Towards Indexing Representative Images on The Web

Xin-Jing Wang Zheng Xu[1*] Lei Zhang Ce Liu[2] Yong Rui
Microsoft Research Asia
[1]University of Science & Technology of China
[2]Microsoft Research New England
{xjwang,v-zxu, leizhang, celiu, yongrui}@microsoft.com

ABSTRACT

Even after 20 years of research on real-world image retrieval, there is still a big gap between what search engines can provide and what users expect to see. To bridge this gap, we present an image knowledge base, ImageKB, a graph representation of structured entities, categories, and representative images, as a new basis for practical image indexing and search. ImageKB is automatically constructed via a both bottom-up and top-down, scalable approach that efficiently matches 2 billion web images onto an ontology with millions of nodes. Our approach consists of identifying duplicate image clusters from billions of images, obtaining a candidate set of entities and their images, discovering definitive texts to represent an image and identifying representative images for an entity. To date, ImageKB contains 235.3M representative images corresponding to 0.52M entities, much larger than the state-of-the-art alternative ImageNet that contains 14.2M images for 0.02M synsets. Compared to existing image databases, ImageKB reflects the distributions of both images on the web and users' interests, contains rich semantic descriptions for images and entities, and can be widely used for both text to image search and image to text understanding.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*selection process*; I.5.4 [**Pattern Recognition**]: Applications—*computer vision, text processing*; H.2.8 [**Information Systems**]: Database Management—*Image databases*

General Terms

Algorithms, Performance

*The work was done in Microsoft Research Asia.

MM'12, October 29–November 2, 2012, Nara, Japan.

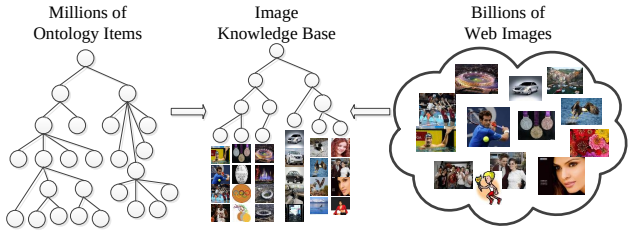

Figure 1: The challenge of ImageKB generation is how to match billions of images onto millions of items of an ontology.

Keywords

Image understanding, large-scale text to image translation, image knowledge base

1. INTRODUCTION

Since the WebSeek project[1] in 1996, there has been tremendous amount of effort in indexing images from the Web [2]. While significant progresses have been made, there still exist gaps between what existing techniques can provide and what users expect to see[3, 4]. This gap reveals the lack of sematic understanding of both images and queries in existing image indexing/search systems, resulting in deficiency of *relevance*, *informativeness*, *comprehensiveness* and *coverage*.

Current major commercial search engines such as Bing and Google index web images by treating them as documents using surrounding texts. Because noisy text tags often do not reflect the true image content, images returned by these search engines may not be *relevant* to users' queries. As billions of images are randomly indexed without top-down management, the search results may not be *informative*, e.g. duplicate images or images with very similar contents are returned so that limited information is delivered. Due to the same reason, search results also suffer from lack of *comprehensiveness*, especially for ambiguous queries. For example, for query "apple", some people intend for fruits while others intend for Apple products. Search engines can hardly distinguish multiple intents for one query, and therefore often fail to show images of all possibilities, and cannot separate them in display to disambiguate user intent.

Although the quality of search engines has been improving rapidly because of the increasing amount of user clicks, this massive crowdsourcing does not fundamentally solve these issues.

Contrast to this bottom-up approach in indexing web im-

Table 1: Overlap of Vocabularies with Query Log

		WordNet[7]	NeedleSeek[8]	ImageKB
#total item (#total category)		117,023 (26,150)	12.83M (185,158)	155.02M (-)
exact match	#item∩qlog (% of qlog)	88,724 (0.03%)	4.76M (1.85%)	13.26M (5.16%)
	#phrase∩qlog (% of qlog)	41,132 (0.02%)	3.70M (1.44%)	12.43M (4.84%)
	#category∩qlog (% of qlog)	20,044 (0.007%)	6,425 (0.003%)	- (-)
partial match	#item∩qlog (% of qlog)	94,105 (0.04%)	6.57M (2.49%)	21.22M (8.03%)
	#phrase∩qlog (% of qlog)	43,823 (0.02%)	4.88M (1.85%)	20.06M (7.59%)
	#category∩qlog (% of qlog)	20,973 (0.008%)	7,927 (0.003%)	- (-)

item: a noun in WordNet (single letters and single digits removed), a concept in NeedleSeek, or an entity in our approach
category: a non-leaf noun in WordNet, or a category in NeedleSeek. ImageKB uses NeedleSeek categories.
phrase: an item which has more than one words.
xxx∩qlog: the intersection of xxx and a query log
% of qlog: the coverage of a query log
exact match: an item exactly matches a query
partial match:exact match or matches a subphrase of a query

ages by search engines, researchers have started to create large image databases [5, 6] using a top-down approach. Based on pre-defined vocabularies (e.g. WordNet[7]), images are collected through querying search engines with items in the vocabularies. However, since such vocabularies are manually built with limited scales, the *coverage* on general user interests is too small to be used by search engines. For instance, our evaluation shows that the overlap between the WordNet vocabulary and a six-month query log of Bing is only 0.03%, as shown in Table 1).

To bridge the gap and to overcome these issues, we want to assign correct semantics to images and to manage images according to human knowledge. In this paper, we introduce a novel, scalable, both bottom-up and top-down approach to automatically generating a large-scale image knowledge base, ImageKB. The key of our approach is to associate billions of web images with an immense ontology of human knowledge.

ImageKB contains three types of information: 1)⟨entity, category⟩ pairs to represent high-level human knowledge, 2) ranked representative images for each pair, and 3) links between categories to indicate certain relationships. We call a general concept an *item* (e.g. significant, apple) and a visualizable item an *entity* (e.g. green, apple). An entity can be a concrete physical object, an event, an abstract concept, etc, as long as representative images can be identified to visualize it. The semantic class of an entity is called a *category*, e.g. since apple is a type of fruit, fruit is a category name. A category itself can be an entity if it is visualizable.

We adopt a both bottom-up and top-down strategy to generate ImageKB by mining from 2B web images dumped from Bing search engine and associating them with an ontology NeedleSeek [8] (an automatical ontology construction approach by mining webpages, see Section 4.1). In the bottom-up step, we propose a novel duplicate discovery approach to find duplicate image clusters and annotate each cluster by aggregating the surrounding texts of the duplicates. These clusters with text annotations form the set of candidate entities. We build an inverted index for the candidate entities for efficiency. There are in total 155.2M entities and 569.2M images in the inverted index. In the top-down step, we match the candidate entities and images to NeedleSeek, an ontology with millions of nodes, to associate images with entities in the NeedleSeek through robust filtering and ranking.

