up to $20 \times 50 \times 2$=2,000 base classifiers in total.

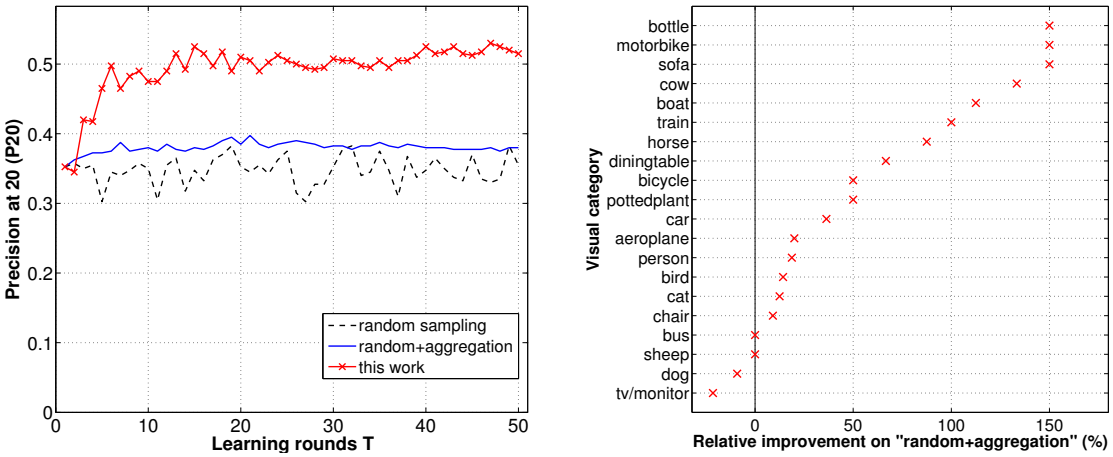
Evaluation criteria. For each visual category, we predict the presence of that category in a test image with a real-valued confidence score. Images in a test set are ranked according to their scores in descending order. To evaluate the performance, we adopt Precision at 20 (P20) to compare the top ranked results, and Average Precision (AP) for the whole ranked list.

4.5 Results

4.5.1 Comparing Different Approaches

As shown in Fig. 4.4(a), the proposed approach compares favorably to the baselines for ranking positive results at the top. In the first round, as no classifier is available, all approaches start with the same negative set $B^{(1)}_{w-}$, and consequently produce the same classifier $G_1(x, w)$. Afterwards, while the baseline approaches keep selecting random negatives, our approach starts seeking the most informative negatives. The "random sampling" approach is affected by the random factor in sampling, so its performance varies. The "random+aggregation" approach reduces such variance by combining classifiers. However, as the performance curves show, "random+aggregation", with a P20 score of 0.380 at T=50, can hardly go beyond the best performance of "random sampling", which is 0.383. The results clearly show the limitation of obtaining negatives by random sampling. In contrast, our approach reaches a P20 score of 0.513, which is 34.1% better than the best performance of "random sampling", and a 35.0% relative improvement over "random+aggregation". By adaptively and iteratively sampling the most informative negatives, we obtain visual classifiers with higher accuracy.

As shown in Fig. 4.4(b), for 10 out of the 20 categories, we obtain a relative improvement of at least 50% on "random+aggregation". Note that for four categories, i.e., 'bus', 'sheep', 'dog', and 'tv/monitor', our approach does not improve the baseline. For the category 'dog', we find that close-ups of flowers are frequently selected as the most informative negatives, and consequently, examples of other good negative classes, e.g., 'horse', are outnumbered. Probably because of such bias, the performance degenerates. For the category 'tv/monitor', images of rectangular objects are continuously selected. Classifiers trained on such negatives seem to be less powerful to separate the category from 'person', the most frequent category in the VOC08-val set. These results suggest that for certain categories, the diversity of negative training examples might be reduced to some extent. Nevertheless, our approach indeed correctly ranks the first result of the four categories, while the baseline fails. In general, when compared to random sampling, adaptive sampling results in more



(a) Overall comparison (b) Per-category comparison (T=50)

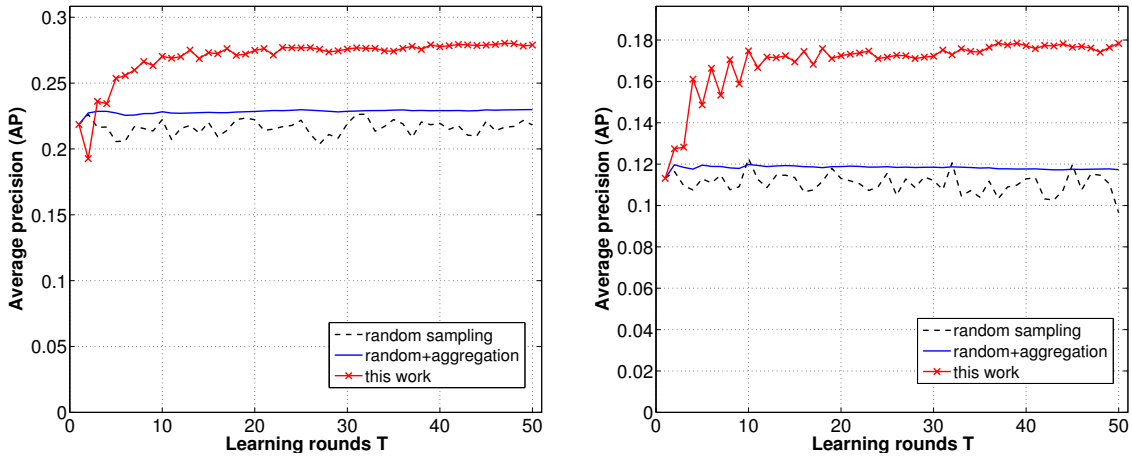
Figure 4.4: Within-dataset visual categorization. Test set VOC08-val. For 10 out of the 20 categories, classifiers trained by the proposed approach is at least 50% better in terms of Precision at 20. The considerable improvement is achieved without manually labeling any negative examples.

accurate visual classifiers.

The proposed approach is also effective for ranking an entire set, as shown in Fig. 4.5. Notice that our performance curve dips at T=2. This is because $g_2(x, w)$ is derived from $B^{(1)}_{w-}$, which are the most misclassified negatives by $g_1(x, w)$, and thus much distinct from generic negatives. Consequently, $g_2(x, w)$ is less effective for classifying generic negatives. Nevertheless, as subsequent classifiers are designed to be complementary to their ancestors, such ineffectiveness is tentative and will be resolved by adaptive sampling. When compared to "random+aggregation" with an AP score of 0.117, our approach reaches an AP score of 0.178 on NUS-OBJECT. The cross-dataset experiment shows the robustness of the proposed approach.

4.5.2 Examples

We show in Fig. 4.6 the most informative negative examples found by our approach. As we use the dense-SURF feature, negative examples visually close to the positives in terms of their structural patterns are predicted as informative for classifier training. See the categories 'cow', 'train', and 'bus' for instance. Because the examples are selected without manual verification, genuine positives may be included occasionally, see Fig. 4.6(c). Nevertheless, as they are in the minority, their impact on the bootstrapping process is minimal. Further, by visualizing the distribution of user tags in the selected negatives with a tag cloud, we see which negative classes are most informative to a certain category. We conjecture that such a relationship is feature-dependent, i.e., different features result in different informative negatives



(a) Test set VOC08-val (b) Test set NUS-OBJECT

Figure 4.5: Cross-dataset visual categorization. The proposed approach is not only superior to the baselines for ranking an entire set, but also effective in the cross-dataset setting.

for the same category. Moreover, we observe that the informativeness relationship between categories seems to be asymmetric. For instance, while 'bus' and 'car' are most informative to 'train', the most informative negative class for 'bus' and 'car' is 'firetruck', rather than 'train'. A plausible explanation is that 'firetruck' bears more resemblance to the former two categories in terms of properties of the object, e.g., rectangular curves, and visual context, in casu, street. In sum, the qualitative results in Fig. 4.6 further illustrate the effectiveness of the proposed approach in finding negative examples informative for learning visual classifiers.

4.6 Conclusions

In this chapter we study how to sample informative negative examples from widely available social-tagged images for visual categorization. To that end, we propose the *social negative bootstrapping* approach. Our major findings are as follows. Negative examples can be obtained, with no human interaction, by the designed virtual labeling procedure which exploits tag statistics and semantics. Virtual labeling, in combination with adaptive sampling, allows us to harvest informative negatives from those negatives having the highest probability of being misclassified. When compared to classifiers trained on randomly sampled negatives, classifiers derived from such informative negatives have better discrimination ability. The proposed approach is thus strategically better than random negative bootstrapping.

Experiments on two image benchmarks and 650K virtually labeled negative examples verify our proposal. For the majority of visual categories, we obtain a relative improvement of at least 50%, in terms of precision at 20. Moreover, cross-dataset

(a) Negatives of 'cow'

(b) Negatives of 'train'

(c) Negatives of 'car'

(d) Negatives of 'bus'

Figure 4.6: The 80 most informative negative examples, found by the proposed approach, for specific visual categories. By visualizing the distribution of user tags in the selected negatives as a tag cloud, we see which negative classes are most informative to a given category.

visual categorization shows the robustness of the proposed approach. Notice that the substantial progress is achieved without the need of labeling any negative examples. As the promising results suggest, social negative bootstrapping opens up interesting avenues for future research.

Part II: Online Use

Chapter 5

Harvesting Social Images for Bi-Concept Search

How to exploit social-tagged images for complex visual search? We introduce in this chapter the notion of bi-concepts, a retrieval method for two concepts that is directly learned from social data. To materialize, we propose a bi-concept image search engine which integrates the techniques developed in the previous chapters*.

*Submitted [66].

5.1 Introduction

Searching pictures on smart phones, PCs, and the Internet for specific visual concepts, such as objects and scenes, is of great importance for users with all sorts of information needs. As the number of images is growing so rapidly, full manual annotation is unfeasible. Therefore, automatically determining the occurrence of visual concepts in the visual content is crucial. Compared to low-level visual features such as color and local descriptors used in traditional content-based image retrieval, the concepts provide direct access to the semantics of the visual content. Thanks to continuous progress in generic visual concept detection [57, 108, 128, 129], followed by novel exploitation of the individual detection results [25, 43, 107, 132], an effective approach to unlabeled image search is dawning.

In reality, however, a user's query is often more complex than a single concept can represent [146]. For instance consider the query: "an image showing a horse next to a car". To answer this query, one might expect to employ a 'car' detector and a 'horse' detector, and combine their predictions, which is indeed the mainstream approach in the literature [2, 43, 69, 86, 107, 132]. But is this approach effective? We observe that the single concept detectors are trained on typical examples of the corresponding concept, e.g., cars on a street for the 'car' detector', and horses on grass for the 'horse' detector. We hypothesize that images with horses and cars co-occurring also have a characteristic visual appearance, while the individual concepts might not be present in their common form. Hence, combining two reasonably accurate single concept detectors is mostly ineffective for finding images with both concepts visible, as illustrated in Fig. 5.1.

Ideally, we treat the combination of the concepts as a new concept, which we term *bi-concept*. To be precise, we define a bi-concept as the co-occurrence of two distinct visual concepts, where its full meaning cannot be inferred from one of its component concepts alone. According to this definition, not all combinations of two concepts are bi-concepts. For instance, a combination of a concept and its superclass such as 'horse + animal' is not a bi-concept, because 'horse + animal' bears no more information than 'horse'. Besides, specialized single concepts consisting of multiple tags such as 'white horse' [34, 143] and 'car driver' are not bi-concepts as the two tags refer to the same visual concept. The same holds for events such as "airplane landing" where the tag landing is not a distinct visual concept by itself.

Although not all semantic combinations are bi-concepts, the number of possible bi-concepts is still quadratic to the number of single concepts. Even when we assume that a set of only 5,000 concepts is enough for general purpose search [43], finding sufficient labeled examples for each possible bi-concept already becomes a problem of big proportion. The amount of labeling effort is so considerable that it puts the scalability of expert annotation and the recent Amazon Mechanical Turk service into question. We consider obtaining bi-concept examples without expert labeling as a key problem for bi-concept search in unlabeled images.

A novel source of labeled images for concept detection are user-tagged images on

(a) Searching for 'car' by a 'car' detector

(b) Searching for 'horse' by a 'horse' detector

(c) Searching for 'car + horse' by combining the results of (a) and (b)

(d) Searching for 'car + horse' using the proposed bi-concept search engine

Figure 5.1: Searching for two visual concepts co-occurring in unlabeled images. A (green) tick indicates a positive result. Given two single concept detectors with reasonable accuracy, a combination using their individual confidence scores yields a bad retrieval result (c). We propose to answer the complex query using a bi-concept detector optimized in terms of mutual training examples (d).

the social web such as those on Flickr and Facebook. However, due to the subjectiveness of social tagging, social tags often do not reflect the actual content of the image. It has been shown in previous studies that directly training on social-tagged images results in suboptimal single concepts detectors [54, 121, 151]. Learning bi-concept detectors from social-tagged images is, to the best of our knowledge, non-existing in the literature. By definition, the number of images labeled with a bi-concept is less than the number of images labeled with a single concept, meaning bi-concepts have a worse starting point for obtaining examples. In addition, a scene with two concepts present tends to be visually more complex, requiring multi-modal analysis. Given these difficulties, effective bi-concept search demands an approach to harvesting appropriate examples from social-tagged images for learning bi-concept detectors.

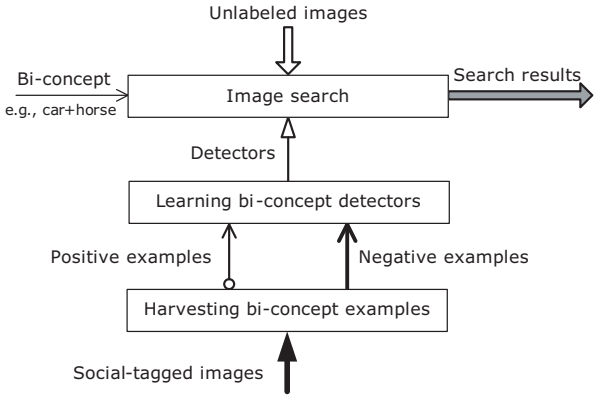


Figure 5.2: A conceptual diagram of the proposed bi-concept image search engine.

In this chapter, we introduce the notion of bi-concepts, and propose a multimedia search engine to study their instantiation, development, and applicability. We present a multi-modal approach to collect de-noised positive as well as informative negative training examples from the social web. We learn bi-concept detectors from these examples, and later apply them for retrieving bi-concepts in unlabeled images. A conceptual diagram of the proposed search engine is illustrated in Fig. 6.1.

5.2 Related work

We first review related work on combining single concept detectors for visual search, and then discuss recent progress on harvesting training examples from the (social) web. For the ease of consistent description, we use x to represent an image, w for a single concept, and $\mathbf{w}_{1,2}$ for a bi-concept comprised of two single concepts w_1 and w_2. We use $p(w|x)$ to represent a concept detector which produces a posterior probability of observing w given the image. In a similar fashion, we define a bi-concept detector $p(\mathbf{w}_{1,2}|x)$.

5.2.1 Visual Search by Combining Single Concepts

Given hundreds of single concept detectors trained on well-labeled examples [85], a considerable amount of papers have been published on how to combine the detectors for visual search. We refer to the review paper by Snoek and Worring [109] for a comprehensive discussion. Here we discuss two effective, simple and popular combination methods: the product rule [2] and linear fusion [19, 69, 87, 107, 132, 141].

If the assumption would hold that individual concepts are conditionally independent given the image content [2], a bi-concept detector can be approximated by the product of its component concepts, namely

$$p(\mathbf{w}_{1,2}|x) = p(w_1|x) \cdot p(w_2|x). \tag{5.1}$$

The linear fusion version of the bi-concept detector is commonly expressed as

$$p(\mathbf{w}_{1,2}|x) = \lambda \cdot p(w_1|x) + (1 - \lambda) \cdot p(w_2|x), \tag{5.2}$$

where $\lambda \in [0, 1]$ is a weighting parameter. To automatically determine the weight, many have used a heuristic approach. Natsev et al. [87] suggest average fusion with $\lambda = 0.5$. Chang et al. [19] weight the individual single concept detectors in terms of their training performance. However, Snoek et al. [107] argued that the best performing individual detectors such as 'person' and 'outdoor' are often the least informative for retrieval. So Li et al. [69] set λ to be proportional to the informativeness of the concepts. How to determine the proper combination of the single concept detectors remains a open question [109]. It is the reason why many resort to an oracle combination using the best possible weights [43, 87, 107].

One may also consider utilizing an object localization system which relies on region-based image analysis to pinpoint regions of the single concepts [96]. Such a system involves image segmentation, a challenging problem in computer vision. Moreover, Training the system requires learning examples labeled at a region level, which are more expensive to obtain than global level annotations. In contrast, we are interested in searching for bi-concepts by holistic analysis, based on the observation that examples with two concepts co-occurring have a characteristic holistic scene. Moreover, we obtain training data without the need of manual annotation other than using social tags.

5.2.2 Harvesting Training Data from the (Social) Web

Obtaining training examples from the web with expert annotation for free is receiving much attention recently, with sources ranging from generic web images [58, 100, 119, 131, 142], professional photo forums [77], to social-tagged images [54, 68, 99, 103, 121]. Training data consists of positive and negative image examples for a given concept. Therefore, we discuss work on positive examples and on negative examples, respectively.

Harvesting Positive Examples. Given a single concept as a textual query, Yanai and Barnard [142] and Li et al. [58] collect positive examples by re-ranking web image retrieval results using probabilistic models derived from the initial search results. Since the amount of returned examples is limited by the image search engine used, Schroff et al. [100] propose to directly extract images from web search results. As the images vary in quality and come with noisy annotations, dedicated preprocessing such as filtering of drawings and symbolic images is required. The remaining top ranked images are treated as positive examples together with randomly sampled images as negative examples. Based on these examples an SVM classifier is trained and then applied for image re-ranking. As an alternative, Liu et al. [77] rely on a professional photo forum for harvesting training examples, where image quality is considered higher and the annotations are better [148].