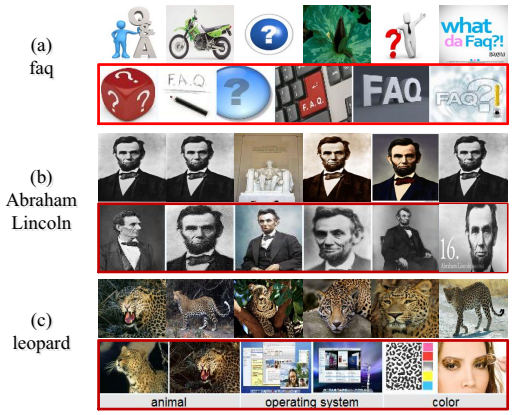

Figure 2: Examples of entities, top Bing search results, and our suggestions (in red block). Our method improves Bing on (a)relevance, (b)informativeness, and (c)comprehensiveness.

ImageKB has much larger coverage than existing, research-purpose databases. Table 1 evaluates the overlap of WordNet and a six-month log of 264.17M unique user queries collected from Bing image search during November 2011 to April 2012. All WordNet nouns cover only 0.03% user queries, which is 62 and 172 times smaller than those of NeedleSeek[8] and our approach[1]. By far, ImageKB contains 0.52M entities and 235.3M representative images, compared to 14.2M images for 21.8K synsets by ImageNet[6] and 80M tiny images for 53.5K English nouns by Visual Dictionary[5], which are the state-of-the-art alternatives.

ImageKB tries to tackle the aforementioned four issues of indexing web images by providing following advantages:

1. Scale. ImageKB is large enough for practical use, i.e. the identified visual entities have a good coverage on user queries, and each entity is associated with a large number of images.
2. Content. The images associated with an entity are not only definitive and representative, but also diverse in appearance. Meanwhile, each image is associated with rich, definitive texts extracted from surrounding texts which describe the image's content, compared to existing alternatives that generally show an entity name for a group of images[9, 10, 11, 5, 6]
3. Structure. ImageKB has a graph structure from which hierarchy and relationships between entities can be inferred.

Our intention of generating a knowledge base towards covering all visual entities and providing representative images for each entity is twofold. First, ImageKB can foster computer vision research in many aspects, e.g. object recognition[6], distance metric learning[12], and image annotation[13]. Second, ImageKB can redefine the infrastructure of image search engines. As it is challenging for a search engine to index trillions of images on the Web [14], ImageKB provides a selected image set to divide-and-conquer the image

[1]The details of how the vocabularies of our approach and NeedleSeek[8] are provided in Section 3 and Section 4.1 respectively.

indexing problem - instead of directly working on a generic algorithm to order images in an index, we can order the index by putting the "good" images that are high-quality, relevant, and user-interested, higher than the "bad" ones that are of low-quality and less-interested, so that more accurate images can be processed in fixed query evaluation time, or less evaluation time is needed for a fixed number of returned images. Meanwhile, knowledge learnt from the "good" images can be applied to improve the ordering of "bad" ones, so as to improve the quality of the whole image index[13, 12].

The paper is organized as follows. In Section 2, we present some facts of ImageKB, including the framework, its structure, and some statistics to date. From Section 3 to Section 4, we detail the steps of ImageKB generation. Detailed evaluations are given in Section 5 and we conclude our work in Section 6 .

2. IMAGEKB - SUMMARY

In this section, we outline the process of ImageKB generation, the structure of ImageKB, and a summary of the statistics revealing its scale and practicality.

2.1 The Framework

The basic idea of ImageKB construction is first to generate candidate entities and images from a large-scale dataset of web images, and then remove noisy entities and identify a ranked list of the most representative images for each remained entity. Our approach consists of two steps: a duplicate image discovery approach to obtain candidate entities and images from 2B web images, and an algorithm to match billions of images onto an ontology with millions of nodes. Both these algorithms are efficient and scalable for billions of images. Fig.3 summarizes this process: ImageKB obtains candidate entities and images by image annotation, and filters and ranks images by text mining. Both approaches leverage duplicate images to generate definitive texts as features.

Candidate vocabulary and image generation. In ImageKB, entities are defined as terms that describe the semantics of images, and representative images are images that visualize the semantics of corresponding entities. Therefore, to identify terms that hit the semantics of images is the key. We adopt a data-driven annotation approach[15, 16, 17] to achieve this goal (Fig.3(1b)), while the images that are annotated by the same term are candidate images of this term. The annotation is based on duplicate image clusters generated by a duplicate discovery approach (Fig.3(1a)). Fig.3(1) illustrates this process.

Using duplicate images has following advantages. Effective image annotation can be performed on duplicate images[16], the number of copies an image has on the Web suggests popularity among users and representativeness to an entity, and the result can be directly used for informativeness - since duplicates are discovered, they can be directly removed in ImageKB.

This step tackles the coverage and informativeness issues. The details are given in Section 3.

Image filtering and ranking. The task of this step is to disambiguate an entity and to classify the corresponding candidate images into the different semantics an entity may have (we used categories to differentiate semantics), and to output a ranked image list for each <entity, category> pair. The key idea is to measure the degree of an image representing a category of a certain entity. Fig.3(2) illustrates this process.

We leveraged a term-category look-up table to obtain two types of knowledge for image filtering and ranking: the categories of an entity, and the textual descriptors of a category w.r.t. an entity. Section 4 provides the details. This step addresses the relevance and comprehensiveness problems simultaneously.

2.2 The Structure

ImageKB is a large graph with overlapping hierarchy. Fig.4 illustrates a subgraph on "product" and "dog", and some of their related categories. There are in total 2,118 entities of "dog", 42 of which belong to multiple categories. In addition, there are 53,176 entities of "product". ImageKB contains both leaf (i.e. no hyponyms, e.g. "bordercollie") and categorical entities (e.g. "toy dog"), and connects categories via entities (e.g. "dog" overlaps "product" on entities "dog clothes", "dog book", "dog shelter", etc.). Each image in ImageKB is also associated with a bunch of text to describe the content of this image, which are automatically generated from the surrounding texts. Fig.7 shows six real examples of apple images and their selected text.

The overlapping hierarchy has high commercial values. For example, it can directly be used for "related searches" for commercial search engines[2]. On the other hand, though the local hierarchies of ImageKB are flat, as contrast to existing datasets[6, 5] generated based on WordNet[7], we can also build branch hierarchies from ImageKB. Fig.5 illustrate this idea. The images in the figure all come from ImageKB. However, though it was shown that a branch hierarchy can help distance metric learning[12], it is still unclear if such metrics has good generalization capability for search engines due to the gap between WordNet vocabulary and real user queries. In fact, we matched the ImageNet[6] hierarchy of "dog" (i.e. "canine" → "carnivore" → "placental" → "mammal") to the six-month query log of Bing, and found that the four terms have been queried 374, 1694, 41, and 1177 times respectively, which are negligible compared to the 2.6M times of the top query.

2.3 Statistics of ImageKB

Table 2 summaries a few statistics of ImageKB[3]. Table 2(a) shows that there are in total about 0.52M entities in ImageKB, with 0.48M of them single-category entities and 38.6K multi-category entities (e.g. "hearing dog" which indicates either a dog or a service). Some entities are quite popular - about 40.0K entities index more than 500 images per entity whereas 22.4K entities have more than 1K images. Note that indexing 500-1K images per entity is one target of ImageNet[6].

Table 2(b) gives statistics on the images in ImageKB. In total, there are 235.3M images and 202.4M of them have unique semantics, i.e. single-category. There are 202.4M and 190.1M images corresponding to "size>500" and "size>1K"

[2]On image search result page of modern commercial search engines such as Google and Bing, there is a row of "related searches" above the image search result panel, which suggests related hot queries to the current one.

[3]These statistics are based on our approach that runs to date. We expect ImageKB to grow rapidly in the future.

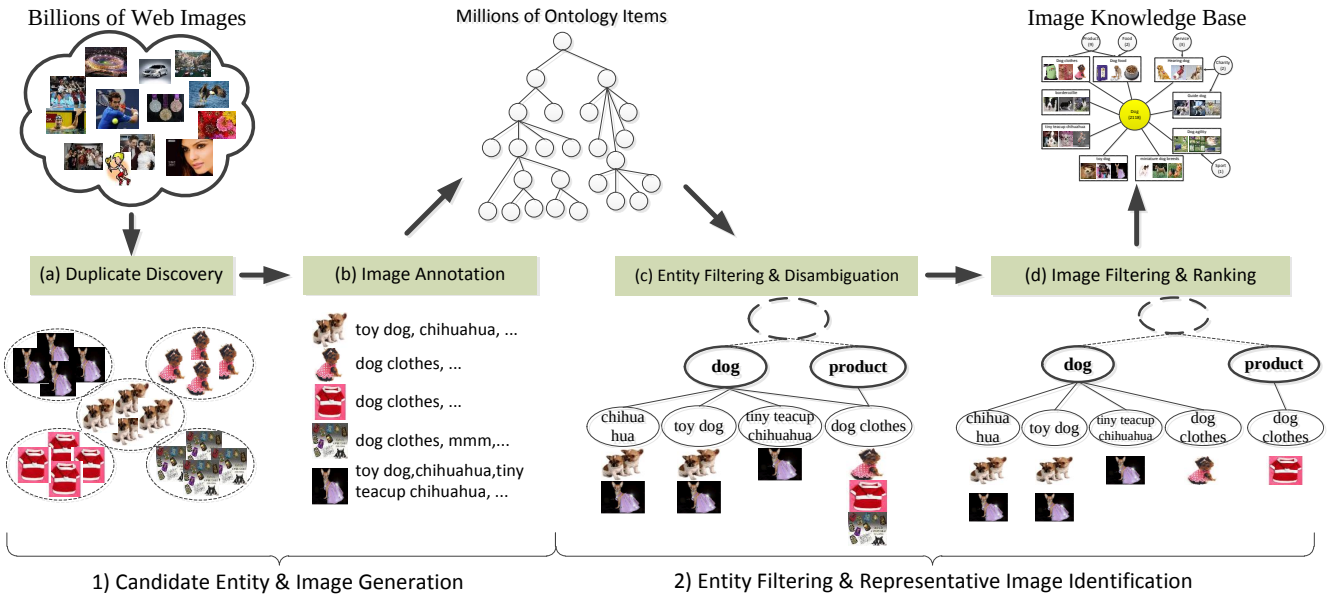

Figure 3: The framework of ImageKB construction: 1) generate candidate entities and their images by (a) discovering all duplicate image clusters from 2B images and (b) annotate the clusters, which gives the index on entities and images; 2) filter entities and identify representative images by (c) generate ⟨entity, category⟩ pairs for entity disambiguation by looking up a term-category table, and (d) filtering and scoring an image against a ⟨entity, category⟩ pair. ImageKB is then a space of ⟨entity, category, image list⟩ tuples.

Table 2: Properties of ImageKB

(a) Number of Entities

total	single-cate	multi-cate	total (size>500)	total (size>1K)
518,072	479,471	38,601	40,027	22,393

*total(size> n): means the total number of entities which have more than n images. 500-1K images per entity is a goal of ImageNet.

(b) Number of Images

total	single-cate	multi-cate	total(size>500)	total(size>1K)
235.3M	202.4M	32.9M	202.4M	190.1M

(c) Average Precision & Recall on Top 10

	overall	single-category	multi-category
Avg. Precision	0.80	0.82	0.78
Avg. Recall	0.31	0.35	0.27

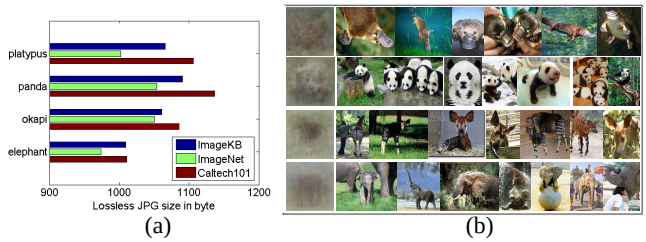

Figure 6: The diversity of ImageKB is between ImageNet[6] and Caltech101[9] because ImageKB identify representative images of which the main objects are large and centralized. (a)Comparison of lossless JPG file sizes of average images for four entities. A smaller size means a more diverse result. (b)Example images from ImageKB and the average images of entities in (a).

entities respectively. Therefore, each of "size>500" and "size>1K" entities has 5,056 and 8,488 images on average.

Compared to ImageNet[6] which holds 14.2M images and 21.8K synsets to date, ImageKB is much larger. In fact, 86,216 out of 117,023 WordNet[7] terms[4] are covered by ImageKB. Meanwhile, since ImageKB is mined from billions of real Web data and leverages popular images (i.e. duplicate images), it aligns better with users' interest than existing large-scale datasets[11, 5, 6], as suggested by Table 1. By exact match, ImageKB vocabulary covers 5.16% of real queries, whereas by partial match, the number increases to 8.03%.

We use the same approach as proposed by ImageNet[6] to measure the diversity of ImageKB, i.e. to generate an average image from randomly sampled images of a certain entity and save it in a lossless JPG file. The smaller the JPG file size, the more diverse the dataset may be. All Caltech101 images and an equal number of randomly sampled images from ImageNet and ImageKB are used for this evaluation. Fig.6 shows that the diversity of ImageKB is between ImageNet[6] and Caltech101[9]. This is reasonable because though ImageKB takes web images as input, it targets at finding representative images for visualizable entities. Such images generally have large percentages of pixels correspond to the main objects which visualize entities, and such objects are generally in the center of the images.

[4]The same WordNet vocabulary as in Table1 is used. This statistics is based on exact match and it should be much larger for partial match since 99.0% of ImageKB entities are phrases.