In contrast to web images loosely connected with free text, images on the social web are described by user-contributed tags. Moreover, one has access to social-tagged images without any constraint on the amount, making social-tagged images an appealing source for harvesting positive examples. Kennedy et al. [54] and Ulges et al. [121] consider images labeled with a certain concept as positive examples. However, due to the subjectiveness of social tagging, the accuracy of such social positive examples varies per concept [54]. To improve the social tagging accuracy, a number of methods have been proposed, ranging from semantic analysis [151], visual analysis [65] to multi-modal analysis [74, 149]. Zhu et al. in [151] estimate the relevance of a given single concept with respect to an image by measuring the semantic consistency between the concept and the image's social tags. In our previous work [65], we proposed UniformTagger, which estimates image tag relevance by a uniform fusion of neighbor voting results driven by diverse visual features. As determining the relevance for a bi-concept is more difficult than its single concept counterpart, combining textual and visual analysis seems important for obtaining bi-concept examples. Multi-modal analysis by jointly exploiting image-wise visual similarity and tag-wise semantic similarity is considered by Zhu et al. [149] and Liu et al. [74]. As these methods require matrix analysis on the entire image collection and the whole tag vocabulary, their scalability for exploiting a large amount of social-tagged images is questionable.

Harvesting Negative Examples. Surprisingly, in contrast to extensive research on positive examples, the importance of negative examples is often overlooked. The mainstream approach is to randomly sample a relatively small subset from a large pool of images [54, 100, 121, 151]. For instance, Kennedy et al. [54] and Ulges et al. [121] construct a negative set of a fixed size for a given single concept, by randomly sampling from examples not labeled with the concept. If the pool is sufficiently large, one might end up with a set of reliable negatives, but not necessarily the most informative ones.

For bi-concepts, negative examples are even more important as one not only has to distinguish the bi-concept from 'normal' negative classes, but also from its component single concepts. In a labeled image re-ranking context, Allan and Verbeek [1] suggest to insert examples of the component concepts into the negative set, from which they train an image re-ranking model. In our previous work [68], we proposed a social negative bootstrapping approach to adaptively and iteratively sample informative examples for single concepts, with a prerequisite that manually labeled positive examples are available. However, the prerequisite is unlikely to exist for the bi-concept case.

Given the related work, we consider the absence of the notion of bi-concepts as a major problem for multi-concept search in unlabeled data. The lack of bi-concept training examples is a bottleneck for learning bi-concept detectors. Previous work on harvesting single-concept examples from social images including our earlier work [65, 68] yields a partial solution, but needs to be reconsidered for bi-concept learning.

5.3 Bi-Concept Image Search Engine

To make the new notion of bi-concept explicit, we study its characteristics in a bi-concept image search engine for unlabeled data. To search for a specific bi-concept $\mathbf{w}_{1,2}$ in the unlabeled data, we first harvest bi-concept examples from social-tagged images, namely positive examples in Section 5.3.1 and negative examples in Section 5.3.2. Our choice of the bi-concept detector $p(\mathbf{w}_{1,2}|x)$ is explained in Section 5.3.3. Finally, we obtain image search results by sorting the unlabeled collection in descending order by $p(\mathbf{w}_{1,2}|x)$.

5.3.1 Harvesting Bi-Concept Positive Examples

In order to obtain accurate positive examples for a bi-concept $\mathbf{w}_{1,2}$, we need a large set of social-tagged images and a means to estimate the relevance of a bi-concept with respect to an image. Let X_{social} indicate such a large set, and let $X_{\mathbf{w}_{1,2}+}$ be images in X_{social} which are simultaneously labeled with w_1 and w_2. To simplify our notation, we also use the symbol w to denote a social tag. We define $g(x, w)$ as a single-concept relevance estimator, and $g(x, \mathbf{w}_{1,2})$ an estimator for bi-concepts. Finally, we denote $\mathbf{w}_x$ as the set of social tags assigned to an image.

We choose two state-of-the-art methods originally designed for the single-concept problem. One method uses semantic analysis [151], and the other method is our previous work, using multi-feature visual analysis [65]. We adapt them to the bi-concept problem: estimating the co-relevance of two single concepts with respect to an image.

The Semantic Method. Given the assumption that the true semantic interpretation of an image is reflected best by the majority of its social tags, a tag that is semantically more consistent with the majority is more likely to be relevant to the image [151]. We express the semantic-based relevance estimator for single concepts as

$$g_s(x, w) = \frac{\sum_{w' \in \mathbf{w}_x} sim(w', w)}{|\mathbf{w}_x|}, \tag{5.3}$$

where $sim(w', w)$ denotes semantic similarity between two tags, and $|\cdot|$ is the cardinality of a set. Zhu et al. [151] interpret $sim(w', w)$ as the likelihood of observing w' given w. To cope with variation in tag-wise semantic divergence, we use

$$sim(w', w) = \exp(-\frac{d^2(w', w)}{2\sigma^2}), \tag{5.4}$$

where $d(w', w)$ measures a semantic divergence between two tags, and the variable σ is the standard derivation of the divergence. Note that (5.3) is not directly applicable for bi-concepts. To address the issue, we adopt a similarity measure intended for two short text snippets [83], and derive our semantic-based relevance estimator as

$$g_s(x, \mathbf{w}_{1,2}) = \frac{\sum_{w' \in \mathbf{w}_x} maxSim(w', \mathbf{w}_{1,2}) \cdot idf(w')}{\sum_{w' \in \mathbf{w}_x} idf(w')}, \tag{5.5}$$

where $maxSim(w', \mathbf{w}_{1,2})$ is the maximum value of $sim(w', w_1)$ and $sim(w', w_2)$, and $idf(w')$ is the inverse image frequency of w', reflecting the tag' informativeness.

The Visual Method. Given an image x represented by visual feature f, we first find k nearest neighbors of the image from X_{social}, and estimate the relevance of every single concept w to x in terms of the concept's occurrence frequency in the neighbor set. To overcome the limitation of single features in describing the visual content, tag relevance estimates based on multiple features are uniformly combined [65]. We express the visual-based single-concept relevance estimator as

$$g_v(x, w) \quad = \frac{1}{|F|} \sum_{f \in F} \left(\frac{c(w, X_{x,f,k})}{k} - \frac{c(w, X_{social})}{|X_{social}|} \right), \tag{5.6}$$

where F is a set of features, $c(w, X)$ is the number of images labeled with w in an image set X, and $X_{x,f,k}$ is the neighbor set of x, with visual similarity measured by f.

A straightforward solution to compute (6.10) for a bi-concept $\mathbf{w}_{1,2}$ is to view the bi-concept as a new tag. This solution boils down to counting the number of images labeled with both w_1 and w_2 in the neighbor set $X_{x,f,k}$. These images are relatively sparse when compared to images labeled with either of w_1 and w_2. The estimator is accurate, but unreliable because $c(\mathbf{w}_{1,2}, X_{x,f,k})$ is often zero or very small. Combining relevance estimates of the two single concepts by linear fusion as described in Section 5.2.1 is also problematic, because determining a proper weight is difficult. Simply averaging $g(x, w_1)$ and $g(x, w_2)$ is reliable, yet less accurate. Hence, we need an estimator which accurately reflects the co-relevance of a bi-concept, and can be computed in a more reliable manner than the straightforward solution. Note the following inequality:

$$c(\mathbf{w}_{1,2}, X) \leq \min\{c(w_1, X), c(w_2, X)\} \leq \frac{1}{2}(c(w_1, X) + c(w_2, X)). \tag{5.7}$$

In practice the inequality is mostly strict. This means that if we compute $g_v(x, \mathbf{w}_{1,2})$ as the minimum value of $g_v(x, w_1)$ and $g_v(x, w_2)$, the value will be larger than the output of the straightforward solution, and smaller than the output of averaging $g_v(x, w_1)$ and $g_v(x, w_2)$. Moreover, the genuine occurrence of a bi-concept is always lower than any of the two concepts making up the bi-concept. Based on the above discussion, we choose the min function to balance the reliability and the accuracy for bi-concept relevance estimation. Consequently we define our visual-based bi-concept relevance estimator as

$$g_v(x, \mathbf{w}_{1,2}) = \min\{g_v(x, w_1), g_v(x, w_2)\}. \tag{5.8}$$

An advantage of (5.8) is that once we have single-concept relevance pre-computed, bi-concept relevance can be rapidly calculated.

Multi-modal: Semantics + Visual. As the semantic method and the visual method are orthogonal to each other, it is sensible to combine the two methods for

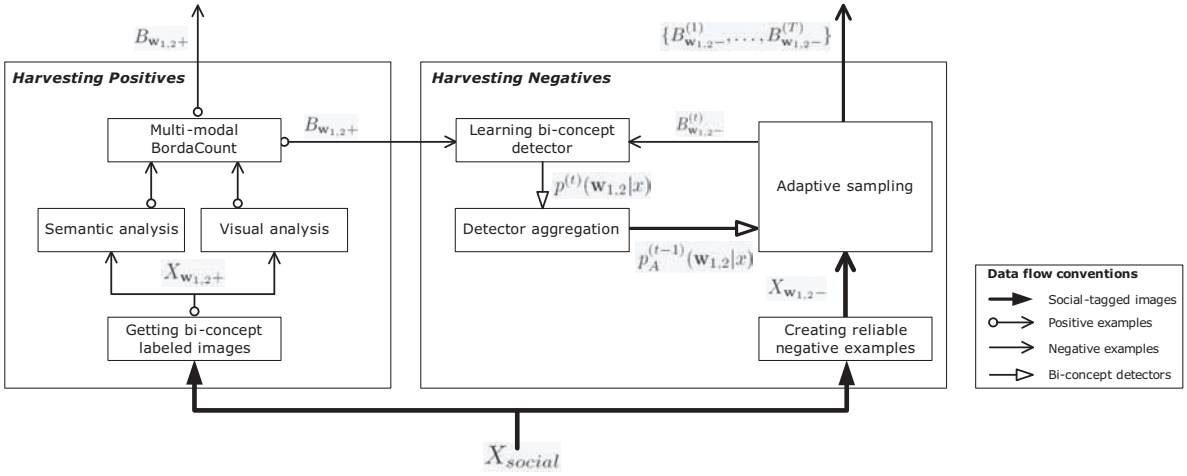


Figure 5.3: Harvesting bi-concept training examples from social-tagged images: A zoom-in view of the "Harvesting bi-concept examples" component in Fig. 6.1. We exploit multi-modal analysis to harvest accurate positive examples (Section 5.3.1), and adaptive sampling for informative negative examples (Section 5.3.2). Other than widely available social-tagged images, the process requires no manual annotation.

obtaining bi-concept examples with higher accuracy. As the outputs of $g_s(x, \mathbf{w}_{1,2})$ and $g_v(x, \mathbf{w}_{1,2})$ reside at different scales, normalization is necessary before combining the two functions. Since Borda Count is well recognized as a solid choice for combing rankings generated by multiple sources of evidence [4,89], we adopt the Borda method for our bi-concept search engine. Given a bi-concept $\mathbf{w}_{1,2}$, we first sort $X_{\mathbf{w}_{1,2}+}$ in descending order by $g_s(x, \mathbf{w}_{1,2})$ and $g_v(x, \mathbf{w}_{1,2})$, respectively. We then aggregate the two rankings by Borda Count to obtain a final ranking. We preserve the top ranked images as positive training examples for the bi-concept detector, denoted as $B_{\mathbf{w}_{1,2}+}$.

Next, we will use $B_{\mathbf{w}_{1,2}+}$ in combination with adaptive sampling to harvest informative negative examples from social-tagged images.

5.3.2 Harvesting Bi-Concept Negative Examples

Due to the relatively sparse occurrence of a bi-concept, random sampling already yields a set of accurate negatives. Harvesting negative examples for bi-concepts seems trivial. However, to create an accurate bi-concept detector, we need informative negatives which gives the detector better discrimination ability than the random negatives can contribute. We hypothesize that for a given bi-concept, its informative negatives have visual patterns overlapping the patterns of its positive instances. Following this thought, one might consider positive examples of the individual concepts informative. However, how a bi-concept detector actually works in visual feature spaces with thousands of dimensions is not necessarily consistent with what a human might expect. Given the considerable amount of bi-concepts, it is also impossible to prescribe proper

informative negative classes for each bi-concept, say by intensive domain knowledge. Therefore, we leverage the Social Negative Bootstrapping approach proposed in our earlier work [68], and adapt it to the bi-concept search problem. The approach, as detailed next, selects informative negatives from the viewpoint of a detector, but without the need of any human interaction.

Creating Reliable Negative Examples. For a given bi-concept $\mathbf{w}_{1,2}$, we first create a set of reliable negative examples, denoted as $X_{\mathbf{w}_{1,2}-}$, by simple tag reasoning. To describe the procedure, let V_w be a tag set comprised of synonyms and child nodes of w in WordNet [32]. For each image in X_{social}, if the image is not labeled with any tags from $V_{w_1} \cup V_{w_2}$, we add it into $X_{\mathbf{w}_{1,2}-}$.

Adaptive Sampling. Informative negatives are iteratively selected from $X_{\mathbf{w}_{1,2}-}$ by a multiple-round adaptive sampling strategy. Let T be the number of sampling rounds, and $t = 1, \ldots, T$ the index of a specific round. We denote with $p_A^{(t)}(\mathbf{w}_{1,2}|x)$ a bi-concept detector obtained after t rounds. In round t, we first randomly sample n_u samples from $X_{\mathbf{w}_{1,2}-}$ to form a candidate set U_t,

$$U_t \leftarrow \texttt{random-sampling}(X_{\mathbf{w}_{1,2}-}, n_u). \tag{5.9}$$

Then, we use $p_A^{(t-1)}(\mathbf{w}_{1,2}|x)$ to score each example in U_t, and obtain $\widetilde{U}_t$ in which each example is associated with a confidence score of being positive to the bi-concept,

$$\widetilde{U}_t \leftarrow \texttt{prediction}(U_t, p_A^{(t-1)}(\mathbf{w}_{1,2}|x)). \tag{5.10}$$

We consider examples which are *most misclassified*, i.e., wrongly predicted as positive with the largest confidence scores, the *most informative* negatives. So we rank examples in $\widetilde{U}_t$ by their scores in descending order and preserve the top ranked examples as the informative negative set found in round t. We denote this new negative set as $B_{\mathbf{w}_{1,2}-}^{(t)}$. By repeating the adaptive sampling procedure, we incrementally select informative negatives from social-tagged images in an adaptive manner.

Learning a New Bi-Concept Detector. In each round t, we learn a new detector $p^{(t)}(\mathbf{w}_{1,2}|x)$ from $B_{\mathbf{w}_{1,2}+}$ and $B_{\mathbf{w}_{1,2}-}^{(t)}$. To prevent class imbalance which often hampers classifier learning [49], we enforce the size of $B_{\mathbf{w}_{1,2}-}^{(t)}$ to be equal to $|B_{\mathbf{w}_{1,2}+}|$.

Detector Aggregation. As $B_{\mathbf{w}_{1,2}-}^{(t)}$ is composed of negatives which are most misclassified by the previous classifier, we consider new detector $p^{(t)}(\mathbf{w}_{1,2}|x)$ complementary to $p_A^{(t-1)}(\mathbf{w}_{1,2}|x)$. Therefore, we use model average to aggregate the two detectors to obtain the final detector:

$$p_A^{(t)}(\mathbf{w}_{1,2}|x) = \frac{t-1}{t} p_A^{(t-1)}(\mathbf{w}_{1,2}|x) + \frac{1}{t} p^{(t)}(\mathbf{w}_{1,2}|x). \tag{5.11}$$

We illustrate the proposed framework for harvesting bi-concept training data in Fig. 5.3. We first collect positive examples, and then start the social negative bootstrapping process for obtaining informative negative examples. To trigger the bootstrapping process, we train an initial detector $p^{(1)}(\mathbf{w}_{1,2}|x))$ on $B_{\mathbf{w}_{1,2}+}$ and $B_{\mathbf{w}_{1,2}-}^{(1)}$,

which consists of examples randomly sampled from $X_{\mathbf{w}_{1,2}-}$. We cache bi-concept detectors trained in the bootstrapping process so that we do not have to re-train a detector after training data collected. We use the aggregated detector $p_A^{(T)}(\mathbf{w}_{1,2}|x)$ as the input detector for the "Image search" component in Fig. 6.1. Given a collection of unlabeled images, our search engine sorts the collection in descending order by $p_A^{(T)}(\mathbf{w}_{1,2}|x)$, and returns the top-ranked results.

5.3.3 Learning Bi-Concept Detectors

To learn a concept detector $p^{(t)}(\mathbf{w}_{1,2}|x)$ using the positive set $B_{\mathbf{w}_{1,2}+}$ and the informative negative set $B^{(t)}_{\mathbf{w}_{1,2}-}$, we follow the standard procedure form the literature. Namely bag-of-keypoints features [122] plus SVM classifiers [123]. We extract Dense-SIFT features, i.e., dense sampling to localize keypoints and SIFT as a keypoint descriptor, using the state-of-the-art [122]. With the SIFT descriptors quantized by a precomputed codebook, each image is represented by a histogram with its length equal to the size of the codebook. Each bin of the histogram corresponds a certain code, and its value is the l_1-normalized frequency of the code extracted from the image. Let $h^{(t)}(x, \mathbf{w}_{1,2})$ be an SVM decision function trained on $B_{\mathbf{w}_{1,2}+}$ and $B^{(t)}_{\mathbf{w}_{1,2}-}$. To convert SVM decision values into posterior probabilities, we adopt a sigmoid function

$$p^{(t)}(\mathbf{w}_{1,2}|x) = \frac{1}{1 + \exp(a \cdot h^{(t)}(x, \mathbf{w}_{1,2}) + b)}, \tag{5.12}$$

where a and b are two real-valued parameters optimized by solving a regularized maximum likelihood problem as described in [71].

5.4 Experimental setup

5.4.1 Dataset Construction

Bi-Concepts. In order to evaluate the proposed bi-concept image search engine, we need to specify a list of bi-concepts for our experiments. Since searching for single concepts in unlabeled images remains challenging, the single concepts in a prospective bi-concept shall be detected with reasonable accuracy, otherwise searching for the bi-concept is very likely to be futile. Also, there shall be a reasonable amount of social-tagged training images, say thousands, labeled with the bi-concept. Bearing these considerations in mind, we choose three daily concepts commonly used in the literature [22, 48, 85, 105, 109], namely: 'beach', 'car', and 'flower'. We obtain bi-concepts by combining the concepts with other objects and scenes, resulting in the following 15 bi-concepts: 'beach + bird', 'beach + boat', 'beach + car', 'beach + girl', 'beach + horse', 'bird + flower', 'bird + snow', 'car + flower', 'car + horse', 'car + showroom', 'car + street', 'car + snow', 'cat + flower', 'cat + snow', and 'girl +

Table 5.1: Experiment 1. Comparing methods for harvesting positive training examples of single concepts, measured in terms of precision at 100. We sort the concepts by their frequency in the 1.2 million set in descending order. A gray cell indicates the top performer.