3. ENTITY AND IMAGE COLLECTION

We propose a data-driven approach to mine a vocabulary and candidate images efficiently from 2B images. The

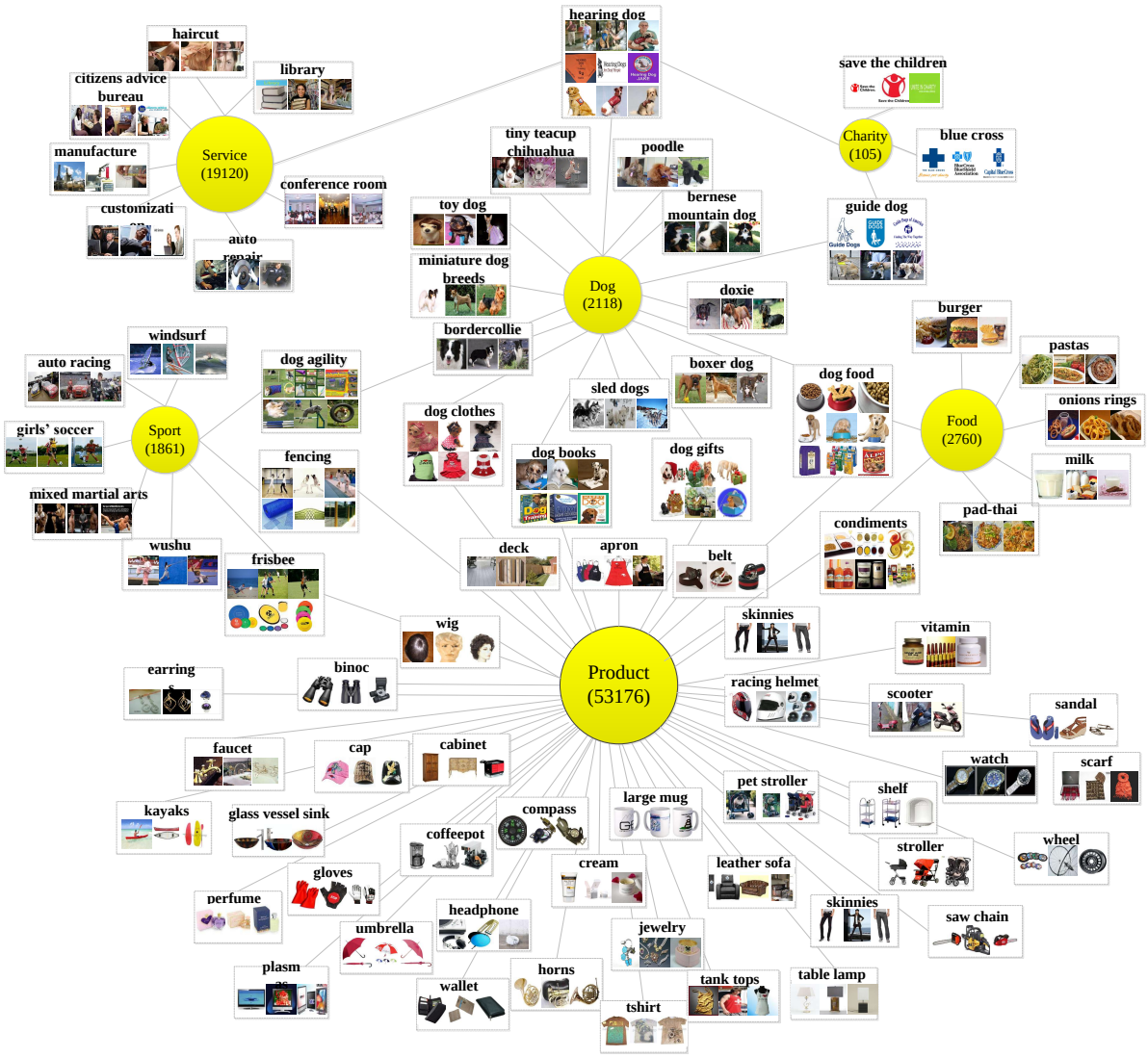

Figure 4: A subgraph of ImageKB on product and dog and three of their related categories. Circles and rectangles represent categories and entities (with ImageKB images) respectively. The digital number in a circle indicates the number of entities belonging to that category, e.g. there are 53,176 entities of product. ImageKB contains both leaf (e.g. bordercollie) and categorical entities (e.g. toy dog) and has a overlapping hierarchy. Some entities have unique categories, e.g. tiny teacup chihuahua, and some belong to multiple categories, e.g. hearing dog.

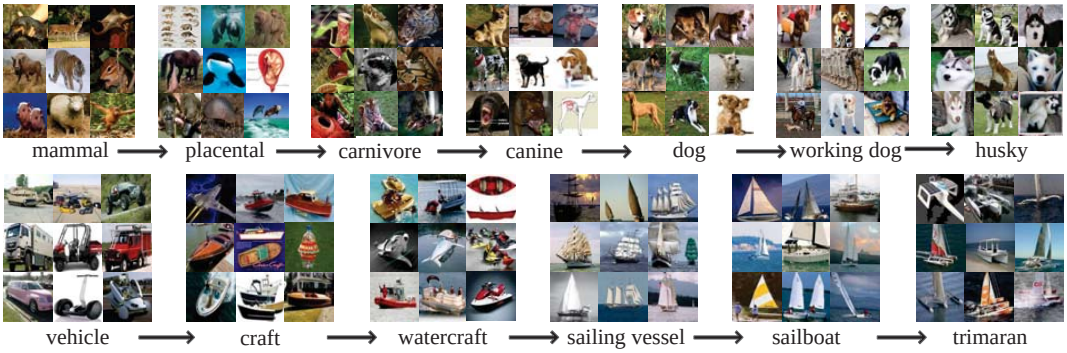

Figure 5: Though ImageKB is locally flat, we can also build a visualized branch hierarchy from ImageKB based on some ontology such as WordNet[7]. The entities and images all come from ImageKB.

Table 3: Examples of NeedleSeek[8] Output

Item	Category	Instances
puma	animal	jaguar, cougar, panther, ocelot, leopard
	brand	adidas, nike, reebok, timberland, gucci
java	language	perl, c++, php, python, c#, javascript
	country	sumatra, bali, borneo, sulawesi
TLC	celebrity	usher, toni braxton, mariah carey
	network	animal planet, discovery, mtv, cnn

process contains two steps, duplicate discovery and image annotation.

3.1 Duplicate Image Discovery

Our system was designed starting from the fact that there are many duplicate images on the web. On one hand, only one out of many duplicate images should be included in the database for compactness. On the other hand, rich tags of many duplicate images make it possible to accurately infer annotation information[16]. Therefore, it is important to automatically discover clusters of duplicate images from a large corpus of web images.