Concepts		Social Tagging Baselines				Proposed Search Engine		
w	*Frequency*	*Random*	*DateUploaded*	*Views*	*TagNum*	*Semantics*	*Visual*	*Borda*
car	71,367	0.69	0.75	0.87	0.61	0.85	**1.00**	0.99
flower	64,233	0.79	0.69	0.64	0.94	0.95	**1.00**	**1.00**
street	61,877	0.52	0.55	0.66	0.42	0.47	**1.00**	0.96
beach	47,636	0.53	0.53	0.69	0.59	0.63	**0.97**	0.95
snow	42,327	0.82	0.85	0.77	0.73	0.90	**1.00**	0.99
bird	33,841	0.79	0.80	0.67	0.94	0.92	**1.00**	0.99
girl	32,983	0.75	0.75	0.91	0.79	0.85	**0.97**	0.94
horse	28,724	0.70	0.60	0.74	0.79	0.85	**1.00**	**1.00**
cat	19,712	0.67	0.68	0.56	0.82	0.96	0.99	**1.00**
boat	15,239	0.75	0.75	0.74	0.76	0.85	0.94	**0.97**
showroom	4,947	0.43	0.43	0.61	0.34	0.34	**0.95**	0.78
MEAN	38,444	0.68	0.67	0.71	0.70	0.78	**0.98**	0.96

horse'. While the list of potential bi-concepts is exhaustive, the selection serves as a nontrivial illustration of bi-concept possibilities.

Social Source for Harvesting Training Examples. We use the 15 bi-concepts as well as the 11 single concepts from which they are composed as queries to randomly sample images uploaded on Flickr between 2006 and 2010. We remove batch-tagged images due to their low tagging accuracy, and obtain 300K images in total. To harness a large data set for multi-modal analysis, we further gather 900K social-tagged images from Flickr in a random fashion. Our total training collection thus contains 1.2 million images. We list the single concept and bi-concept statistics in Table 5.1 and Table 5.2.

Test Data. For each bi-concept, we create a ground truth positive set by manually checking images labeled with the bi-concept in the 1.2M set, and randomly selecting 50 positively labeled examples. Although the test images are associated with social tags, we ignore the tags and treat the images as unlabeled. Note that these selected examples are held-out from the training process. We supplement the collection of bi-concept images with distracter images from the publicly available NUS-WIDE set [22], which is also from Flickr but independent of our training data. Since this set was constructed by single-concept queries, it rarely contains genuine positives of the 15 bi-concepts. For reasons of efficiency, we randomly sample a subset of 10K images from NUS-WIDE as our negative test data. For each bi-concept, we examine how its 50 positive examples are ranked within the 10K negative set.

5.4.2 Experiments

In order to provide a step-by-step analysis on the entire bi-concept search framework, we evaluate in the following two experiments: the accuracy of harvested positive

Table 5.2: Experiment 1. Comparing methods for harvesting positive training examples of bi-concepts, measured in terms of precision at 100. We sort the bi-concepts by their frequency in the 1.2 million set in descending order.

Bi-Concepts			Social Tagging Baselines				Proposed Search Engine			
w_1	w_2	*Frequency*	*Random*	*DateUploaded*	*Views*	*TagNum*	*Semantics*	*Visual*	*Multi-kernel*	*Borda*
car	*street*	22788	0.64	0.57	0.70	0.53	0.58	**0.97**	0.23	0.86
car	*snow*	7109	0.62	0.62	0.54	0.68	0.66	**0.91**	0.67	0.85
beach	*car*	5432	0.10	0.14	0.11	0.19	0.25	0.27	0.42	**0.43**
car	*flower*	3604	0.13	0.11	0.05	0.39	0.40	0.21	0.37	**0.42**
beach	*girl*	3507	0.29	0.36	0.60	0.53	0.57	0.60	**0.78**	0.76
beach	*bird*	3093	0.26	0.25	0.20	0.51	0.57	0.56	**0.68**	0.63
beach	*boat*	2659	0.42	0.36	0.50	0.62	**0.66**	0.39	0.53	0.58
cat	*flower*	2316	0.07	0.05	0.05	0.55	**0.59**	0.14	0.55	0.42
bird	*flower*	2103	0.11	0.10	0.11	0.38	0.43	0.36	0.41	**0.51**
car	*horse*	1496	0.19	0.15	0.16	0.41	0.29	0.46	0.22	**0.59**
bird	*snow*	1352	0.64	0.44	0.52	0.77	0.75	0.77	**0.94**	0.83
car	*showroom*	1301	0.55	0.70	0.85	0.61	0.72	**0.92**	0.70	0.86
cat	*snow*	788	0.20	0.18	0.07	0.48	0.57	0.46	0.58	**0.65**
girl	*horse*	692	0.45	0.48	0.44	0.69	0.68	0.71	0.66	**0.77**
beach	*horse*	622	0.48	0.57	0.53	0.69	0.71	0.65	**0.81**	0.80
MEAN		3924	0.34	0.34	0.36	0.54	0.56	0.56	0.57	**0.66**

training examples and the various mechanisms for bi-concept search.

Experiment 1. Harvesting Bi-Concept Positive Examples. For each concept, we take all images labeled with the concept in our 1.2M set as candidate positives. We sort the candidate set by each of the three methods described in Section 3, namely Semantics, Visual, and Multi-modal Borda. In addition to the Borda method, we also consider multi-kernel learning plus SVM [111], which directly combines multi-modal similarities. For each bi-concept, we train a multi-kernel SVM on images labeled with the bi-concept, and then use the SVM to predict the positive training examples. For a more comprehensive comparison, we also report the performance of image ranking using three simple metadata features: DateUploaded, TagNum, and Views. Given an image, TagNum is the number of tags contributed by its user, while Views indicates how many times the image has been viewed.

As there is no ground-truth available for the 1.2M set, we manually check for genuine positives in the top ranked images. To reduce the manual annotation effort and (possible) labeling bias towards certain methods, we employ a pooling mechanism similar to the TRECVid benchmark [105]. For each method, we put its top 100 ranked images into a common pool without indicating their origin. For a given query of a single or bi-concept, we label an image as positive if the (bi-)concept is (partially) visible in the image. Artificial correspondences of the (bi-)concepts such as drawing, toys, and statues are labeled as negative. Notice that as the chance of including genuine positives in the negative sets is very small, we do not assess the accuracy of the negatives.

Experiment 2. Bi-Concept Search in Unlabeled Images. To configure a

bi-concept search engine, we have to specify the following three choices:
1) detector: building a bi-concept detector versus combining the confidence scores of two single-concept detectors,
2) positive: random sampling versus the multi-modal Borda fusion of semantic and visual selection,
3) negative: random sampling versus adaptive sampling.

In order to study the impact of the individual choices on bi-concept search, we design three setups for a head-to-head comparison, namely *social*, *borda*, and *full*, as listed in Table 5.3. The optimal choice of the amount of positive examples may vary over bi-concepts. For bi-concepts whose positive data can be collected at a higher accuracy, it is sensible to preserve more top ranked examples for training. Note that this study does not aim for the best possible performance, but rather focuses on revealing the advantages of bi-concepts as a retrieval method, in the context of the existing works using single-concept detectors. Hence, for each setup, we simply set the number of positive examples per bi-concept to 100. For harvesting informative negative examples, we set the number of iterations T to 10. Consequently, we also create 10 sets of randomly sampled negatives for the reason of fair comparisons. By comparing *borda* and *social*, we study the impact of positive training data. By comparing *full* and *borda*, we assess the effectiveness of informative negatives.

For combining single-concept detectors, we investigate the most common methods from the retrieval literature: the product rule and linear fusion. While the product rule helps to make a combined detector more discriminative, averaging the individual detectors helps to improve the robustness of the combined detector. Linear fusion is often used to establish a performance upper bound [43, 107]. We follow this idea, and establish a performance upper bound of linear fusion for bi-concept search by grid search with a step size of 0.05 on λ. We use average precision, a common choice for evaluating visual search engines [105].

5.4.3 Implementation

Parameters for Training (Bi-)Concept Detectors. We create a codebook with a size of 1,024 by K-means clustering on a held-out set of random Flickr images. So each image is represented by a vector quantized Dense-SIFT histogram of 1,024 dimensions. For a fair comparison between detectors trained using different setups, we train a two-class SVM using the χ^2 kernel, setting the cost parameter to 1.

Parameters for the Semantic Method. As an instantiation of $d(w', w)$ in (5.4), we choose the Normalized Google Distance [23], which measures semantic divergence between two tags based on their (co-)occurrence frequency in a large collection of social-tagged images. As our 1.2M set might be relatively small for computing this distance, we use the full list of LSCOM concepts [85] as queries, and collect up to 10 million Flickr images with social tags. The *idf* values in (5.5) are also computed on the 10M set.

Table 5.3: Experiment 2. Configuring our bi-concept image search engine using three setups.

Setup	Positive training data	Negative training data
social	100 examples randomly sampled from $X_{\mathbf{w}_{1,2}+}$	10 negative sets, each having 100 randomly generated negatives
borda	The top 100 examples retrieved by Borda Count	The same negatives as *social*
full	The same positives as *borda*	Social negative bootstrapping with T=10, n_u=1000

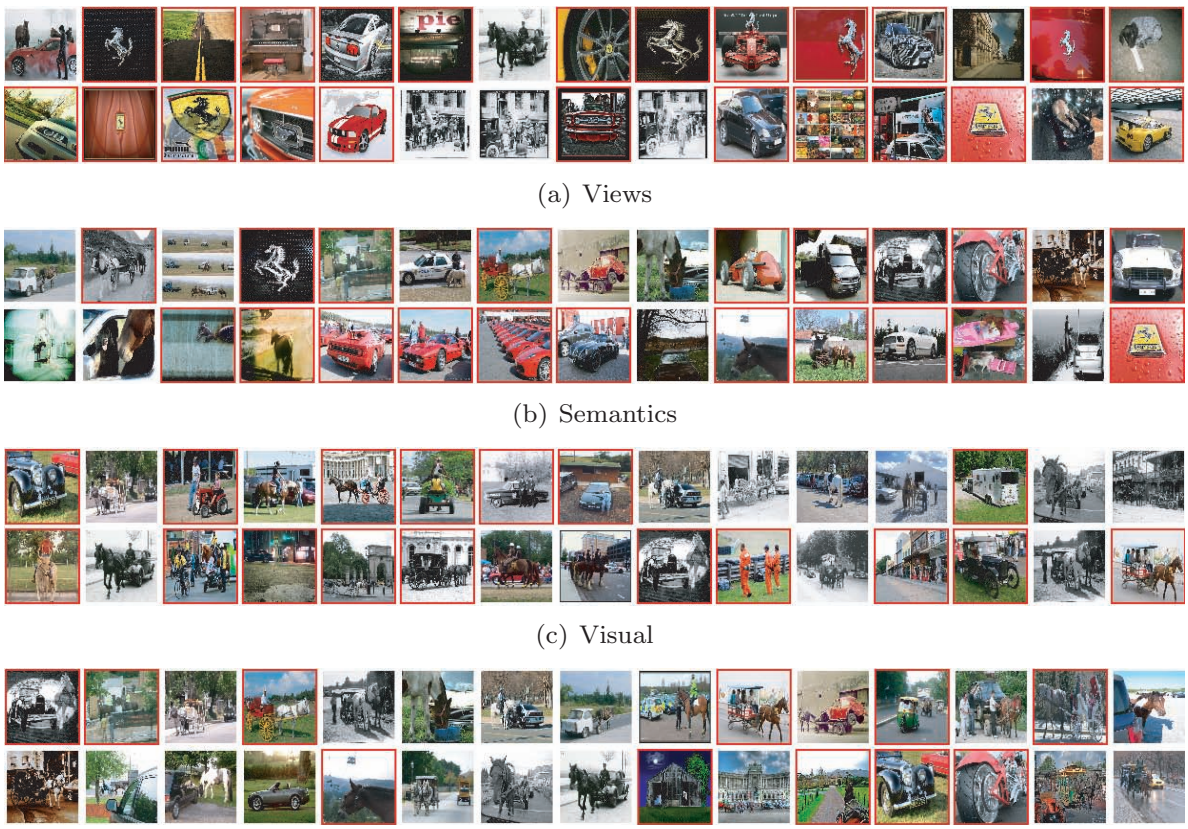
Parameters for the Visual Method. We choose the following four visual features which describes image content from different perspectives: COLOR, CSLBP, GIST, and Dense-SIFT. COLOR is a 64-d global feature, combining the 44-d color correlogram [47], the 14-d texture moments [145], and the 6-d RGB color moments. CSLBP is a 80-d center-symmetric local binary pattern histogram [44], capturing local texture distributions. GIST is a 960-d feature describing dominant spatial structures of a scene by a set of perceptual measures such as naturalness, openness, and roughness [88] (using software from [29]). Dense-SIFT [122] is the same bag-of-keypoints feature as we have used for concept detection. We compute (6.10) with the feature set F = {COLOR, CSLBP, GIST, Dense-SIFT} on the 1.2M set. We set k=1,000, a good choice for the neighbor voting algorithm [65].

Parameters for the Multi-kernel SVM. We construct multiple kernels as follows. For each of the four visual features, we use the χ^2 kernel. To measure the semantic similarity between two images, we adopt the choice of Guillaumin et al. [42], and define a tag kernel which return the number of tags shared between two images. To train a multi-kernel SVM, we take the top 100 examples ranked by TagNum as positive training data and 100 examples sampled at random as negative training data. We use the Shogun software [111], with the l2 normalization on the combination weights

5.5 Results

5.5.1 Experiment 1. Harvesting Bi-Concept Positive Examples

Social Tagging Baselines. Comparing single concept harvesting results in Table 5.1 and bi-concepts harvesting results in Table 5.2, we observe that the social tagging accuracy of bi-concepts (P@100=0.34) is much lower than its single-concept counterpart (P@100=0.68). Recall that we already removed batch-tagged images beforehand. So a possible explanation could be that when users label images with multiple tags, they tend to add more tags irrelevant to the actual content to improve the retrievability of their images. This explanation is confirmed to some extent by the behavior of the



(a) Views

(b) Semantics

(c) Visual

(d) Multi-modal Borda

Figure 5.4: Positive training examples of 'car + horse' automatically obtained from social-tagged images by different methods. The top 30 results of each method are shown. A red border indicates a false positive example.

TagNum feature. While TagNum is slightly better than random sampling for single concepts, it clearly outperforms random sampling for bi-concepts. Simply ranking images by DateUploaded does not improve the accuracy at all, indicating that freshness has no impact on relevance. The result of Views (P@100=0.71 for single concepts and P@100=0.36 for bi-concepts) shows popularity has a limited positive impact on relevance.

Bi-Concept Search Engine versus Social Tagging Baselines. As shown in Table 5.2, for bi-concepts, our search engine with Multi-modal Borda doubles the accuracy, with P@100=0.66, when compared to random sampling from the social web with P@100=0.34. The results show the effectiveness of the proposed search engine for harvesting positive training examples from social-tagged images. We demonstrate some harvested bi-concept training examples in Fig. 5.4 and Fig. 5.5. Compared to the *Random* run, the performance of the *Visual* run improves for all bi-concepts except for 'beach + boat'. For that bi-concept, Visual incorrectly ranks images of

(a) Views

(b) Semantics

(c) Visual

(d) Multi-modal Borda

Figure 5.5: Positive training examples of 'cat + snow' automatically obtained from social-tagged images by different methods. The top 30 results of each method are shown. A red border indicates a false positive example.

"boats in sea" at the top due to visual ambiguity between sea and beach. Although the multi-kernel method performs well for some bi-concepts such as 'bird + snow', it is not as effective as the Borda method in general.

Multi-modal versus Uni-modal. For single concepts, Visual reaches the best performance, on average, having 98 genuine positive examples in its top 100 retrieved results. We attribute the success of Visual to two reasons. First, while visual appearance of a single concept, e.g, 'bird', may vary significantly, the typical visual context where the concept is observed is relatively consistent, e.g., 'water' and 'sky' for 'bird'. The visual method, empowered by diverse visual features, thus estimates single concept relevance accurately. Second, rather than re-ranking a small number of image search results [1,100], we analyze all images labeled with a given single concept. The large candidate pool allows the search engine to find the images for which it is most confident. The results for bi-concepts are different. Neither of the two uni-modal methods is a clear winner. For bi-concepts with 'flower' as a component concept,

Semantics tend to outperform Visual. Recall that the features used in our experiments are global, meaning they are better at describing visual context than capturing insignificant objects within an image. As a flower is often small in an example of 'car + flower', retrieving a number of flower images in the neighbor set becomes difficult. Region-level analysis could be helpful in this case. As Semantics and Visual are complementary, combining them with the simple Borda method results in a relative improvement of 18%.

In sum, our main findings after experiment 1 are as follows. Since the social tagging accuracy of bi-concepts is much lower than that of single-concepts, harvesting positive bi-concept examples is more difficult than harvesting positive single-concept examples. While visual analysis seems adequate for single-concepts, multi-modal analysis is crucial for bi-concepts. When compared to selecting bi-concept labeled images from the social web in a random fashion, our proposed bi-concept search engine harvests bi-concept examples with doubled accuracy.

5.5.2 Experiment 2. Bi-Concept Search in Unlabeled Images

Comparing Methods for Combining Single Concepts For single concept search, unsurprisingly, single concepts trained using the *full* setup, with an MAP of 0.120, is better than single concepts trained using the *social* setup, with an MAP of 0.080. As shown in Fig. 5.6, compared to single concepts trained on random samples, single concepts learned from informative negatives are more discriminative, favoring precision over recall. As shown in Table 5.4, multiplication works slightly better than averaging for combing single concepts. This result implies that searching for bi-concepts demands detectors with better discrimination ability.

Bi-Concept Search Engine versus Combining Single Concepts. As shown in Table 5.4, our bi-concept search engine, using the *full* setup, performs best, with an MAP of 0.106. For single concept detectors trained using the *full* setup, the upper bound on the performance of linear fusion with an oracle is 0.053 , which is unlikely to be approached in practice. Even with this upper bound, we still outperform linear fusion for most bi-concepts, and overall with a relative improvement of 100%.

The Impact of Positive Training Data. Concerning positive training data for learning bi-concept detectors, the *borda* setup improves the MAP of the search engine from 0.042 to 0.080 when compared to *social*. The bi-concept comparison shows that for most bi-concepts *borda* is better than *social*. Because *borda* and *social* use the same negative training data, the result allows us to conclude that positive examples harvested by our system are better than the original social-tagged positives.

The Impact of Negative Training Data. Comparing the *full* setup with the *borda* setup, we observe from Table 5.4 that for most bi-concepts *full* surpasses *borda*, with a relative improvement of 32.5%, in terms of MAP. Since the two setups use the same positive training data, the results show the importance of informative negatives for accurate bi-concept search. To see what negative classes are recognized as informative for a given bi-concept, we show in Fig. 5.7 the most informative neg-

Table 5.4: Experiment 2. Comparing methods for bi-concept search in terms of average precision. We compare our proposed bi-concept search engine with approaches combining single concepts using product rule, linear fusion with $\lambda = 0.5$ and linear fusion with an oracle estimator for λ. For *borda* and *full*, their positive training data are obtained by the borda method reported in Table 5.2.

Bi-Concepts		$p(w_1\|x) * p(w_2\|x)$			$0.5p(w_1\|x) + 0.5p(w_2\|x)$			$\lambda p(w_1\|x) + (1-\lambda)p(w_2\|x)$			**Proposed Search Engine**		
w_1	w_2	*social*	*borda*	*full*	*social*	*borda*	*full*	*social*	*borda*	*full*	*social*	*borda*	*full*
car	*street*	0.033	0.039	0.049	0.033	0.041	**0.051**	0.015	0.019	0.036	0.041	0.040	0.050
car	*snow*	0.014	0.017	0.018	0.014	0.014	0.019	0.034	0.044	0.053	0.026	0.056	**0.109**
beach	*car*	0.040	0.035	0.035	0.040	0.032	0.033	0.042	0.033	0.037	0.037	0.040	**0.068**
car	*flower*	0.006	0.010	0.011	0.007	0.009	0.009	0.010	0.012	**0.013**	0.009	0.010	0.011
beach	*girl*	0.058	0.040	0.027	0.054	0.012	0.019	0.054	0.026	0.026	0.039	0.139	**0.180**
beach	*bird*	0.129	0.125	0.123	0.131	0.122	0.125	0.143	0.129	0.126	0.085	**0.195**	0.188
beach	*boat*	0.064	0.055	0.059	0.063	0.054	0.056	**0.082**	0.056	0.056	0.039	0.067	0.075
cat	*flower*	0.015	0.018	0.020	0.014	0.018	0.021	0.021	0.023	**0.032**	0.006	0.017	0.025
bird	*flower*	0.019	0.015	0.016	0.020	0.015	0.015	0.024	0.020	0.019	0.011	**0.039**	0.022
car	*horse*	0.032	0.029	0.032	0.032	0.017	0.021	0.048	0.037	0.026	0.022	0.038	**0.061**
bird	*snow*	0.012	0.012	0.010	0.012	0.013	0.011	0.013	0.018	0.016	0.047	0.067	**0.079**
car	*showroom*	0.108	0.103	0.118	0.108	0.106	0.130	0.108	0.157	0.197	0.142	0.185	**0.271**
cat	*snow*	0.014	0.015	0.014	0.014	0.017	0.020	0.014	0.020	0.057	0.017	0.091	**0.096**
girl	*horse*	0.021	0.026	0.038	0.020	0.028	0.034	0.024	0.035	0.038	0.028	0.030	**0.049**
beach	*horse*	0.036	0.069	0.086	0.041	0.062	0.041	0.053	0.069	0.061	0.085	0.193	**0.309**
MEAN		0.040	0.041	0.044	0.040	0.037	0.040	0.046	0.047	0.053	0.042	0.080	**0.106**

ative examples harvested from the 1.2M set by the proposed search engine. Note that negative examples which are visually close to bi-concept examples are automatically identified as informative for optimizing bi-concept detectors. Consider 'car + showroom' for instance. As shown in Fig. 5.7(a), indoor scenes such as offices and restaurants and outdoor scenes such as streets are selected by our system. Images such as close-ups of electronic device as one often see in a showrooms are also found by our system. These negative examples are helpful for the search engine to distinguish genuine examples of 'car + showroom' from examples where only one of the two single concepts are present, resulting in an absolute improvement of 0.086. Further, by visualizing social tag frequency in the informative negative set with a tag cloud, we see which negative classes are most informative with respect to a specific bi-concept. Both the quantitative and qualitative results demonstrate the viability of the proposed bi-concept image search engine.

Concerning generalization of the proposed method for more complex queries such as "finding an image showing a girl and a horse on a beach", a straightforward extension is to harvest examples for the tri-concept 'beach + girl + horse'. Though images with the three tags are relatively sparse, positive examples of the tri-concept may have a more characteristic visual appearance. Hence less training data is required. Using the same protocol as used for the bi-concepts, we have conducted an additional experiment for searching for 'beach + girl + horse'. Compared to linear fusion of the

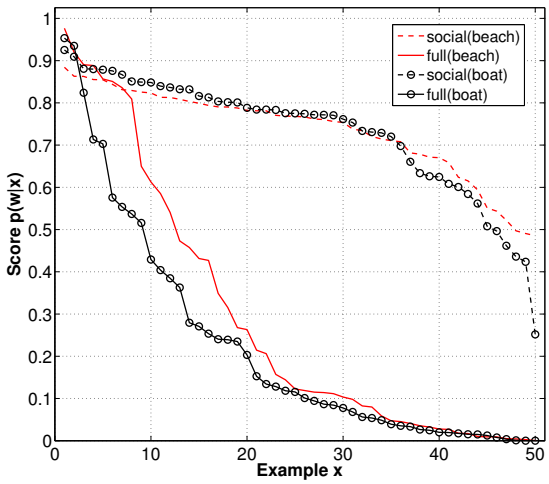


Figure 5.6: Comparing predicted scores of single concept detectors trained using different setups. social(beach): a 'beach' detector trained using the *social* setup. full(beach): a 'beach' detector trained using the *full* setup. social(boat) and full(boat) are defined in a similar fashion. Each curve is obtained by running a detector on 50 positive examples of 'beach + boat', and sorting by predicted scores in descending order. Steeper slopes indicate that detectors trained using the *full* setup are more discriminative, favoring precision over recall.

three single concepts with an AP of 0.058, the proposed search engine obtains a better performance with an AP of 0.290. Fig. 5.8 shows the top 20 search results by different methods. The results demonstrate the potential of our method for tri-concept search.

5.6 Discussion and Conclusions

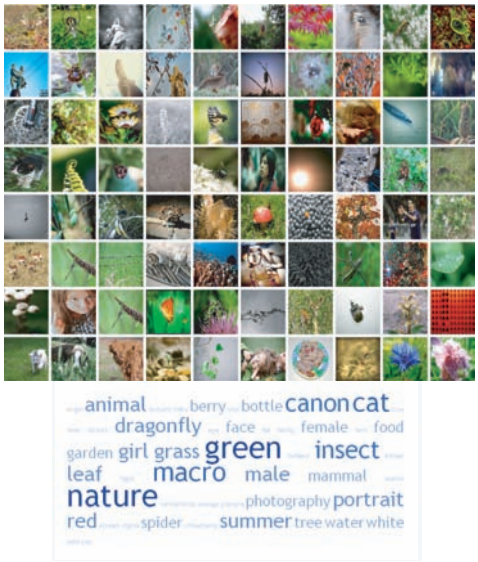
This chapter establishes *bi-concepts* as a new method for searching for the co-occurrence of two visual concepts in unlabeled images. To materialize, we propose a bi-concept image search engine. This engine is equipped with bi-concept detectors directly, rather than artificial combinations of individual single-concept detectors. Since the cost of manually labeling bi-concept training examples is prohibitive, harvesting social images is one – if not the – main enabler to learn bi-concept semantics.

The core of our search engine is a multimedia data-driven framework which collects from the social web 1) de-noised positive training examples by multi-modal analysis and 2) informative negative training examples by adaptive sampling. We study the behavior of the search engine using 1.2M social-tagged images as a data source.

Obtaining positive training examples for bi-concepts is more difficult than for single concepts, as the social tagging accuracy of bi-concepts is much lower. For single concepts, uni-modal (visual) analysis is often sufficient for de-noising. For bi-concepts, multi-modal analysis is crucial, gaining a relative improvement of 18%

(a) Informative negative training examples of 'car + showroom'

(b) Informative negative training examples of 'beach + girl'



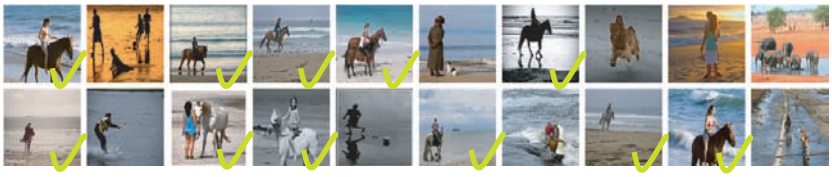
(c) Informative negative training examples of 'bird + flower'

(d) Informative negative training examples of 'car + horse'

Figure 5.7: The 80 most informative negative examples for specific bi-concepts, harvested from social-tagged images by the proposed bi-concept image search engine. By visualizing tag frequency in the selected negatives as a tag cloud, we see which negative classes are most informative to a given bi-concept.

over uni-modal. When compared to the social tagging baseline, we obtain positive examples of bi-concepts with doubled accuracy.

(a) Searching for 'beach + girl + horse' by linear fusion of beach, girl, and horse detectors



(b) Searching for 'beach + girl + horse' by the proposed search engine

Figure 5.8: Finding unlabeled images with three visual concepts co-occurring. Compared to linear fusion of single concepts, the proposed search engine obtains better search results for tri-concept 'beach + girl + horse'.

The training examples, obtained without the need of any manual annotation other than social tags, are used to train bi-concept detectors. These detectors are applied to 10K unlabeled images. Using the de-noised positive data allows us to lift the performance of the social baseline from 0.042 to 0.080, in terms of MAP. Substituting informative negative examples for randomly sampled negatives further improves the performance, reaching an MAP of 0.106. Our system even compares favorably to the oracle linear fusion of single concept detectors, with an upper bound MAP of 0.053. The results allow us to conclude that compared to existing methods which combine single concept detectors, the proposed method is more effective for bi-concept search in unlabeled data.

One concern of the chapter might be that the number of bi-concepts in our current evaluation is relatively small, when compared to single concept benchmarks [22, 48, 105]. Though our framework needs no manual verification for exploiting bi-concept examples, we actually require manually verified ground truth for a head-to-head comparison. Therefore, a novel benchmark dedicated to bi-concepts or even higher-order semantics is urged for.

Our study is orthogonal to work which aims to detail a single concept by describing it's visual attributes [34, 143], e.g., automatically adding the tag 'red' to 'car' to generate a more specific single concept 'red car'. These methods might be incorporated into our bi-concept search engine to answer bi-concept queries with two specified single concepts such as 'red car + white horse'. This would lead to a search engine capable of answering very precise queries.

Our proposed methodology is a first step in deriving semantics from images which goes beyond relatively simple single-concept detectors. We believe that for specific

pre-defined bi-concepts, they already have great potential for use in advanced search engines. Moving to on-the-fly trained queries based on bi-concepts opens up promising avenues for future research.

Chapter 6

Personalizing Automated Image Annotation using Cross-Entropy

How to personalize automated image annotation with respect to a user's preference? In this chapter, we answer the question by jointly exploiting personalized tag statistics and content-based image annotation. Using cross-entropy-minimization based Monte Carlo sampling, the proposed algorithm optimizes the personalization process in terms of a performance measurement which can be flexibly chosen. This innovation leads to automated image taggers suited for a specific user in a given situation*.

*Published in *the Proceedings of the ACM International Conference on Multimedia* 2011 [60].

6.1 Introduction

Annotating large personal collections of pictures on smart phones, personal computers, and the web is of great social importance. With the size of such collections growing so rapidly, full manual annotation is unfeasible. Thus, automatic image annotation is crucial, but this is challenging due to the well-known semantic gap: "the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for *a user in a given situation*" [106]. Much work has been conducted to (partially) bridge the gap by learning a mapping between visual features and objective semantics [41,57,97,103,134,135]. However, in the above efforts the *user* factor in the semantic gap is completely ignored. Clearly, users have personal preferences for image subjects. For instance, some users collect pictures of flowers, while others may favor images of cars. This real-world phenomenon suggests that an off-the-shelf image annotation system is unlikely to be universally applicable to the large variations in personal albums. The absence of personal information in devising image annotation models results in unsatisfactory annotations.

Some research has been conducted towards personalizing automated image annotation [30, 37, 77, 98]. Sawant et al. were among the first to leverage user tagging preferences, with their novel observation that "a user's previously used tags can be the best determinants of her future uploads" [98]. Interestingly, their study revealed that simply annotating a user's new images with tags most frequently used by the same user in the past yields a much higher accuracy than several automated methods. This result leads the authors to conclude that prediction by personal tag statistics is a reasonable upper bound on personalized image annotation performance.

In this chapter we study the problem of *personalizing automated* image annotation in a social web context. We tackle the problem by proposing a generic framework which jointly exploits generic content-based image annotation and personal multimedia tagging history. We present a learning algorithm which optimizes the personalization process in terms of a performance measurement which can be arbitrarily chosen. Due to this technical advantage, we go beyond the performance upper bound of personalized image annotation established in [98].

6.2 Related Work

Instead of proposing a new generic image annotation model, this chapter studies the personalization of automated image annotation. We first review recent progress in generic image annotation, and then we discuss related work on image annotation personalization.

6.2.1 Generic Image Annotation

A considerable amount of papers have been published for generic image annotation [41, 57, 76, 79, 103, 104, 115, 134]. We divide existing work into content-based methods and content-context-based methods.

The content-based methods predict tags purely based on image content analysis [41, 57, 76, 103, 115, 134]. Li and Wang [57] train a multivariate Gaussian mixture model for each tag, while Support Vector Machines are used in [103]. Liu et al. [76] annotate images by maximizing the joint probability of images and tags. The authors in [134] perform image annotation by learning a mapping into a common feature space where both images and tags are represented. They rank tags in terms of their distance to a test image in the common space. In contrast to per-tag modeling, k-nearest-neighbors based methods make predictions by propagating tags to the unlabeled image from its visual neighbors [80]. Weighted nearest neighbors are considered in [41]. Sparse reconstructions are employed in [115] to reduce the chance of incorrectly including neighbors which are semantically irrelevant to the unlabeled image. To enhance content-based image annotation, contextual information on the creation of an unlabeled image has been investigated [79, 104]. In [79], GPS data, indicating where the image was captured, is employed, whereas in [104], camera metadata such as shutter speed and focal length describing how the image was captured is studied. In both content-based and content-context-based methods, all users are treated equally, without taking personal preferences into account.

Work such as [26, 74, 149] studies learning image annotation models from social-tagged images. Datta et al. [26] treat user tags as positive feedback to incrementally update an existing model. Liu et al. [74] and Zhu et al. [149] measure both image-wise visual similarity and tag-wise semantic similarity to refine existing annotations. These methods learn from the social community but do not consider personalized image annotation.

6.2.2 Personalized Image Annotation

Recently some papers have appeared towards automated approaches to personalized image annotation [30, 37, 77, 98]. According to whether user interaction is needed, we divide existing work into two types of methods, automatic methods and semi-automatic methods.

The automatic methods achieve personalization by inferring from personal digital calendars [37] or multimedia tagging history [98], or training a generic model on personal collections [30]. In [37], Gallagher et al. explore the possibility of using personal calendar event annotations to label images. The rationale for the idea is based on the coincidence between the calendar event and image capture time. Calendar annotations are not always available. More importantly, tagging a calendar event is different from tagging an image. Therefore, personalizing image annotation based on the calendar tagging history seems questionable. Assuming that context-constrained images

such as those captured at the same location have a similar visual style, Duan et al. [30] propose a probabilistic model where styles are viewed as latent variables. While learning from visual features of high dimensionality requires many labeled examples, the number of personal images for a specific user is relatively small and many of them are unlabeled. Hence, learning models using personal collections alone seems problematic. To overcome the problem, Liu et al. [77] propose a semi-automatic method, by first learning a generic model for each tag using images from a professional photo forum. They then solicit user feedback to adapt the learned model to personal collections. The difficulty in obtaining user feedback for thousands of tags puts the scalability of the semi-automatic method into question. Moreover, in [30, 37, 77], personal tagging history, a strong clue for building personalized annotation models, is untouched.

In [98], Sawant et al. propose to combine personal tagging history, in the form of tag frequency, and predictions made by a content-based image annotation system [57] in a Naïve Bayes formulation. They conclude that combining tagging history and content analysis is inferior to using the history alone. We argue that their conclusion is true but for their Naïve Bayes model only. In that model the performance of the individual pieces of evidence is not considered. In contrast, we propose a personalization model which is directly optimized in terms of the prior annotation performance. As a consequence, we reach the novel conclusion that combining the personal tagging history and content analysis yields better personalized image annotation.

The rest of the chapter is organized as follows. We formulate the personalization problem and elaborate the proposed personalization model in Section 6.3. We setup experiments in Section 6.4, with results analyzed in Section 6.5. We conclude the chapter in Section 6.6.

6.3 Personalized Image Annotation

6.3.1 Problem Formalization

Let u be a user for whom we want to provide personalized image annotation. Let $X_{u,past}$ be a set of images the user has already labeled, and $X_{u,future}$ a set of unlabeled images the user wants to have tags for. We use w to denote a tag and $V = \{w_1, \ldots, w_m\}$ for a large vocabulary. For each image $x \in X_{u,future}$, we aim to annotate it with tags from the vocabulary such that the annotations are relevant with respect to the image from the user's standpoint. To do so, we use both information from the user as well as the social web community. So let X_{comm} indicate images in the community, and $X_{u,past}$ is a subset of X_{comm}. We define $G_u(x, w)$ as a *personalized* image annotation function whose output is a confidence score of the tag w being relevant to the image x. This allows us to rank tags by $G_u(x, w)$ in descending order and preserve the top ranked tags as annotations of unlabeled images for this particular user.