However, the task is very challenging since the input is 2B images. This problem cannot be solved by existing duplicate search/detection approaches[18, 19, 20] as every image would otherwise be used as a query with computation complexity $O(n^2)$, where n is the dataset size. Existing duplicate discovery solutions[21, 22], on the other hand, appear to require too high memory and time cost to scale up to billions of images.

We propose a novel duplicate discovery approach which is feasible on 2B images, containing three steps:

1. **Space partitioning**: We extract a global vector of color, texture and edge features for an image, and encode the descriptor into a binary signature with a PCA model[23, 24]. Images having equal signatures are assigned into the corresponding hash buckets. Thus, the 2B images are efficiently partitioned into multiple buckets so that image clustering is feasible within each bucket.

2. **Image clustering**: Pair-wise image matching is performed within a bucket based on their original global visual features. Images whose distances are smaller than a certain threshold are regarded as duplicate images. Accuracy is ensured in this step.

3. **Cluster merging**: An average image is computed from each cluster, and two clusters are merged into one if the distance between their average images are smaller than the threshold. This step improves recall.

With 20-bit signatures, 180.1 million duplicate image clusters are discovered in total, which correspond to 569.2M images. The average precision of images that are truly duplicate in a cluster is 98.37%, estimated based on 1,000 randomly sampled outputs. Our approach has very low computational and memory cost. The task on our 2B images was finished in 5 days on 10 servers, each having 16GB memories and running ten threads.

3.2 Image Annotation

We adopt the text mining step of the Arista approach[16, 17] to generate our entity vocabulary from the duplicate image clusters. Given a cluster, the approach takes as input the surrounding texts of each duplicate image and identify salient terms and phrases which are common among the texts. The unique annotations make up of the candidate vocabulary of visualizable entities.

In total, we found 155.2M candidate entities out of 120.70M duplicate image clusters. These are the input of the filtering and ranking step.

4. IMAGE FILTERING AND RANKING

Given the candidate entities and duplicate image clusters, we now work on identifying representative images for an entity. On one hand, if we can identify some example images of an entity, the representativeness of an image can be naturally measured against the example images. On the other hand, some entities are ambiguous, e.g. "apple" can either indicate a fruit apple or Apple Inc. Therefore, we define our problem as generating representative images for ⟨entity, category⟩ pairs, rather than for entities directly. We obtain the knowledge of ⟨entity, category⟩ pairs from a term-category look-up table mined from 0.5B webpages[8], and the features representing a category also leverages this look-up table.

Our process contains two steps: to identify relevant images for an ⟨entity, category⟩ pair, and to rank the relevant images. The top-ranked images are assumed as representative images for an ⟨entity, category⟩ pair.

4.1 The Term-Category Table

We use a term-category table (physically an ontology) to structurize the entities in ImageKB to obtain the knowledge of relationships among entities. This knowledge can be very useful. For example, if we know entity A never co-occurs with entity B, it is unnecessary for an object recognition model to differentiate A from B but only focus on A and its related entities. This may result in more accurate recognition models. Technically, a term-category table enables an efficient and scalable image filtering technique in categorizing an entity and generating the textual descriptors of a category, which will be explained in the next section.

We use the output of NeedleSeek [8] as our term-category table. NeedleSeek mines with a bottom-up strategy three types of knowledge from large-scale webpages, i.e. items, categories, and the mapping between them. A typical NeedleSeek look-up table is shown in Table 3. For example, NeedleSeek identifies two semantic categories for puma - animal and brand; each category has a list of instances which are analogies for puma. The look-up table is learnt with natural language processing and data mining techniques. For example, from a sentence "apple is a kind of fruit", NeedleSeek analyzes that "apple" is an item whereas "fruit" is a category for "apple". From another sentence "pear is a kind of fruit", NeedleSeek learns that "pear" is also an item and it is peer for "apple" w.r.t. the category "fruit".

NeedleSeek currently hosts about 12.83M terms and 0.19M categories mined from 0.5B webpages, much larger than WordNet[7] and Wikipedia, as suggested by the coverage on user queries in Table 1.

However, we should point out that there is still a big mismatch between NeedleSeek vocabulary and the vocabulary of image annotations. Based on an order-insensitive match-

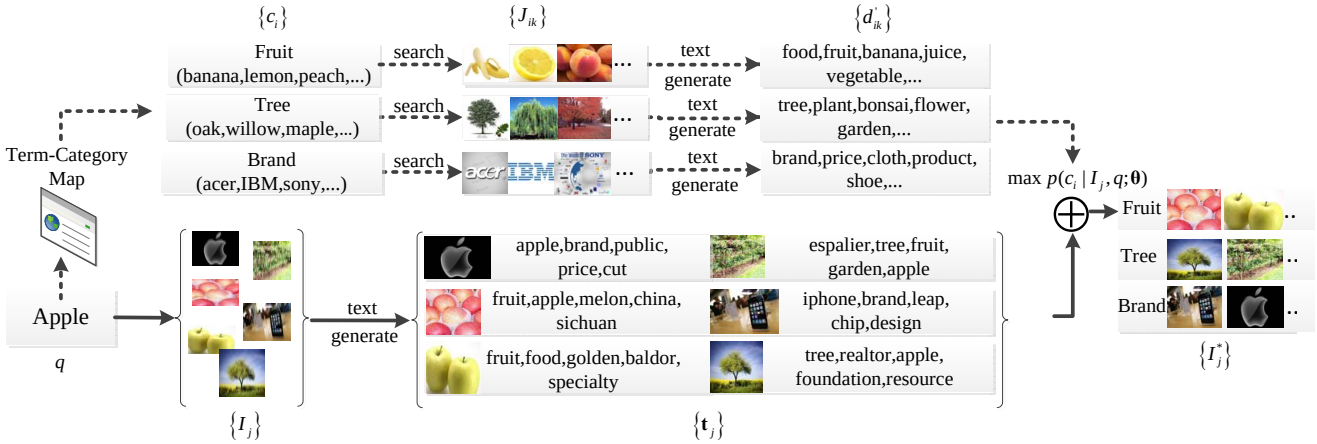

Figure 7: Basic idea of image filtering: an apple image is measured against the three categories of apple: fruit, tree, and brand. On one hand, definitive texts of a category are generated by first using 100 instances of the category for image search, and then find salient terms from the surrounding texts of duplicate images of the image search results. On the other hand, definitive texts of an image are also identified from surrounding texts of its duplicates.

ing[5] (e.g. "tiny teacup chihuahua" is assumed a match to "teacup chihuahua tiny" or "teacup tiny chihuahua", but not "teacup chihuahua" or "tiny chihuahua"), only 2.71M output 155.0M image annotations appear in NeedleSeek ontology, as shown in Table 5. We will investigate this problem in future work.

4.2 Feature Extraction

We represent both images and categories using *definitive texts* for semantic communication.