Personalized image annotation is a complex task. It is unlikely that an image

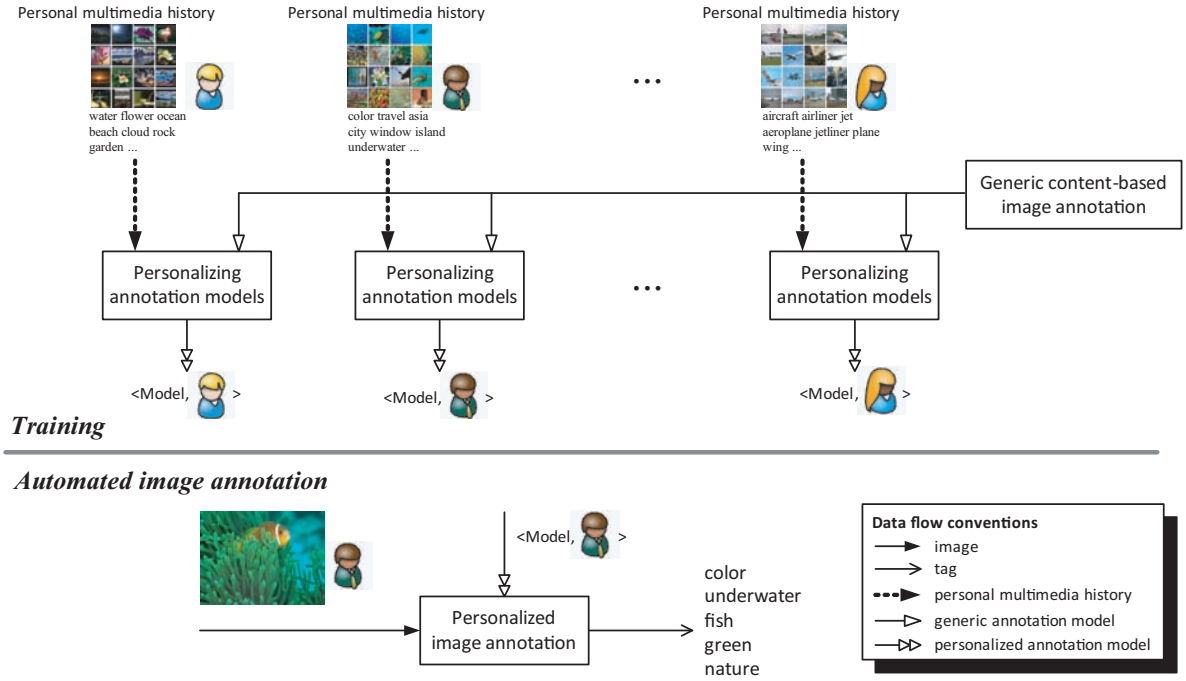


Figure 6.1: The proposed framework for *personalizing* automated image annotation. For a given user, we deliver a personalized image annotation model, by jointly exploiting content-based image annotation and the user's multimedia tagging history.

annotation function based on a single modality can capture all relevant characteristics. We therefore need to look at the problem from multiple perspectives. Ideally, we exploit varying evidence such as content-based image annotation driven by diverse features [41,65], tag statistics in personal collections and networks [98], personal daily activities [37], geographic context [79], or camera metadata [104]. In this study, we focus on combining content-based annotation and personal tag statistics, as they are two fundamental elements related to the problem and are often more accessible than other elements. Nevertheless, to make our discussion general, we consider combining annotation functions driven by multiple sources of evidence. To that end, let $\{g_1(x,w),\ldots,g_t(x,w)\}$, with $g_j(x,w) \in [0,1]$, $j = 1,\ldots,t$, be a set of such image annotation functions. As combining these functions can be viewed as a multi-modal fusion problem, we choose to use a linear weighted sum, an effective strategy for multi-modal fusion according to [5]. The importance of individual tags varies per user, so tag-dependent weights are necessary. To formalize the above notion, for each tag w_i, $i = 1,\ldots,m$, we express a parameterized version of $G_u(x,w)$ as

$$G_{u,\Lambda}(x,w_i) = \sum_{j=1}^{t} \lambda_{i,j} g_j(x,w_i), \tag{6.1}$$

where $\{\lambda_{i,j}\}$ are non-negative weighting parameters, and $\Lambda = [\lambda_{i,j}]_{m\times t}$ is the pa-

rameter matrix. While $\lambda_{i,j}$ indicates the importance of $g_j(x, w_i)$ for predicting w_i, their summation, namely $\sum_{j=1}^{t} \lambda_{i,j}$, reflects the importance of w_i for annotating the personal collection. By optimizing the weights per user, we obtain personalized image annotation models.

To find the optimal weights for a given user, we need information about what tags the user is likely to use for tagging her/his personal collections. We assume that a user's tagging preference is relatively consistent within a certain period. Therefore, the images the user has already labeled $X_{u,past}$ are the prime candidates. To make the above notion operational, we need to formulate an optimization goal per user. Let $rank(V|x, G_{u,\Lambda})$ be a ranking of the vocabulary V for an image $x \in X_{u,past}$, obtained by sorting tags in descending order by $G_{u,\Lambda}(x, w)$. Let $\mathbf{w}_x$ be the set of tags assigned to x by its user, serving as ground truth to assess $rank(V|x, G_{u,\Lambda})$. We define

$$E(rank(V|x, G_{u,\Lambda}), \mathbf{w}_x)$$

as a performance measure function which produces a real-valued score indicating ranking quality. As we learn from a set of images, rather than from a single image, we define a set-level performance measure as

$$S(X, \Lambda) = \frac{1}{|X|} \sum_{x \in X} E(rank(V|x, G_{u,\Lambda}), \mathbf{w}_x), \tag{6.2}$$

where $|\cdot|$ is the cardinality of a set. Putting everything together, we formulate the problem of personalizing automated image annotation for a given user as solving the following optimization problem:

$$\Lambda^* = \underset{\Lambda}{\operatorname{argmax}}\, S(X_{u,past}, \Lambda), \tag{6.3}$$

subject to

$$0 \leq \lambda_{i,j} \leq 1. \tag{6.4}$$

Solving Eq. 6.3 yields the optimal parameters for the personalized model defined in Eq. 6.1, which are then used to annotate new, yet unlabeled, images of the user. We illustrate the proposed framework in Fig. 6.1.

6.3.2 Personalization using Cross-Entropy

Finding a solution for the optimization problem in Eq. 6.3 is nontrivial. As the performance measure function E is often not differentiable, a standard gradient-ascent based algorithm is inapplicable. A common approach to such a problem is Monte Carlo simulation. But, when the parameter space is large, as in our case, finding Λ^* or its good approximation by random sampling is a rare event. A crude Monte Carlo approach would imply an unfeasibly large simulation effort. A solution for rare event search is offered by the cross-entropy method [93], which iteratively optimizes

an arbitrary function by importance sampling. We first describe the cross-entropy method in general, and then present a cross-entropy based learning algorithm for solving Eq. 6.3.

The Cross-Entropy Method

Imagine that our goal is to maximize an objective function $S(\Lambda)$, and its maximum is found at Λ^*. The cross-entropy method [93] assumes that Λ is a random variable following a parametric distribution $p(\Lambda;\Theta)$ which is specified by an (unknown) hyper parameter Θ. In a nutshell, the method consists of the following two steps executed iteratively:

Step 1. Randomly generate n samples using $p(\Lambda;\Theta)$; We use $\{\Lambda^{(1)},\ldots,\Lambda^{(n)}\}$ to denote the n samples.

Step 2. Select the top s samples $\{\widehat{\Lambda}^{(1)},\ldots,\widehat{\Lambda}^{(s)}\}$ by sorting the n samples in descending order by $S(\Lambda)$, where the selected samples are called *elite* samples. Re-estimate Θ by maximum likelihood estimation on the s elite samples.

From the second step we see that the hyper parameter Θ is updated in terms of the elite samples. As a consequence, the probability of generating good samples is progressively increased, making Λ converge towards its optimal value. The procedure repeats until it hits certain stop criteria. For instance, the performance does not improve or the number of iterations exceed a given threshold. We compute Λ^* as the expectation of $p(\Lambda;\Theta)$.

The theoretical foundation of the two-step procedure is that, the original optimization problem can be tackled by iteratively solving the following problem,

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}} \int_{\Lambda} I(S(\Lambda) \geq \gamma) p(\Lambda;\Theta) d\Lambda, \tag{6.5}$$

where I is an indicator function, and γ is a given level. The rationale for Eq. 6.5 is that a good choice of Θ shall generate more elite Λ with $I(S(\Lambda \geq r)) = 1$. We use T to indicate the number of iterations in total and $q = 1,\ldots,T$ to index a specific iteration. For a given level $\gamma^{(q)}$, let $\Theta^{(q)}$ be the solution to Eq. 6.5. By constructing an increasing sequence of levels $\{\gamma^{(q)}\}$, and correspondingly finding a sequence of hyper parameters $\{\Theta^{(q)}\}$ by solving Eq. 6.5, the optimal solution is progressively approached. When $\gamma^{(T)}$ is close to $S(\Lambda^*)$, the expectation of $p(\Lambda;\Theta^{(T)})$ will be close to Λ^*.

So in each iteration, our goal is to find Θ such that the cross entropy between $p(\Lambda;\Theta^{(q)})$ and $p(\Lambda;\Theta)$ is minimized. According to [93], minimizing the cross entropy turns out to be maximum likelihood estimation on those elite samples with $S(\Lambda) \geq \gamma^{(q)}$. To see how this conclusion results in the second step of the cross-entropy method, let $\{\widehat{\Lambda}^{(1,q)},\ldots,\widehat{\Lambda}^{(s,q)}\}$ be the s elite samples found in the q-th iteration. By setting $\gamma^{(q)} = S(\widehat{\Lambda}^{(s,q)})$, $\Theta^{(q)}$ is the result of maximum likelihood estimation on $\{\widehat{\Lambda}^{(1,q)},\ldots,\widehat{\Lambda}^{(s,q)}\}$.

The Cross-Entropy based Learning Algorithm

We now present an algorithm for image annotation personalization, on the basis of the cross-entropy method. For reasons of simplicity, we assume that the weighting parameters $\{\lambda_{i,j}\}$ are independent of each other. Each parameter $\lambda_{i,j}$ follows a distribution $p(\lambda_{i,j};\theta_{i,j})$, and $\Theta = [\theta_{i,j}]_{m\times t}$ is the hyper parameter matrix. Moreover, we choose binomial distributions to be the distribution family, because the parameter of a binomial distribution, namely $\theta_{i,j}$, directly measures the impact of $\lambda_{i,j}$ on the personalization process. If a larger (smaller) value of $\lambda_{i,j}$ contributes more to the objective function $S(X_{u,past},\Lambda)$ in the current learning round, $\theta_{i,j}$ increases (decreases) such that a larger (smaller) value is more likely to be assigned to $\lambda_{i,j}$ in next rounds. Concretely, in the q-th iteration, we first randomly generate a sequence of n samples, $\{\Lambda^{(1,q)},\ldots,\Lambda^{(n,q)}\}$, where

$$\lambda_{i,j}^{(l,q)} \leftarrow \frac{1}{N} Binomial(N,\theta_{i,j}^{(q-1)}),\ \text{for } l=1,\ldots,n. \tag{6.6}$$

Note that to satisfy the constraints that $0 \le \lambda_{i,j} \le 1$, we divide the output of the Binomial function by the number of trials. Subsequently, we find s elite samples from the n samples by sorting them in descending order according to our objective function $S(X_{u,past},\Lambda)$. As we have mentioned in Section 6.3.2, the optimal $\Theta^{(q)}$ is found by maximum likelihood estimation on the s elite samples. For a Binomial distribution, this amounts to averaging over the elite samples, namely

$$\theta_{i,j}^{(q)} = \frac{1}{s}\sum_{l=1}^{s} \widehat{\lambda}_{i,j}^{(l,q)}. \tag{6.7}$$

Table 6.1: The proposed cross-entropy based learning algorithm for optimizing automated image annotation per user.

INPUT: A user's multimedia tagging history $X_{u,past}$, base image annotation functions $\{g_1,\ldots,g_t\}$,

OUTPUT: optimized weights Λ^* for the personalized image annotation function $G_{u,\Lambda}(x,w)$.

1. Initialize $\Theta^{(0)}$
2. for $q=1,\ldots,T$
3. Randomly generate $\{\Lambda^{(1,q)},\ldots,\Lambda^{(n,q)}\}$ using Eq. 6.6
4. Evaluate the generated samples using Eq. 6.2, and select the s elite samples
5. Obtain $\Theta^{(q)}$ by maximum likelihood estimation on the s samples using Eq. 6.7
6. $\Lambda^* \leftarrow \Theta^{(T)}$

Since the expectation of $\frac{1}{N}Binomial(N, \theta)$ is θ, the optimal set of weights Λ^* found by the proposed algorithm is $\Theta^{(T)}$.

We summarize our algorithm in Table 6.1. As there is no need to compute gradients for the objective function, the proposed algorithm can optimize the personalization process in terms of a performance measure which can be arbitrarily chosen. Moreover, its convergence is theoretically guaranteed by the underlying cross-entropy method [93].

Concerning the complexity of our algorithm, the main computational effort is spent on evaluating $S(X_{u,past}, \Lambda)$. We assume that the generic annotation functions $\{g_j(x, w)\}$ are precomputed. For a given Λ, the complexity of constructing a tag rank for an image is $O(m \cdot t + m^2)$, and consequently $O(|X_{u,past}| \cdot (m \cdot t + m^2))$ for the entire training set. Notice that the evaluations of the n parameters $\{\Lambda^{(l,q)}\}$ are independent of each other. The computation associated with each image is also independent of other training images. Therefore, the algorithm can be easily parallelized.

6.4 Experimental Setup

To verify our proposal of personalized image annotation, we conduct a series of experiments on realistic personal image sets collected from the social web.

6.4.1 Data Sets

Community Image Set X_{comm} for building a content-based image annotation system. We use a set of 3.5 million images randomly sampled from Flickr by our earlier work [64]. Because batch-tagged images are often (nearly) duplicate and of low tagging accuracy, such images are not helpful for content-based image annotation. Also, we want tags to be meaningful. With these two considerations, we remove batch-tagged images and tags not defined in WordNet [32]. We use the remaining 800K images as X_{comm}. Since tags with very low frequency are unlikely to be well predicted, we preserve tags assigned to at least 100 images, and thus obtain a vocabulary V with $m = 5,073$ tags.

Personal Image Sets X_u for testing personalized image annotation. As this work studies how to personalize automated image annotation, the personal image sets for evaluation should be independent of the 800K community image set. To that end, we choose NUS-WIDE [22], which consists of 20K Flickr images after the same preprocess as we used for the community set. We aim to learn from a user's multimedia tagging history. Therefore, for each user, instead of splitting her/his image set at random, we divide the set into two distinct subsets, namely *Past* and *Future*, such that images from *Past* were uploaded before images from *Future*. The *Past* and *Future* sets are instantiations of $X_{u,past}$ and $X_{u,future}$ defined in Section 3.1. To reveal how much personal tagging history is required for the history information to be useful, we conduct a study on 5,315 users with varying amounts of personal

Table 6.2: We build personalized image annotation models for 5,315 users with varying amounts of personal tagging history. The amount of tagging history per user is measured by $|X_{u,past}|$.

$\lvert X_{u,past}\rvert$	Number of users
1	1,422
$2 \sim 9$	2,554
$10 \sim 49$	1,221
≥ 50	118

tagging history, as shown in Table 6.2. The number of images in the *Past* sets ranges from 1 to 205, with an average value of 7.9. The *Future* set has similar statistics. For each user, we use $X_{u,past}$ for training and $X_{u,future}$ for evaluation. Note that we treat each test image as unlabeled. Its user tags are merely used for ground-truth purposes.

6.4.2 Base Image Annotation Functions

We choose two state-of-the-art models, *PersonalPreference* [98] and *Visual* [65], which predict tags using tag statistics and visual content, respectively.

PersonalPreference. We choose this function for its good performance for personalized image annotation, as suggested in [98]. Given an unlabeled image x from a user u, the PersonalPreference model simply annotates x with the most frequent tags in $X_{u,past}$. Let $P(w|X)$ be the tag distribution in a social-tagged image set X, computed as

$$P(w|X) \approx \frac{\text{freq}(w|X) + \epsilon}{\sum_{j=1}^{m} \text{freq}(w_j|x) + \epsilon \cdot m}, \tag{6.8}$$

where $\text{freq}(w|X)$ is the number of images labeled with w in X, and ϵ is a small positive constant for smoothing. We express the PersonalPreference version of $g(x, w)$ as

$$g_{pp}(x, w) = P(w|X_{u,past}). \tag{6.9}$$

Visual. This model as introduced in our previous work [65] predicts tags purely based on image content. Our experiments show that it outperforms ALIPR [57], the content-based model used in [98]. Given an image x represented by a visual feature f, the Visual model first finds k neighbor images visually close to x from the community image set X_{comm}, and then selects the most frequent tags in the neighbor set as annotations of x. To overcome the limitation of single features in describing image content, predictions made based on individual features are uniformly combined. We express the Visual version of $g(x, w)$ as

$$g_v(x, w) = \frac{1}{|F|}\left(\sum_{f \in F} \frac{\text{freq}(w|X_{x,f,k})}{k} - \frac{\text{freq}(w|X_{comm})}{|X_{comm}|}\right), \tag{6.10}$$

where F is a set of features, and $X_{x,f,k}$ are the k visual neighbors of x with the visual similarity defined by f.

To implement Eq. 6.10, we choose three decent visual features as follows: COLOR, CSLBP, and GIST. The COLOR feature is a 64-d global feature, combining the 44-d color correlogram [47], the 14-d texture moments [145], and the 6-d RGB color comments. The CSLBP feature is a 80-d center-symmetric local binary pattern histogram [44], capturing local texture distributions. The GIST feature is a 960-d feature describing dominant spatial structures of a scene by a set of perceptual measures such as naturalness, openness, and roughness [88]. The parameter k is set to 500.

By incorporating the two complementary base functions, $g_{pp}(x, w)$ and $g_v(x, w)$, into our unified framework, we aim for good personalized image annotation.

6.4.3 Implementation

Parameters of the Proposed Model. There are $m = 5,073$ tags and $t = 2$ base image annotation functions. We empirically set the parameters of the algorithm described in Table 6.1 as follows: $n = 10$, $s = 2$, and $T = 200$. The computational time of the algorithm is linearly proportional to the size of the training data. For a user with 50 tagged images for training, each learning round costs approximately 42 seconds in our prototype system.

Evaluation Criteria. We use precision at top 1 (P@1) and precision at top 5 (P@5) to evaluate the accuracy of the top predicted tags. To evaluate entire tag rankings, we use average precision (AP), a good combination of precision and recall. The personalization process is optimized in terms of AP. The performance for a given user is averaged over all test images of this user.

6.4.4 Experiments

Experiment 1: User Tagging Consistency. We aim to verify to what extent our conjecture about user tagging consistency made in Section 3.1 is valid. Due to the lack of golden criteria for judging consistency, we compare the divergence between tag distribution of the same user and from different users. Given two users u_i and u_j, we compute the Jensen-Shannon divergence between $P(w|X_{u_i,past})$ and $P(w|X_{u_j,future})$, where the probability masses are computed using Eq. 6.8. So $i = j$ indicates intra-user divergences, while $i \neq j$ indicates inter-user divergences.

Experiment 2: Comparing Models. We compare the proposed model with the following three baselines: two generic models, namely CommunityPreference and Visual, and one personalized model, PersonalPreference. CommunityPreference annotates an image by simply predicting the most frequent tags within the community set. Visual and PersonalPreference, as mentioned in Section 6.4.2, are two ingredients in the proposed model.

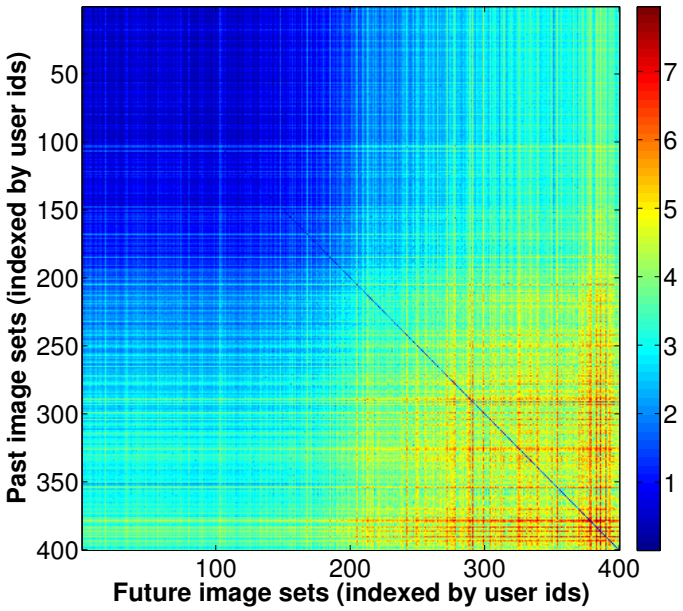


Figure 6.2: Experiment 1. User tagging consistency. The axes represent the *Past* and *Future* sets of 400 users with $|X_{u,past}|$ ranging from 1 to 168. Each entry in the matrix is the Jensen-Shannon divergence between the tag distribution in a *Past* set and in a *Future* set. The matrix is asymmetrical due to inter-user tagging divergences. The diagonal line indicates intra-user divergences. Best viewed in color.

6.5 Results

6.5.1 Experiment 1: User Tagging Consistency

The intra-user and inter-user divergence matrix is shown in Fig. 6.2, where the diagonal line denotes the intra-user divergences. For a better view of the four user groups in Table 6.2, we randomly select 100 users from each group, and arrange the matrix in ascending order in terms of $|X_{u,past}|$. For users with a very short tagging history as shown in the top left corner of Fig. 6.2, the intra-user divergences are smaller than their inter-user counterparts within the same group. But the difference is relatively small, largely due to the fact that the lack of tagging history makes the estimated tag distributions less distinguishable. As the amount of the tagging history increases, we observe a more clear difference between intra-user divergences and inter-user divergences. See for instance $|X_{u,past}| \geq 50$ as shown in the bottom right corner of Fig. 6.2. Viewing the inter-user divergences as a baseline, we conclude that the tagging preferences of the same user is relatively consistent.

Table 6.3: Experiment 2. Comparing the overall performance of generic and personalized image annotation models. The amount of personal tagging history is reflected by $|X_{u,past}|$. Scores are averaged over users. A gray cell indicates the top performer.

	$\|X_{u,past}\| = 1$			$\|X_{u,past}\| = 2 \sim 9$			$\|X_{u,past}\| = 10 \sim 49$			$\|X_{u,past}\| \geq 50$			**MEAN**		
Annotation models	*P@1*	*P@5*	*AP*	*P@1*	*P@5*	*AP*	*P@1*	*P@5*	*AP*	*P@1*	*P@5*	*AP*	*P@1*	*P@5*	*AP*
CommunityPreference	0.033	0.046	0.036	0.044	0.058	0.043	0.058	0.088	0.053	0.077	0.106	0.061	0.045	0.063	0.044
Visual [65]	0.175	0.108	0.079	0.198	0.128	0.090	0.249	0.164	0.105	0.304	0.200	0.118	0.206	0.132	0.091
PersonalPreference [98]	0.271	0.233	0.194	0.403	0.273	0.219	0.571	0.398	0.302	0.610	0.437	0.328	0.411	0.295	0.232
This chapter	**0.307**	**0.244**	**0.209**	**0.439**	**0.293**	**0.245**	**0.597**	**0.419**	**0.328**	**0.655**	**0.469**	**0.356**	**0.445**	**0.313**	**0.257**

6.5.2 Experiment 2: Comparing Models

Personalized Models versus Generic Models. As shown in Table 6.3, when a user's personal tagging preference is unknown, content-based prediction, i.e, the *Visual* model, with an AP score of 0.091, is much better than *CommunityPreference* with an AP score of 0.044. Once a user's tagging history is available, the simple *PersonalPreference* model, with an AP score of 0.232, clearly outperforms content-based prediction. The statement is valid even for $|X_{u,past}| = 1$. The result is consistent with the observation made by [98] that a user's previously used tags are important for predicting her/his future uploads.

Comparing Two Personalized Models. As shown in Table 6.3, the proposed model compares favorably to *PersonalPreference* under all evaluation criteria. In contrast to [98] where *PersonalPreference* is considered as an upper bound on image annotation performance, our model surpasses the "upper bound".

We compare the two personalized models, given users with varying amounts of personal tagging history. In the extreme case, there are 1,422 users, each having only one tagged images available for training. For 67% of the 1,422 users, we observe improvements, with a relative gain of 8% in terms of AP. Richer tagging history results in better personalized models in general.

For a comprehensive study, we make a per-user comparison between our model and *PersonalPreference*. The **absolute** improvement in terms of AP is shown in Fig. 6.3. For 4,442 out of the 5,315 users in our experiments, the proposed model is better than *PersonalPreference*. For 1,088 users, we obtain an absolute improvement of at least 0.05 in terms of AP. We provide in Table 6.4 a close-up view of the two extremes of the performance curve. In the worst case (Bottom 1), the two ground truth tags 'peninsula' and 'winchester' correspond to abstract notions with rare frequency in the community set. As a consequence, the *Visual* model fails to predict these two tags, resulting in a worse personalized model compared to *PersonalPreference*. In the best case (Top 1), our model, by ranking 'balloon' at the top, improves AP from 0.333 to 1. Overall the proposed algorithm strikes a proper balance when combining *PersonalPreference* and *Visual* in the process of model personalization.

We also look into the scenario when richer personal tagging history is available. For 94.9% of the 118 users with $X_{u,past}| \geq 50$, we observe improvements when compared

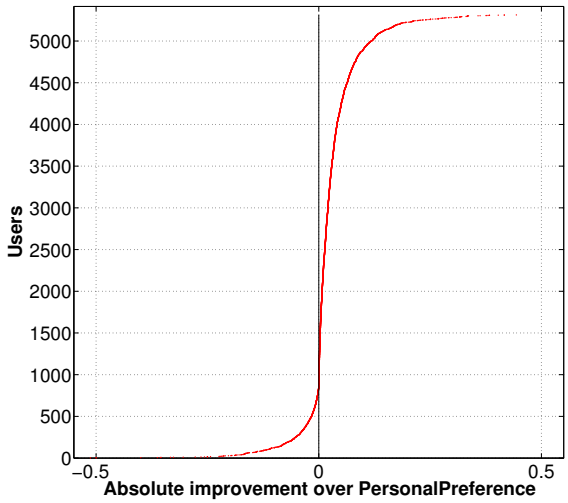


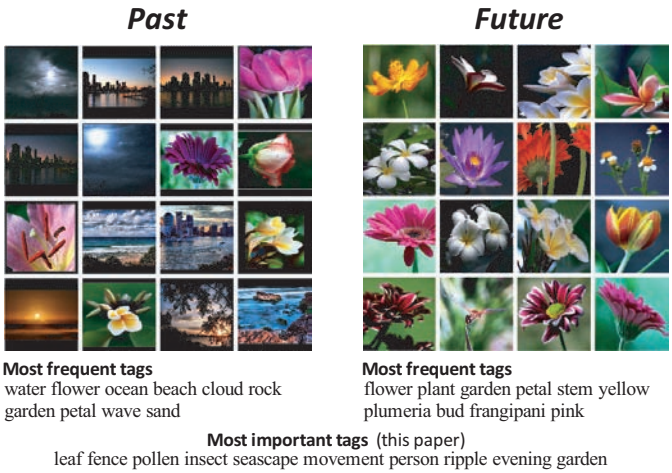
Figure 6.3: Experiment 2. Comparing two personalized models: The proposed model *versus* PersonalPreference. The performance measure is average precision. For the majority of users in consideration, we obtain personalized image annotation with a higher accuracy.

to *PersonalPreference*. While in the worse case there is a relative loss of 4%, in a successful case we reach a relative improvement of 60%. For a better understanding of (un)successful cases, we illustrate two of them in Fig. 6.4. For both cases, due to the divergence between the tag distribution in *Past* and in *Future*, PersonalPreference yields relatively lower performance, with AP scores of 0.285 and 0.195, respectively. For the successful case, however, pictures of flowers can be well annotated by the *Visual* model, with an average precision of 0.343. By cross-entropy based learning, our model reaches an AP of 0.455. Since images in the worst case are heavily edited, making image content analysis more difficult, *Visual* performs badly, with an average precision of 0.073. We conclude that unless both base image annotation functions fail, the combined model in general yields better or at least comparable performance, compared to the base functions.

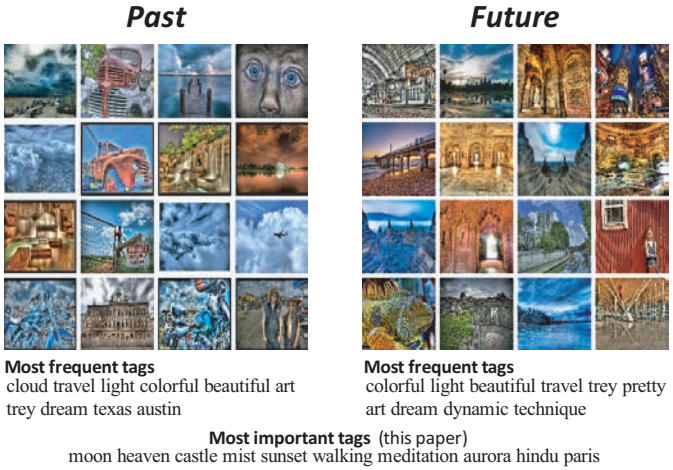
Finally, we present some qualitative results in Table 6.5. The proposed algorithm emphasizes tags which are less frequent, yet more meaningful than the most frequent tags which tend to be general. To summarize, both quantitative and qualitative results verify the effectiveness of the proposed algorithm for personalizing automated image annotation.

6.6 Discussion and Conclusions

Automated image annotation is an important yet challenging research problem. In this chapter, we study a novel aspect of the problem: *personalization* – personalizing



(a) A successful case



(b) An unsuccessful case

Figure 6.4: Illustrating successful and unsuccessful cases of image annotation personalization when rich personal tagging history ($|X_{u,past}| \geq 50$) are available for training. We personalize the *Visual* model $g_v(x, w)$ in terms of a user's *Past* and apply the personalized model $G_u(x, w)$ to annotate the user's *Future* images. In (a), $g_v(x, w)$ is stressed due to its good performance in *Past*, resulting in a relative improvement of 60% over PersonalPreference $g_{pp}(x, w)$. In (b), both $g_v(x, w)$ and $g_{pp}(x, w)$ perform poorly, resulting in a relative loss of 4% against $g_{pp}(x, w)$. Compared to $g_v(x, w)$ and $g_{pp}(x, w)$, unless both base functions fail, our algorithm generally yields better or at least comparable performance.

generic image annotation models with respect to a given user.

We confirm the observation from previous work [98] that personal tagging preference is a strong source of evidence for predicting a user's future annotation. Similar to [98], our experiments also show that a model simply using personal tagging statis-

Table 6.4: A close-up of the two extremes in Fig. 6.3. Images ranging from Top 1 to Top 6 (Bottom 1 to Bottom 6) have the largest (least) absolute improvements, when comparing our model to the *PersonalPreference* model. In the context of personalization image annotation, we consider user tags as the ground truth. The function $g_v(x, w)$ indicates the *Visual* model [65], $g_{pp}(x, w)$ for the *PersonalPreference* model [98], and $G_u(x, w)$ for the proposed model. For each model, the top ranked tags are shown. Correct annotations are marked by an *italic* font.

		Annotation results					Annotation results		
Top 1	**Truth**	$g_v(x,w)$	$g_{pp}(x,w)$	$G_u(x,w)$	**Top 2**	**Truth**	$g_v(x,w)$	$g_{pp}(x,w)$	$G_u(x,w)$
	balloon	flower	mickey	*balloon*		market	beach	vendor	*market*
		pink	school	gym			water	indian	indian
		macro	*balloon*	high			car	*market*	vendor
		cat	high	school			street		bw
		girl	gym	mickey			food		two
Top 3					**Top 4**				
	rabbit	dog	cute	*animal*		toy	flower	duke	*toy*
	sit	cat	coaster	*pet*		anniversary	food	*toy*	*anniversary*
	sitting	*pet*	summer	*portrait*			nature	*anniversary*	ho
	pose	*animal*	color	*bunny*			garden	ho	lady
	portrait	cute	sweet	young			dog	lady	duke
Top 5					**Top 6**				
	navy	boat	usa	*ship*		beach	cloud	panorama	*beach*
	ship	bus	*navy*	*navy*		coast	sky	canon	*storm*
		harbour	*ship*	usa		water	sunset	rip	*weather*
		water	alabama	museum		storm	*beach*	ocean	rock
		river	museum	alabama		weather	*water*	*weather*	rip
Bottom 1					**Bottom 2**				
	peninsula	bridge	*peninsula*	fountain		nj	animal	*nj*	manhattan
	winchester	building	*winchester*	water		squirrel	zoo	*squirrel*	ny
		house	water	square		explore	night	halloween	dog
		city	square	navy			mountain	dog	parade
		river	fountain	pier			building	ny	*nj*
Bottom 3					**Bottom 4**				
	microphone	bw	*microphone*	music		elephant	tree	*elephant*	dog
		blackandwhite	music	*microphone*			animal	dog	*elephant*
		street		smile			nature	mutt	mutt
		people		vienna			green	collar	collar
		portrait		subway			bird		plant
Bottom 5					**Bottom 6**				
	toledo	sky	*windmill*	sky		pottery	food	*pottery*	disaster
	windmill	night	holland	holland		ceramic	flower	*wheel*	big
		cloud	sky	landscape		wheel	dog	disaster	bowl
		bridge	landscape	cloud		me	macro	big	*wheel*
		blue	cloud	*windmill*			cat	*ceramic*	*pottery*

tics clearly outperforms a content-based model [65] which exploits multi-feature visual content analysis. This nontrivial phenomenon implies that the landscape of personalized image annotation is much different from generic image annotation. Let us now look back at the fundamental challenge in image annotation, namely the semantic gap [106]. The objective aspect of the gap might be ultimately surmounted by machine vision, which aims for an understanding of the visual content independent of the user. Since the human interpretation of the content depends on the specific user in a given situation [106], personalization is essential for solving the subjective aspect of the gap. Thus, to fully bridge the semantic gap, personal information such as tagging history is a factor of major importance. Nevertheless, we challenge the conclusion

of [98] that annotating images using personal tagging statistics is a performance upper bound for personalized image annotation.

To personalize generic image annotation models, we propose a linear fusion framework which jointly exploits a user's personal multimedia tagging history and content-based image annotation. The proposed cross-entropy based model enables the personalization process to be optimized in terms of an (arbitrarily) chosen performance measure. It is due to this technical innovation that we can go beyond the performance upper bound defined in [98].

We have conducted an extensive evaluation on 5,315 realistic users with varying amounts of personal tagging history. For the majority of users in consideration, the proposed personalization model surpasses the "upper bound". In an extreme scenario, where a user has only one tagged image available for training, we observe improvements for 67% of these one-training-image users, with a relative gain of 8% in terms of average precision. In general, richer personal tagging history leads to better personalized annotation models. These results clearly verify the effectiveness of the proposed framework for personalized image annotation.

Thus far, we have successfully exploited two heterogenous image annotation functions in the proposed framework. Since our framework is general, other annotation functions driven by varied evidence could be easily added in the future.

Table 6.5: Important tags *versus* frequent tags. The most important tags within a user's tagging vocabulary are found by sorting tags in descending order using $\sum_{j=1}^{t} \lambda_{i,j}$. Recall that $\lambda_{i,j}$ reflects the importance of an image annotation function $g_j(x, w)$ for predicting tag w_i. As the cross-entropy method was original invented for rare event search [93], our model recognizes tags which are less frequent yet more meaningful. For instance, 'sheep' and 'basket' for User 1, 'stage' and 'racing' for User 2, and 'house' and 'sunlight' for User 3.

	Top ranked tags	
$X_{u,past}$ **of User 1** 	*Frequency*	*Importance*
	washington	sheep
	animal	dock
	wildlife	tent
	vancouver	peanut
	outdoor	basket
	nature	snake
	pet	feather
	female	holiday
	bird	black
	rescue	rooster
$X_{u,past}$ **of User 2** 	*Frequency*	*Importance*
	image	stage
	picture	music
	photo	racing
	animal	aquarium
	zoo	bird
	nature	swim
	austria	angry
	tier	mare
	google	flying
	art	floral
$X_{u,past}$ **of User 3** 	*Frequency*	*Importance*
	cloud	land
	brasil	tunnel
	road	house
	sky	sol
	brazil	sunlight
	car	field
	blue	rural
	highway	ga
	landscape	muscle
	green	pb

Chapter 7

Summary and Conclusions

7.1 Summary

This thesis contributes to social image search, a research field emerging due to the fact that digital images have become social. For effective retrieval and repurposing of images on the social web, we have to determine whether what people spontaneously say about an image is factually in the pictorial content. Moreover, as the majority of social images are untagged, methods for deriving semantics from the content are required. Social image search is thus of scientific and social importance. We exploited socially tagged images for extracting objective semantics perceived by the community and subjective semantics related to individual users from the pictorial content.