4.2.1 Category representation

Let $V = \{v_l\}_{l=1}^{N_V}$ be the vocabulary and v_l be an entry of V. $Q \subset V$ and $C \subset V$ be the query set and category set, respectively. We denote $q \in Q$ an entity and $\{c_i^q | c_i^q \in C, i = 1, \ldots, N_q\}$ its related categories, where c_i^q is the ith category and N_q denotes the total number of categories of q. For simplicity, we drop q in c_i^q.

We leverage a term-category look-up table to identify example images $S = \{J_{ik}\}$ for c_i, where J_{ik} is an image. As shown in Fig.7 the dotted line process, using c_i = fruit as an example, we first get a number of instances of fruit (i.e. banana, lemon, peach, etc.) from the look-up table. The assumption is that the semantics of a category is defined by all its instances, and can be approached by a large enough number of instances. We used 100 instances in our implementation. For each instance, we get top five image search results as the example images of c_i (five is used is to control the accuracy of image search results).

We then generate one document $d_{ik} = \{w_l | w_l \in V\}$ for each J_{ik}. Specifically, instead of simply defining d_{ik} as surrounding texts of J_{ik}, we aggregate all the surrounding texts of the duplicate images of J_{ik} because semantic-related terms have larger chance to be repeated among duplicate images than noisy terms[16, 17].

Although d_{ik} can be directly used for image filtering (see details in Section 4.3), we do some further work on feature selection to identify category-specific words $\mathbf{w}_i = \{w_{il} | w_{il} \in V, l = 1, \cdots, |d_{ik}|, i = 1, \cdots, N_q\}$ for c_i, and the words in d_{ik} which are out of the vocabulary $V_q = \bigcup_{i=1}^{N_q} \mathbf{w}_i$ are removed. We denote such cleaned d_{ik} as d'_{ik}, which assigns the definitive texts for category c_i.

To learn d'_{ik}, in the case of $N_q > 1$, we weight each word by the TF-CHI (i.e. term frequency-χ^2) weighting scheme, one of the best feature selection methods for text classification[25]. Categorical information is needed by TF-CHI, so that discriminative words can be selected for different categories. Table 4 gives the top 5 words that have the highest χ^2 values of 14 categories, corresponding to 5 entities. These words are very discriminative. In the case of $N_q = 1$, d'_{ik} contains the words whose term frequencies (TF) are above a threshold.

4.2.2 Image descriptors

Let $\{I_j | j = 1, \ldots, M_q\}$ be the candidate images of q where I_j is the jth candidate image of q and M_q is the total number of images of q. Let $\mathbf{t}_j$ be the feature vector of I_j.

We represent a candidate image with definitive texts, illustrated by the solid-line process in Fig.7. For an image I_j, once again the surrounding texts of its duplicate images are aggregated and only the words whose term frequencies (TF) are above a threshold are kept. An image is then represented by the TF-weighted vector on remained words, i.e. $\mathbf{t}_j = \langle tf(t_{j1}), tf(t_{j2}), \ldots, tf(t_{jm}) \rangle$ where $tf(t_{jl}) > \delta, l = 1, \ldots, m$ and δ is a threshold.

[5]We prefer this strict criterion to partial match for noise control.

Table 4: Topics of Categories

Entity	Category	Top 5 terms ranked by χ^2 values
Lincoln	president	president, United State, politics, American, history
	manufacturer	price, product, buy, manufacturer, brand
	county	county, map, county map, location, area
fox	animal	animal, wildlife, pet, specie, wild
	studio	studio, screenshot, game, play station, x-box
	channel	channel, logo, cable, network, media
	celebrity	celebrity, actress, Hollywood, gossip, sexy
mouse	animal	animal, wildlife, pet, specie, wild
	device	device, mobile, gadget, phone, electron
java	language	language, text, book, software, product
	country	map, country, travel, country map, geographic
lotus	flower	flower, rose, garden, florist, gift
	car	car, auto, motor, vehicle, wallpaper

4.3 Image Filtering

The image filtering problem is to identify

$$S^* = \{I_j | p(c_* | I_j, q; \Theta) \geq \xi, 1j = 1, \cdots, M_q\} \quad (1)$$

where $c_* = \arg\max_{c_i} p(c_i | I_j, q; \Theta)$ is the most probable category that generates I_j. Θ is the parameter set. $p(c_i | I_j, q; \Theta)$ is defined as:

$$p(c_i | I_j, q) = \begin{cases} \frac{1}{L}\sigma(f_i(x_j^I)) & N_q > 1 \\ \frac{1}{Z}\cos(x_i^c, x_j^I) & N_q = 1 \end{cases} \quad (2)$$

where x_i^c and x_j^I indicate the features of c_i and I_j, respectively. L and Z are normalization factors. $\sigma(\cdot)$ is the sigmoid function whereas $f_i(\cdot)$ defines the cost function of the classifier of category c_i. For $N_q > 1$, we learn linear one-against-all SVM classifiers[26] to define $f_i(\cdot)$. SVM outputs are mapped to probabilities using the sigmoid function[27] and images whose probabilities are less than a threshold are removed as noises. For $N_q = 1$, we measure $p(c_i | I_j, q; \Theta)$ by the cosine similarity between image I_j and c_i.

Note that for each q, the SVM classifiers are learnt on only related c_i of q. This is the key for effectiveness with simple features as it is unnecessary to learn effective classifiers on millions of categories when we know which categories an entity belongs to.

4.4 Image Ranking

The last step is to rank S^* and the top-ranked images are assumed as representative to a $\langle q, c_i \rangle$ pair. We define a simple yet effective scoring function to measure the representativeness of an image, which infers the authority of an image from its nearest neighbors.

Intuitively, a representative image should first be relevant to the category to which it belongs. We define the relevance score r as the cosine similarity between the textual features of an image and a category. Meanwhile, we measure the confidence g_{ij} of I_j in representing $\langle q, c_i \rangle$ by Eq.3 based on its nearest neighbors. The motivation is that the more K-nearest neighbors of I_j are representative to $\langle q, c_i \rangle$, the more confident that I_j should also represent $\langle q, c_i \rangle$. In our implementation, $K = 5$. The confidence $g_i j$ is defined as

$$g_{ij} = \frac{\sum_{s(I_j, nn_{ik})=1} \cos(I_j, nn_{ik}) \times r_{nn_{ik}}}{\sum_{k=1}^{K} \cos(I_j, nn_{ik}) \times r_{nn_{ik}}}, \quad (3)$$

where nn_{ik} is the kth nearest neighbor of image I_j in category c_i. $\cos(I_j, nn_{ik})$ is the cosine similarity between image I_j and its kth neighbor. $r_{nn_{ik}}$ is the relevance score of nn_{ik} to the semantics of $\langle q, c_i \rangle$. $s(I_j, nn_{ik})$ is defined by

$$s(I_j, nn_{ik}) = \begin{cases} 1, & I_j^k, nn_{ik} \in c_i \\ -1, & otherwise \end{cases} \quad (4)$$

Finally, I_j are ranked by their representativeness scores $score_{ij}$ defined as

$$score_{ij} = \frac{\sum_{k=1}^{K} s(I_j, nn_{ik}) \times \cos(I_j, nn_{ik}) \times r_{nn_{ik}} \times g_{ij}}{K}. \quad (5)$$

The motivation is that if an image has more nearest neighbors that share the same semantics and have high relevance scores, the better chance this image is representative for this semantics. Contrarily, the more nearest neighbors have low relevance scores or have diverse semantics, the less possible that this image is a representative one.