7.1.1 Part I: Offline Learning

Chapter 2. Learning Social Tag Relevance by Neighbor Voting. In this chapter, we propose a neighbor voting algorithm which learns tag relevance by accumulating votes from visual neighbors. We proof that when 1) the probability of correct social tagging is larger than the probability of incorrect social tagging and 2) content-based visual search is better than random sampling, the proposed algorithm produces a good tag relevance estimator for both image ranking and tag ranking. As the visual neighbors are from socially tagged images, our algorithm only requires the existing tags, without the need of any extra manual annotation. Moreover, the algorithm does not build models for individual tags, so it is efficient in handling large amounts of social images with many tags. Three experiments on 3.5 million Flickr photos demonstrate the general applicability of the proposed algorithm in both social image retrieval and image tag suggestion. Thus it also provides a good basis for finding positive examples for learning visual concept classifiers.

Chapter 3. Tag Relevance Fusion for Social Image Search. To overcome the limits of a single tag relevance estimator for harvesting positive training examples from socially tagged images, we propose in this chapter tag relevance fusion as an extension of tag relevance estimation. We develop the notion of early and late fusion from generic multimedia analysis in the new context. Using the neighbor voting algorithm to instantiate base tag relevance estimators, we have conducted a systematic study on early and late tag relevance fusion schemes. Image search experiments on a large benchmark show that compared to a single measurement of tag relevance, fusing multiple tag relevance driven by diverse features results in better image search. Consequently, we obtain positive training examples with a higher accuracy.

Chapter 4. Social Negative Bootstrapping for Visual Categorization. To obtain negative examples without human interaction, in this chapter we go beyond random sampling by introducing a social negative bootstrapping approach. Given a visual category and a few positive examples, the proposed approach adaptively and iteratively harvests informative negatives from a large amount of social-tagged images. In order to reduce false negative examples, we design a virtual labeling procedure based on simple tag reasoning. Virtual labeling, in combination with adaptive sampling, enables us to identify the most misclassified negatives as the informative samples. As experiments on two image benchmarks and 650k virtually-labeled negative examples show, classifiers trained on such informative negative examples are more discriminative then classifiers trained on randomly sampled negatives.

7.1.2 Part II: Online Use

Chapter 5. Harvesting Social Images for Bi-Concept Search. Towards answering complex visual searches, we introduce in this chapter the notion of bi-concepts as a retrieval method for unlabeled images in which two concepts are co-occurring. Different from existing work, which focuses on combining detectors of individual concepts, we propose to learn bi-concept detectors directly. As the number of potential bi-concepts is gigantic, manually collecting training examples is infeasible. Instead, we propose a multimedia framework to collect de-noised positive as well as informative negative training examples from the social web. The ingredients of the framework are derived from the methods developed in Part I. We study the behavior of our bi-concept search engine using 1.2M social-tagged images as a data source. Our experiments show that directly learning bi-concepts is better than combining single-concept detectors.

Chapter 6. Personalizing Automated Image Annotation using Cross-Entropy. In this chapter, we aim for personalizing automated image annotation by jointly exploiting personalized tag statistics and content-based image annotation. We propose a cross-entropy based learning algorithm which personalizes a generic annotation model by learning from a user's multimedia tagging history. Using cross-entropy-minimization based Monte Carlo sampling, the proposed algorithm optimizes the personalization process in terms of a performance measurement which can be flexibly chosen. Automatic image annotation experiments with 5,315 realistic users

in the social web show that the proposed method compares favorably to a generic image annotation method and a method using personalized tag statistics only.

7.2 Future Directions

With the methods developed, this thesis has built the foundation of exploiting socially tagged images for visual search, but it is not meant to cover all aspects of this exciting multi-disciplinary field. The next generation search engines should go beyond search and allow to make sense of data, observations, and patterns on the social web. To that end, we consider the following directions important for future research.

To obtain accurate positive examples, an extended exploration will be to analyze other dimensions of the data including social notes, comments, and spontaneous structures such as photo groups. For acquiring examples with multiple concepts co-present where individual evidence tends to be limited, such a joint analysis may be beneficial. For tag relevance estimation, more voting models, especially ones that make use of social structure, can be considered in the future.

For obtaining informative negative examples, the proposed social negative bootstrapping approach employs a number of base classifiers to assess the informativeness of candidate negative examples. The time complexity of finding informative examples grows with the number of the base classifiers employed in each iteration. Thus, in order to favorably exploit the big data, accelerating the proposed approach will be a valuable topic for future work.

For complex visual searches, it will be interesting to extend the current bi-concept search engine by adding visual attributes to single concepts. For instance, adding the tag 'red' to a 'car' generates a more specific single concept 'red car'. Integrating such detailed single concepts into bi-concept search would lead to an image search engine capable of answering precise visual queries.

Last but not least, personalized multimedia content analysis will be an important direction to pursue. The social web provides opportunities than ever before to "understand" a user in a given situation. As we have shown in Chapter 6, having the knowledge of personal tagging preference makes visual content understanding much easier. As a user's information need may change over time, research on personalization methods capable of tracking such changes will be valuable for making sense of social multimedia in a personalized manner.

7.3 Conclusions

What is the value of socially tagged images for visual search? To answer the fundamental question of this thesis, we have decomposed it into five sub questions. Now, given what we have achieved, we draw the following conclusions.

W*hat determines the relevance of a social tag with respect to an image?* As the question is rooted in the subjective tagging nature of individual users, we need a mechanism to aggregate the users into a wise crowd which conducts objective tagging. The neighbor voting algorithm proposed in Chapter 2 is such a mechanism. By neighbor voting, tags relevant to the visual content receive more votes than tags irrelevant to the content. Experiments on image retrieval and tag ranking show that methods using learned tag relevance compare favorably to baseline methods without tag relevance estimates. The results allow us to conclude that the number of votes on a tag from an image's visual neighbors is a good indicator of the relevance of the tag with respect to the image.

How to fuse tag relevance estimators? Our study in Chapter 3 shows that the early and late tag relevance fusion schemes each have their merit. Early fusion, which directly manipulates the neighbor sets, is more effective for addressing concepts rarely tagged. Late fusion is more robust to differences between the data on which the method is trained and the data to which it is applied. Moreover, the LateFusion-Average method, which simply averages multiple tag relevance estimates, is comparable to its supervised alternatives, with a loss of 1.9% only. Thus, we recommend LateFusion-Average as the method of choice.

Which social images are informative negative examples? According to the results in Chapter 4, the negativeness of a socially tagged image can be automatically determined by the designed virtual labeling procedure, which exploits tag statistics and semantics. Given the virtually labeled negative examples, their informativeness for creating a visual concept classifier is determined by their probability of being misclassified. The most misclassified elements are the most informative. For harvesting informative negative examples from social images, the proposed social negative bootstrapping approach outperforms the state-of-the-art.

How to exploit social-tagged images for complex visual searches? We give an initial answer in Chapter 5 by establishing the notion of bi-concepts as a retrieval means. Compared to the social tagging baseline, the proposed multi-modal approach obtains bi-concept positive examples with doubled accuracy. The experimental results show that directly learning bi-concept detectors is better than (oracle) combinations of single-concept detectors, with a performance gain of 48%. We believe that for specific pre-defined bi-concepts, they already have great potential for use in advanced image search engines.

How to personalize automated image tagging with respect to a user's preference? The key is to strike a proper balance between multiple sources of evidence including personal tagging preference and content-based annotation models. Our study in Chapter 6 shows that, in general, richer personal tagging history leads to better personalized image annotation models. Even when a user has one tagged image available for training, we observe improvements for the majority of such one-training-image users. The experimental results allow us to conclude that personalization using the proposed cross-entropy based optimization is effective.

On the basis of the above reported theories, algorithms, and experiments, this thesis has revealed the value of socially tagged images for visual search and provides a basis for revealing universal knowledge on images and semantics. With the methodologies established, the thesis opens up promising avenues for image search which provides access to the semantics of the visual content, but without the need of manual annotation other than social tags.

Appendix A

Appendix

A.1 Learning Tag Relevance by Neighbor Voting

We proof *Theorem 1* and *Theorem 2* proposed in Chapter 2.

Theorem 1: Image ranking. *Given assumption 1 and assumption 2, tagRelevance yields an ideal image ranking for tag w, that is, for $I_1 \in R_w$ and $I_2 \in R_w^c$, we have $tagRelevance(w, I_1) > tagRelevance(w, I_2)$.*

Proof. Recall Eq. 2.8 and Eq. 2.9 that

$$tagRelevance(w, I_1) = k \cdot (P(w|R_w) - P(w|R_w^c))\, \varepsilon_{I_1,w},$$
$$tagRelevance(w, I_2) = k \cdot (P(w|R_w) - P(w|R_w^c))\, \varepsilon_{I_2,w},$$

we have

$$tagRelevance(w, I_1) - tagRelevance(w, I_2) = k \cdot (P(w|R_w) - P(w|R_w^c))\, (\varepsilon_{I_1,w} - \varepsilon_{I_2,w}).$$

Given assumption 1, we have

$$P(w|R_w) - P(w|\bar{\mathcal{R}}_w) > 0,$$

and given assumption 2, we get

$$\varepsilon_{I_1,w} - \varepsilon_{I_2,w} > 0.$$

Hence, $tagRelevance(w, I_1) > tagRelevance(w, I_2)$. Note that we only require $\varepsilon_{I_1,w} - \varepsilon_{I_2,w} > 0$, thereby the assumption 2, namely $\varepsilon_{I_1,w} > 0 > \varepsilon_{I_2,w}$, can be relaxed as $\varepsilon_{I_1,w} > \varepsilon_{I_2,w}$. We call the latter relaxed assumption 2. □

Theorem 2: Tag ranking. *Given assumption 1 and assumption 2, tagRelevance yields an ideal tag ranking for image I, that is, for two tags w_1 and w_2, if $I \in R_{w_1}$ and $I \in R^c_{w_2}$, we have $tagRelevance(w_1, I) > tagRelevance(w_2, I)$.*

Proof. Recall Eq. 2.8 and Eq. 2.9 that

$$\begin{aligned} tagRelevance(w_1, I) &= k \cdot \left(P(w_1|R_{w_1}) - P(w_1|R^c_{w_1})\right) \varepsilon_{I,w_1}, \\ tagRelevance(w_2, I) &= k \cdot \left(P(w_2|R_{w_2}) - P(w_2|R^c_{w_2})\right) \varepsilon_{I,w_2}. \end{aligned}$$

Given assumption 1, we have

$$\begin{aligned} P(w_1|R_{w_1}) - P(w_1|R^c_{w_1}) > 0, \\ P(w_2|R_{w_2}) - P(w_2|R^c_{w_2}) > 0. \end{aligned}$$

and given assumption 2, we get

$$\varepsilon_{I,w_1} > 0 > \varepsilon_{I,w_2}.$$

Note that multiplying positive factors does not change the direction of an inequation. Therefore, by multiplying the left side and the right side of the above inequation by $k\left(P(w_1|R_{w_1}) - P(w_1|R^c_{w_1})\right)$ and $k\left(P(w_2|R_{w_2}) - P(w_2|R^c_{w_2})\right)$ respectively, we obtain

$$k \cdot \left(P(w_1|R_{w_1}) - P(w_1|R^c_{w_1})\right) \varepsilon_{I,w_1} > 0 > k \cdot \left(P(w_2|R_{w_2}) - P(w_2|R^c_{w_2})\right) \varepsilon_{I,w_2}.$$

Hence, $tagRelevance(w_1, I) > tagRelevance(w_2, I)$. $\square$

Bibliography

[1] M. Allan and J. Verbeek. Ranking user-annotated images for multiple query terms. In *BMVC*, 2009.

[2] R. Aly, D. Hiemstra, A. de Vries, and F. de Jong. A probabilistic ranking framework using unobservable binary events for video search. In *CIVR*, 2008.

[3] A. Amir, M. Berg, S.-F. Chang, W. Hsu, G. Iyengar, C.-Y. Lin, M. Naphade, P. Natsev, C. Neti, H. Nock, J. Smith, B. Tseng, Y. Wu, and D. Zhang. IBM research TRECVID-2003 video retrieval system. In *TRECVID Workshop*, 2003.

[4] J. Aslam and M. Montague. Models for metasearch. In *SIGIR*, 2001.

[5] P. Atrey, M. Hossain, A. El Saddik, and M. Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia Syst.*, 16(6):345–379, 2010.

[6] E. Auchard. Flickr to map the world's latest photo hotspots. Reuters [Online], 2007, Nov. Available: `http://www.reuters.com/article/technologyNews/idUSHO94233920071119?sp=true`.

[7] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[8] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan. Matching words and pictures. *J. Mach. Learn. Res.*, 3(6):1107–1135, 2003.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, 2008.

[10] G. Begelman, P. Keller, and F. Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *WWW Collaborative Web Tagging Workshop*, 2006.

[11] B. Billerbeck and J. Zobel. Questioning query expansion: an examination of behaviour and parameters. In *Australasian database conference*, 2004.

[12] S. Börkur and R. van Zwol. Flickr tag recommendation based on collective knowledge. In *WWW*, 2008.

[13] L. Breiman. Bagging predictors. *J. Mach. Learn. Res.*, 24(2):123–140, 1996.

[14] G. Brown, J. Wyatt, R. Harris, and X. Yao. Diversity creation methods: A survey and categorisation. *Int. J. Information Fusion*, 6(1):5–20, 2005.

[15] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):394–410, 2007.

[16] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *SIGCOMM*, 2007.

[17] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[18] E. Chang, G. Kingshy, G. Sychay, and G. Wu. CBSA: content-based soft annotation for multimodal image retrieval using Bayes point machines. *IEEE Trans. Circuits Syst. Video Technol.*, 13(2):26–38, 2003.

[19] S.-F. Chang, W. Hsu, W. Jiang, L. Kennedy, D. Xu, A. Yanagawa, and E. Zavesky. Columbia university trecvid-2006 video search and high-level feature extraction. In *TRECVID workshop*, 2006.

[20] L. Chen, D. Xu, I. Tsang, and J. Luo. Tag-based web photo retrieval improved by batch mode re-tagging. In *CVPR*, 2010.

[21] T.-S. Chua, S.-Y. Neo, K.-Y. Li, G. Wang, R. Shi, M. Zhao, and H. Xu. TRECVID 2004 search and feature extraction task by NUS PRIS. In *TRECVID Workshop*, 2004.

[22] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng. NUS-WIDE: A real-world web image database from National University of Singapore. In *CIVR*, 2009.

[23] R. Cilibrasi and P. Vitanyi. The Google similarity distance. In *IEEE Trans. on Knowl. and Data Eng.*, volume 19, pages 370–383, 2007.

[24] C. Cusano, G. Ciocca, and R. Schettini. Image annotation using SVM. In *SPIE*, 2004.

[25] R. Datta, W. Ge, J. Li, and J. Wang. Toward bridging the annotation-retrieval gap in image search. *IEEE Multimedia*, 14(3):24–35, 2007.

[26] R. Datta, D. Joshi, J. Li, and J. Wang. Tagging over time: real-world image annotation by lightweight meta-learning. In *ACM Multimedia*, 2007.

[27] R. Datta, D. Joshi, J. Li, and J. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):1–60, 2008.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[29] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid. Evaluation of gist descriptors for web-scale image search. In *CIVR*, 2009.

[30] M. Duan, A. Ulges, T. Breuel, and X.-Q. Wu. Style modeling for tagging personal photo collections. In *CIVR*, 2009.

[31] M. Everingham, L. van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

[32] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

[33] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *ECCV*, 2004.

[34] G. Wangand D. Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *ICCV*, 2009.

[35] Y. Freund, R. Iyer, R. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, 2003.

[36] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55:119–139, 1997.

[37] A. Gallagher, C. Neustaedter, L. Cao, J. Luo, and T. Chen. Image annotation using personal calendars as context. In *ACM Multimedia*, 2008.

[38] P. Gehler and S. Nowozin. Let the kernel figure it out; principled learning of pre-processing for kernel classifiers. In *CVPR*, 2009.

[39] S. Golder and B. Huberman. Usage patterns of collaborative tagging systems. *J. Inf. Sci.*, 32(2):198–208, 2006.

[40] D. Grangier and S. Bengio. A discriminative kernel-based approach to rank images from text queries. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(8):1371–1384, 2008.

[41] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. TagProp: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *ICCV*, 2009.

[42] M. Guillaumin, J. Verbeek, and C. Schmid. Multimodal semi-supervised learning for image classification. In *CVPR*, 2010.

[43] A. Hauptmann, R. Yan, W.-H. Lin, M. Christel, and H. Wactlar. Can high-level concepts fill the semantic gap in video retrieval? a case study with broadcast news. *IEEE Trans. Multimedia*, 9(5):958–966, 2007.

[44] M. Heikkilä, M. Pietikäinen, and C. Schmid. Description of interest regions with local binary patterns. *Pattern Recogn.*, 42:425–436, 2009.

[45] E. Hörster, R. Lienhart, and M. Slaney. Image retrieval on large-scale image databases. In *CIVR*, 2007.

[46] W. Hsu, L. Kennedy, and S.-F. Chang. Video search reranking via information bottleneck principle. In *ACM Multimedia*, 2006.

[47] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. In *CVPR*, 1997.

[48] M. Huiskes, B. Thomee, and M. Lew. New trends and ideas in visual concept detection: the MIR flickr retrieval evaluation initiative. In *MIR*, 2010.

[49] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intell. Data Anal.*, 6(5):429–449, 2002.

[50] E. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

[51] Y. Jin, L. Khan, L. Wang, and M. Awad. Image annotations by combining multiple evidence & Wordnet. In *ACM Multimedia*, 2005.

[52] Y. Jing and S. Baluja. VisualRank: Applying pagerank to large-scale image search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(11):1877–1890, 2008.

[53] K. Jones, S. Walker, and S. Robertson. A probabilistic model of information retrieval: development and comparative experiments - part 2. *Inf. Process. Manage.*, 36(6):809–840, 2000.

[54] L. Kennedy, S.-F Chang, and I. Kozintsev. To search or to label?: Predicting the performance of search-based automatic image classifiers. In *MIR*, 2006.

[55] L. Kennedy, M. Naaman, S. Ahern, R. Nair, and T. Rattenbury. How Flickr helps us make sense of the world: Context and content in community-contributed media collections. In *ACM Multimedia*, 2007.

[56] S. Lee, W. De Neve, and Y. Ro. Image tag refinement along the 'what' dimension using tag categorization and neighbor voting. In *ICME*, 2010.

[57] J. Li and J. Wang. Real-time computerized annotation of pictures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(6):985–1002, 2008.