Table 5: Intermediate Results

#duplicate image clusters	180,145,940
#image clusters being tagged	120,697,779
#unique tags	155,024,386
vocabulary size after entity filtering	2,705,075
#final entities	518,072

5. EXPERIMENTS

We conducted thorough experiments to evaluate the effectiveness of the entire process as well as its major components. We present our observations on mining from 2B images in two aspects: dataset properties and performance.

5.1 Dataset Properties

Table 2 summarizes the ImageKB to date, which is the largest in scale compared to existing datasets in the literature[9, 10, 11, 5, 6]. Some statistics of ImageKB have been analyzed in Section 2.3.

It is worthwhile to know to what extent we have achieved. Table 5 provides the intermediate statistics during processing the 2B images. From Table 5, we can see that there are in total 180.15M duplicate image clusters discovered from the 2B images, and after image annotation, the number is reduced to 120.70M, which means $33.0\% = (180.15M - 120.70M)/180.15M$ clusters are not annotated. There are two possible reasons: 1) a cluster is small which contains too few images to be annotated by Arista[16, 17]. Note Arista is not able to annotate images having less than three duplicates, and 2) the surrounding texts are too noisy for Arista to identify semantic words.

The 120.70M annotated clusters contains 155.02M unique annotations. However, only 2.71M out of the annotations are common with the 264.17M unique user queries in the 6-month query log and the 12.83M NeedleSeek[8] items. The ratio is only $2.71M/155.02M = 1.75\%$. This is because Arista tends to generate long phrases and these phrases may contain some noisy terms, e.g. "tom cruise family in town", "tom cruise 101". Though "tom cruise" exists in both the query log and NeedleSeek vocabulary, "tom cruise 101" is not. Two future works can be done here: 1) to perform partial match between items[6], and 2) to improve the Arista tagging technique towards less noise.

On the other hand, from Table 1 we can see that ImageKB-query log overlap is 13.26M, which means $79.56\% = (13.26M - 2.71M)/13.26M$ of the 13.26M tags are further removed by NeedleSeek. Recall ImageKB is built by mapping its entities and images onto NeedleSeek ontology. A future research direction is to propose algorithms to identify representative images for entities not covered by an ontology.

There again is a big gap between the vocabulary after entity filtering (i.e. 2.71M) and the final number of entities in ImageKB (i.e. 0.52M). This is because many entities have no representative images available after our image filtering and ranking step (see Section 4) due to the strict parameter settings for noise control. In the future, we will work on more sophisticated models based on the current ImageKB to collect more images.

[6]To determine when a partial match is safe and when it is not is an open research problem, given the scale and the diversity of data. For example, though "dog" partially matches "dog food", they have totally different semantics.

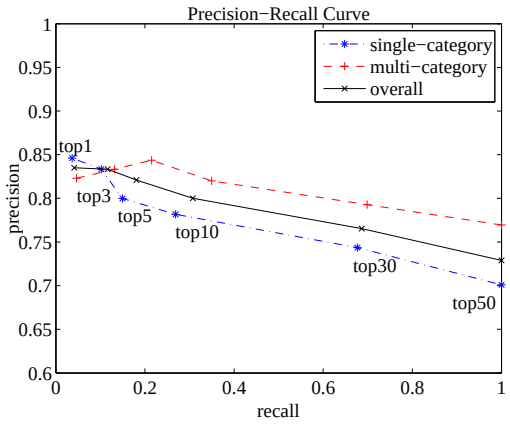

Figure 8: Average Precision Recall Curves.

Table 6: Entities for Evaluation - Examples

(a) Ten Examples of Meaningful (Entity, Category) Pairs

(brute force, game), (drawing the lines, song), (tortes, food), (long jumping, sport),(iphone 3g s, device), (the comic, book), (batman - the dark knight, movie), (nissan pickups, vehicle), (empire state building, building), (capybaras, animal)

(b) Ten Examples of Removed (Entity, Category) Pairs

(money shot, game), (beautiful city, song), (boks, sport), (soliloquy, device), (non food, food), (milf diaries, book), (movie 2, movie), (beautiful, brand), (beautiful, song), (rigby, city)

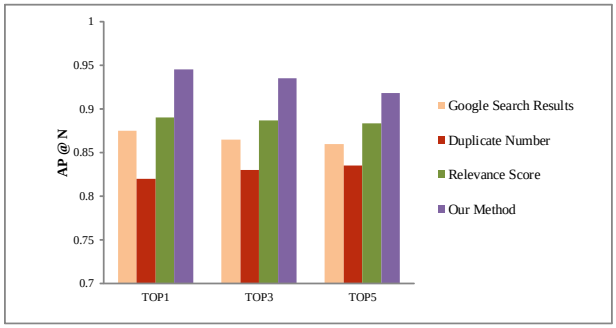

Figure 9: AP@N on single-category entities

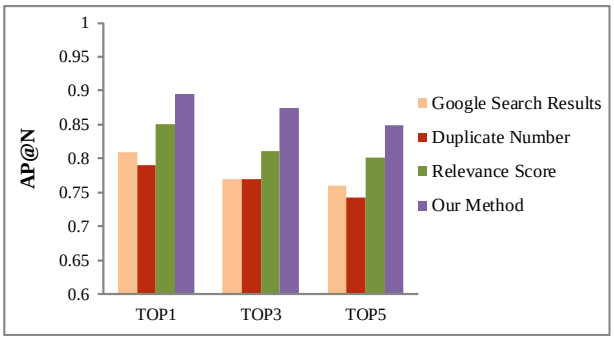

Figure 10: AP@N of multi-category entities

5.2 Performance

5.2.1 *Performance within the 2B images*

We randomly selected 150 entities and manually labeled them to evaluate the precision and recall performance on their top 50 images. Precision ($AP@N$) is defined as the percentage of correct images in top N results, whereas recall ($AR@N$) is defined as the percentage of discovered correct ones in top N results:

$$AP@N = \frac{L(N)}{N}, \qquad AR@N = \frac{L(N)}{M} \tag{6}$$

where $L(N)$ denotes the number of correct images in top N results, and M denotes the ground truth number of correct ones in the top N results.

For each ⟨entity, category⟩ pair, we asked three labelers to label the corresponding ImageKB images independently. The final correctness of an image is determined by majority voting among its three labelers. If it is difficult for the labelers to understand the semantics of a ⟨entity, category⟩ pair even after browsing its Google and Bing's image search results, the pair was regarded as noisy and was removed from the evaluation set[7]. Note that the existence of noisy ⟨entity, category⟩ pairs is reasonable since NeedleSeek[8] itself is an automatic approach. Table 6(a) and Table 6(b) shows ten examples of meaningful and noisy ⟨entity, category⟩ pairs respectively.