[58] L.-J. Li and L. Fei-Fei. OPTIMOL: Automatic online picture collection via incremental model learning. *Int. J. Comput. Vision*, 88(2):147–168, 2010.

[59] X. Li, L. Chen, L. Zhang, F. Lin, and W.-Y. Ma. Image annotation by large-scale content-based image retrieval. In *ACM Multimedia*, 2006.

[60] X. Li, E. Gavves, C. Snoek, M. Worring, and A. Smeulders. Personalizing automated image annotation using cross-entropy. In *ACM Multimedia*, 2011.

[61] X. Li and C. Snoek. Visual categorization with negative examples for free. In *ACM Multimedia*, 2009.

[62] X. Li, C. Snoek, and M. Worring. Learning tag relevance by neighbor voting for social image retrieval. In *MIR*, 2008.

[63] X. Li, C. Snoek, and M. Worring. Annotating images by harnessing worldwide user-tagged photos. In *ICASSP*, 2009.

[64] X. Li, C. Snoek, and M. Worring. Learning social tag relevance by neighbor voting. *IEEE Trans. Multimedia*, 11(7):1310–1322, 2009.

[65] X. Li, C. Snoek, and M. Worring. Unsupervised multi-feature tag relevance learning for social image retrieval. In *CIVR*, 2010.

[66] X. Li, C. Snoek, M. Worring, and A. Smeulders. Harvesting social images for bi-concept search. submitted.

[67] X. Li, C. Snoek, M. Worring, and A. Smeulders. Tag relevance fusion for social image search. submitted.

[68] X. Li, C. Snoek, M. Worring, and A. Smeulders. Social negative bootstrapping for visual categorization. In *ICMR*, 2011.

[69] X. Li, D. Wang, J. Li, and B. Zhang. Video search in concept subspace: A text-like paradigm. In *CIVR*, 2007.

[70] Z. Li, L. Zhang, and W.-Y. Ma. Delivering online advertisements inside images. In *ACM Multimedia*, 2008.

[71] H.-T. Lin, C.-J. Lin, and R. Weng. A note on Platt's probabilistic outputs for support vector machines. *J. Mach. Learn. Res.*, 68:267–276, 2007.

[72] W.-H. Lin and A. Hauptmann. Web image retrieval re-ranking with relevance model. In *Web Intelligence*, 2003.

[73] B. Liu, Y. Dai, X. Li, W. Lee, and P. Yu. Building text classifiers using positive and unlabeled examples. In *ICDM*, 2003.

[74] D. Liu, X.-S. Hua, M. Wang, and H.-J. Zhang. Image retagging. In *ACM Multimedia*, 2010.

[75] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang. Tag ranking. In *WWW*, 2009.

[76] J. Liu, B. Wang, M. Li, Z. Li, W.-Y. Ma, H. Lu, and S. Ma. Dual cross-media relevance model for image annotation. In *ACM Multimedia*, 2007.

[77] Y. Liu, D. Xu, I. Tsang, and J. Luo. Textual query of personal photos facilitated by large-scale web data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):1022–1036, 2011.

[78] Y. Lu, L. Zhang, J. Liu, and Q. Tian. Constructing concept lexica with small semantic gaps. *IEEE Trans. Multimedia*, 12(4):288–299, 2010.

[79] J. Luo, J. Yu, D. Joshi, and W. Hao. Event recognition: viewing the world with a third eye. In *ACM Multimedia*, 2008.

[80] A. Makadia, V. Pavlovic, and S. Kumar. Baselines for image annotation. *Int. J. Comput. Vision*, 90(1):88–105, 2010.

[81] K. Matusiak. Towards user-centered indexing in digital image collections. *OCLC Sys. and Services*, 22(4):283–298, 2006.

[82] D. Metzler and B. Croft. Linear feature-based models for information retrieval. *Inf. Retr.*, 10(3):257–274, 2007.

[83] R. Mihalcea, C. Corley, and C. Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*, 2006.

[84] F. Nah. A study on tolerable waiting time: how long are Web users willing to wait? *Jour. Behaviour and Information Technology*, 23(3):153–163, 2004.

[85] M. Naphade, J. Smith, J. Tešić, 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.

[86] A. Natsev, A. Haubold, J. Tešić, L. Xie, and R. Yan. Semantic concept-based query expansion and re-ranking for multimedia retrieval. In *ACM Multimedia*, 2007.

[87] A. Natsev, M. Naphade, and J. Tešić. Learning the semantics of multimedia queries and concepts from a small number of examples. In *ACM Multimedia*, 2005.

[88] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42(3):145–175, 2001.

[89] X. Olivares, M. Ciaramita, and R. van Zwol. Boosting image retrieval through aggregating search results based on visual annotations. In *ACM Multimedia*, 2008.

[90] G. Park, Y. Baek, and H.-K. Lee. Majority based ranking approach in web image retrieval. In *CIVR*, 2003.

[91] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, and T. Tuytelaars. A thousand words in a scene. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(9):1575–1589, 2007.

[92] M. Radovanović, A. Nanopoulos, and M. Ivanović. Hubs in space: Popular nearest neighbors in high-dimensional data. *J. Mach. Learn. Res.*, 11:2487–2531, 2010.

[93] R. Rubinstein and D. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*. Springer-Verlag, New York, 2004.

[94] Y. Rui and T. Huang. Optimizing learning in image retrieval. In *CVPR*, 2000.

[95] B. Russell, A. Torralba, K. Murphy, and W. Freeman. LabelMe: A database and web-based tool for image annotation. *Int. J. Comput. Vision*, 77(1-3):157–173, 2008.

[96] M. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011.

[97] P. Sandhaus and S. Boll. Semantic analysis and retrieval in personal and social photo collections. *Multimedia Tools Appl.*, 51(1):5–33, 2011.

[98] N. Sawant, R. Datta, J. Li, and J. Wang. Quest for relevant tags using local interaction networks and visual content. In *MIR*, 2010.

[99] N. Sawant, J. Li, and J. Wang. Automatic image semantic interpretation using social action and tagging data. *Multimedia Tools Appl.*, 51:213–246, 2011.

[100] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33:754–766, 2011.

[101] D. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, New York, 1992.

[102] D. Shamma, R. Shaw, P. Shafton, and Y. Liu. Watch what I watch: using community activity to understand content. In *MIR*, 2007.

[103] Y. Shen and J. Fan. Leveraging loosely-tagged images and inter-object correlations for tag recommendation. In *ACM Multimedia*, 2010.

[104] P. Sinha and R. Jain. Classification and annotation of digital photos using optical context data. In *CIVR*, 2008.

[105] A. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVID. In *MIR*, 2006.

[106] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000.

[107] C. Snoek, B. Huurnink, L. Hollink, M. de Rijke, G. Schreiber, and M. Worring. Adding semantics to detectors for video retrieval. *IEEE Trans. Multimedia*, 9(5):975–986, 2007.

[108] C. Snoek and A. Smeulders. Visual-concept search solved? *IEEE Computer*, 43(6):76–78, 2010.

[109] C. Snoek and M. Worring. Concept-based video retrieval. *Found. Trends Inf. Retr.*, 2:215–322, 2009.

[110] C. Snoek, M. Worring, and A. Smeulders. Early versus late fusion in semantic video analysis. In *ACM Multimedia*, 2005.

[111] S. Sonnenburg, G. Rätsch, S. Henschel, C. Widmer, J. Behr, A. Zien, F. de Bona, A. Binder, C. Gehl, and V. Franc. The shogun machine learning toolbox. *J. Mach. Learn. Res.*, 11:1799–1802, 2010.

[112] A. Sun and S. Bhowmick. Quantifying tag representativeness of visual content of social images. In *ACM Multimedia*, 2010.

[113] A. Sun, S. Bhowmick, K. Nguyen, and G. Bai. Tag-based social image retrieval: An empirical evaluation. *J. Am. Soc. Inf. Sci. Technol.*, 2011. in press.

[114] J. Surowiecki. *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*. Doubleday; Anchor, 2004.

[115] J. Tang, R. Hong, S. Yan, T.-S. Chua, G.-J. Qi, and R. Jain. Image annotation by knn-sparse graph-based label propagation over noisily tagged web images. *ACM Trans. Intell. Syst. Technol.*, 2:14:1–14:15, 2011.

[116] D. Tax. *One-class classification*. PhD thesis, Delft University of Technology, 2001.

[117] K. Tieu and P. Viola. Boosting image retrieval. *Int. J. Comput. Vision*, 56(1-2):17–36, 2004.

[118] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *ACM Multimedia*, 2001.

[119] A. Torralba, R. Fergus, and W. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(11):1958–1970, 2008.

[120] J. Uijlings, A. Smeulders, and R. Scha. Real-time visual concept classification. *IEEE Trans. Multimedia*, 12(7):665–681, 2010.

[121] A. Ulges, C. Schulze, M. Koch, and T. Breuel. Learning automatic concept detectors from online video. *Comput. Vis. Image Underst.*, 114(4):429–438, 2010.

[122] K. van de Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32:1582–1596, 2010.

[123] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 2000.

[124] P. Viola and M. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57:137–154, 2004.

[125] C. Wang, F. Jing, L. Zhang, and H.-J. Zhang. Scalable search-based image annotation. *Multimedia Systs.*, 14(4):205–220, 2008.

[126] D. Wang, X. Liu, L. Luo, J. Li, and B. Zhang. Video Diver: generic video indexing with diverse features. In *MIR*, 2007.

[127] J. Wang, J. Li, and G. Wiederhold. SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23:947–963, 2001.

[128] M. Wang, X.-S. Hua, R. Hong, J. Tang, G.-J. Qi, and Y. Song. Unified video annotation via multigraph learning. *IEEE Trans. Cir. and Sys. for Video Technol.*, 19:733–746, 2009.

[129] M. Wang, X.-S. Hua, J. Tang, and R. Hong. Beyond distance measurement: constructing neighborhood similarity for video annotation. *IEEE Trans. Multimedia*, 11:465–476, 2009.

[130] X.-J. Wang, L. Zhang, X. Li, and W.-Y. Ma. Annotating images by mining image search results. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(11):1919–1932, 2008.

[131] X.-J. Wang, L. Zhang, M. Liu, Y. Li, and W.-Y. Ma. ARISTA - image search to annotation on billions of web photos. In *CVPR*, 2010.

[132] X.-Y. Wei, Y.-G. Jiang, and C-W. Ngo. Concept-driven multi-modality fusion for video search. *IEEE Trans. Circuits Syst. Video Techn.*, 21(1):62–73, 2011.

[133] K. Weinberger, M. Slaney, and R. van Zwol. Resolving tag ambiguity. In *ACM Multimedia*, 2008.

[134] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *J. Mach. Learn. Res.*, 81:21–35, 2010.

[135] L. Wu, S. Hoi, R. Jin, J. Zhu, and N. Yu. Distance metric learning from uncertain side information with application to automated photo tagging. In *ACM Multimedia*, 2009.

[136] L. Wu, X.-S. Hua, N. Yu, W.-Y. Ma, and S. Li. Flickr distance. In *ACM Multimedia*, 2008.

[137] Y. Wu, E. Chang, K. Chang, and J. Smith. Optimal multimodal fusion for multimedia data analysis. In *ACM Multimedia*, 2004.

[138] H. Xu, J. Wang, X.-S. Hua, and S. Li. Tag refinement by regularized LDA. In *ACM Multimedia*, 2009.

[139] R. Yan, A. Hauptmann, and R. Jin. Multimedia search with pseudo-relevance feedback. In *CIVR*, pages 649–654, 2003.

[140] R. Yan, A. Hauptmann, and R. Jin. Negative pseudo-relevance feedback in content-based video retrieval. In *ACM Multimedia*, 2003.

[141] R. Yan and Alexander Hauptmann. The combination limit in multimedia retrieval. In *ACM Multimedia*, 2003.

[142] K. Yanai and K. Barnard. Probabilistic web image gathering. In *MIR*, 2005.

[143] K. Yang, X.-H. Hua, M. Wang, and H.-J. Zhang. Tag tagging: towards more descriptive keywords of image content. *IEEE Trans. Multimedia*, 2011. in press.

[144] T. Yeh, J. Lee, and T. Darrell. Photo-based question answering. In *ACM Multimedia*, 2008.

[145] H. Yu, M. Li, H.-J. Zhang, and J. Feng. Color texture moment for content-based image retrieval. In *ICIP*, 2002.

[146] J. Yuan, Z.-J. Zha, Y.-T. Zheng, M. Wang, X. Zhou, and T.-S. Chua. Utilizing related samples to enhance interactive content-based video search. *IEEE Trans. Multimedia*, 2011. in press.

[147] Z.-J. Zha, L. Yang, T. Mei, M. Wang, and Z. Wang. Visual query suggestion. In *ACM Multimedia*, 2009.

[148] L. Zhang, L. Chen, F. Jing, K. Deng, and W.-Y. Ma. EnjoyPhoto: a vertical image search engine for enjoying high-quality photos. In *ACM Multimedia*, 2006.

[149] G. Zhu, S. Yan, and Y. Ma. Image tag refinement towards low-rank, content-tag prior and error sparsity. In *ACM Multimedia*, 2010.

[150] M. Zhu. Recall, precision and average precision. Technical report, University of Waterloo, 2004. Working Paper 2004-09.

[151] S. Zhu, G. Wang, C.-W. Ngo, and Y.-G. Jiang. On the sampling of web images for learning visual concept classifiers. In *CIVR*, 2010.

Samenvatting*

In een wereld waarin de hoeveelheid digitale afbeeldingen alsmaar groeit is inhoud gebaseerd zoeken een belangrijk en wetenschappelijk uitdagend probleem in ICT onderzoek. Dit proefschrift stelt voor het probleem aan te pakken door te leren van sociale media. De fundamentele vraag die in dit proefschrift wordt beschouwd is: *Wat is de waarde van beelden met social tags voor visueel zoeken?*

Om te leren van sociale media stellen we het "neighbor voting" algoritme (hoofdstuk 2) en de multi-kenmerkvariant (hoofdstuk 3) voor om te bepalen of wat mensen spontaan als tag aan een beeld hebben toegevoegd ook daadwerkelijk te zien is in het beeld. De twee algoritmes worden gebruikt om positieve voorbeelden van hoge kwaliteit te vinden voor het leren van automatische tag algoritmes. Om negatieve voorbeelden te verkrijgen zonder dat we deze handmatig moeten selecteren gaan we voorbij aan de klassieke methode van willekeurig selecteren en stellen bootstrapping met informatieve voorbeelden voor (hoofdstuk 4). Voor het beantwoorden van complexe zoekvragen introduceren we het idee van bi-concepten waarmee niet-gelabelde beelden kunnen worden teruggevonden waarin twee concepten samen voorkomen (hoofdstuk 5). Tenslotte, omdat alle gebruikers hun eigen associaties hebben met de semantiek van de beelden stellen we het gepersonaliseerd automatisch taggen van beelden voor. Hierin wordt het eerdere tagging gedrag van gebruikers en analyse van de inhoud van de beelden aan elkaar gekoppeld en daarna geoptimaliseerd op basis van Monte Carlo sampling (Hoofdstuk 6).

Op basis van de ontwikkelde theoriën, de algoritmes en de experimenten heeft dit proefschrift de waarde van beelden met sociale tags voor inhoud gebaseerd visueel zoeken aangetoond. Het geeft een basis voor het blootleggen van algemene kennis over beelden en semantiek. De methodieken openen veelbelovende mogelijkheden voor beeld zoekmachines die toegang geven tot de semantische inhoud van beelden, maar

*Summary, in Dutch.

zonder daarvoor handmatig beelden te hoeven annoteren anders dan social tagging.

Acknowledgements

Being a PhD is a non-trivial trip. I would like to use this dedicated section to thank people who practically or mentally helped me accomplish the trip.

First and foremost, I wish to express my gratitude to my supervisors: Marcel and Cees. Without their guidance, encouragement, and excellent vision of visual search, I would have not been able to finish the thesis.

My connection with Marcel can trace back to late 2006, when I received a recruitment flyer entitled "PhD students for TRECVID relevant project". After a phone interview on 20th of Dec. 2006, Marcel gave me the offer. Of course, by then he did not realize how hard it would be to guide a Chinese student who is lack of patience and often sticks to his own intuitions. Even when the path to my PhD has been illustrated by his supervision, he still had to correct the machine translated samenvatting.

Though being the first PhD student of someone could be risky, this is not the case for me. It seems that Cees comes naturally with intellectual supervision skills. As my daily supervisor, he guided me in many aspects: scientific thinking, defining research questions, paper writing, oral presentation, poster design, rebuttals, reviews, mierenneuken for polishing papers, etc. I would to express my appreciation for his generous offer to extend my contract, letting me continue to finish the thesis.

It is my fortunate to have Arnold as my promotor. Viewing from a high level with his sharp vision and years of experience, he always inspires me in one way or another, unless culture differences make his messages not well perceived.

I would like to thank Dennis, Koen, Jasper, Sander, and Michiel for the Impala framework and the powerful feature descriptors, which greatly facilitated my experiments.

Also, I thank Chen Longyuan, Li Hairong, Liu Lei, Wu Mengxiao, and Zhang Liying for their participation in user studies, and Daan Odijk for crawling data. Special thanks go to Svetlana for designing the cover.

It is my pleasure to work in the ISIS group: Michael, Ivo, and Jan as my ping pong mates for years, Victoria, Vladimir, Dung, and Stratios for sharing offices, Cor, Jan-Mark, and Theo for interesting discussions, Gosia and Hamdi for the ISIS dinners, Daan and Bouke for sharing conference hotels, and our secretary Virginie who helped me much in the past years.

I would like to thank Liu Fangbin and Guang Liang for helping me settle down when I arrived in Amsterdam in 2007. Also, I thank other Chinese friends at Amsterdam, in particular those from the Science Park and Gaoshangshequ rim: Dai Guowen, Li Jianan, Li Bei, Song Yang, Tang Nan, Wang Xiang, Zhang Ling, Zhang Ying, Zhang Zhen, Zhao Jing, Gao Bo & Ma Huiye, Wu Jun & Liao Hong, Yin Si & Zhou Fujin, Zhang Xu & Yang Hong, and my former roommates: Ju Fengkui, Li Chao, and Wang Yanjing. Thanks to them, my 4.5-year life at Amsterdam was not purely scientific :-)

最后，我要特别感谢我的父母和我的妻子王思佳女士。你们的支持和鼓励是我积极工作的动力和快乐生活的源泉！

Xirong

Amsterdam
27-01-2012