Figure 8 illustrates the precision-recall curves on top 50 results of single-category entities (blue dash-dot lines), multi-category entities (red dotted line), and overall results (black real line) respectively. The performance is quite satisfying considering the simple textual features we are using. The AP@1, AP@10, and AP@50 of the overall performance is about 0.84, 0.80 and 0.74 respectively, whereas the overall recall is about 0.05, 0.31 and 1. $AP@50 = 0.74$ when $AR@50 = 1$ means more than 37 out of 50 images on average are correct.

It is interesting that ImageKB obtained better performance on multi-category entities. This may be because a better image filtering model is used in the multi-category case. Recall that simple cosine similarity measure and SVM classifiers are used in the single- and multi-category cases respectively (see Section 4.2).

5.2.2 *Performance outside the 2B images*

To better understand the performance of our approach, we evaluate it with images outside of the 2B database.

Specifically, given an ⟨entity, category⟩ pair, we collect top 50 Google returned images as candidate images, from which representative images are selected. For each image, we obtain its duplicate images from the 2B database, and represent an image with high-quality texts using the same technique as presented in Section 4.2.2. Meanwhile, we learn the descriptive documents of categories using the same method as described in Section 4.2.1; the only difference is that the top 5 images used for document generation are from Google rather than from our own text-based image search engine built upon the 2B images.

We compared our method to three baseline methods:

Search Results (SR) - the top Google image search results are assumed as representative images.

Duplicate Number (DN) - the images which have the largest number of duplicates are representative ones.

[7]We found that about 30% randomly selected evaluation pairs were removed by the labelers. This implies that about 30% nodes of the current ImageKB are unreliable (i.e. have low accuracy) and can be removed. Therefore, it is worth exploring a confidence score on ImageKB nodes.

Relevance Score (RS) - It differs to our approach only in that the relevance score is used to rank images.

Fig.9 and Fig.10 illustrate the overall $AP@N$ performances of our method and the baselines on single-category entities and multi-category entities respectively, of which $N = 1, 3, 5$. It can be seen that our method greatly surpasses the baselines on both types of entities. Specifically, in the top@1 case, our method achieved a precision of 94% on single-category queries and 90% on mutli-category ones.

Moreover, DN performs even worse than SR, which is an unexpected result. People may think that the more duplicates an image has, the larger chance that it is representative to a query, since the number of duplicates indicates the popularity. Our evaluation suggests that such an assumption may not always be true, e.g., funny images and funny comics are very popular but may not be representative to a certain entity. Therefore, an image filtering step is indispensable to ensure system performance.

On the other hand, comparing these Fig.9 and Fig.10 to Fig.8, we can see that using cleaner candidate images (here Google image search results), the accuracy of output images is about 10% higher than that of using images collected from auto-annotation results. This suggests that the cleaner the candidate images, the higher the accuracy of output. This suggests that to reduce noisy candidate images can improve the accuracy of ImageKB. We will investigate this point in our future work.

6. CONCLUSION

This paper reports our first-stage achievement on building an image knowledge base of representative images, ImageKB. ImageKB is automatically and efficiently generated from 2B web images with three main procedures: 1) discovering duplicate image clusters and annotating them to get candidate entities and their images, 2) filtering images by matching the entities and images to an ontology which contains millions of nodes, and 3) re-ranking images by their authority scores inferred from nearest-neighbor graphs. Containing more diverse images and deeper semantics than existing image database, ImageKB can be broadly used in image search, retrieval and computer vision.

7. ACKNOWLEDGEMENT

We highly appreciate Yuan Li's contribution to some initialization work and part of the code.

8. REFERENCES

[1] Smith, J., Chang, S.F.: An image and video search engine for the world wide web (1996) In: SPIE.

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

[3] Garcia, S., Williams, H.E., Cannane, A.: Access-ordered indexes (2004) In: ACSC.

[4] Zobel, J., Moffat, A.: Inverted files for text search engines. ACM Computing Surveys **38** (2006)

[5] Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: a large dataset for non-parametric object and scene recognition. (In: IEEE T-PAMI)

[6] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database (2009) In: CVPR.

[7] Fellbaum, C.: Wordnet: An electronic lexical database (1998) Bradford Books.

[8] Shi, S., Zhang, H., Yuan, X., Wen, J.: Corpus-based semantic class mining: distributional vs. pattern-based approaches (2010) In: ICCL.

[9] Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories (2004) In: CVPR Workshop on Generative-Model Based Vision.

[10] Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report **7694** (2007)

[11] Russell, B., Torralba, A., Murphy, K., Freeman, W.: Labelme: a database and web-based tool for image annotation. In: IJCV **77** (2008) 157–173

[12] Deselaers, T., Ferrari, V.: Visual and semantic similarity in imagenet (2011) In: CVPR.

[13] Weston, J., Bengio, S., Usunier, N.: Large scale image annotation: Learning to rank with joint word-image embeddings (2010) In: ECCV.

[14] Good, J.: How many photos have ever been taken? (2011) http://blog.1000memories.com/94-number-of-photos-ever-taken-digital-and-analog-in-shoebox.

[15] Wang, X.J., Zhang, L., Jing, F., Ma, W.Y.: Lei zhang, feng jing, wei-ying ma, annosearch: Image auto-annotation by search (2006) In: CVPR.

[16] Wang, X.J., Zhang, L., Liu, M., Li, Y., Ma, W.Y.: Arista - image search to annotation on billions of web photos (2010) In: CVPR.

[17] Wang, X.J., Zhang, L., Ma, W.Y.: Duplicate search-based image annotation using web-scale data. Proceedings of IEEE (2012)

[18] Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos (2003) In Proc. ICCV.

[19] Chum, O., Philbin, J., Zisserman, A.: Near duplicate image detection: min-hash and tf-idf weighting (2008) In Proc. BMVC.

[20] Ke, Y., Sukthankar, R., Huston, L.: Efficient near-duplicate detection and sub-image retrieval (2004) In: ACM Multimedia.

[21] Chum, O., Matas, J.: Large scale discovery of spatilly related images. IEEE T-PAMI (2010)

[22] Lee, D., Ke, Q., Isard, M.: Partition min-hash for partial duplicate image discovery (2010) In: ECCV.

[23] Pearson, K.: On lines and planes of closest fit to systems of points in space. Philosophical Magazine **2** (1901) 559–572

[24] Abdi, H., Williams, L.: Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics **2** (2010) 433–459

[25] Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization (1997) In: ICML.

[26] Chang, C., Lin, C.: Libsvm: A library for support vector machines (2012) http://www.csie.ntu.edu.tw/cjlin/libsvm.

[27] Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers. MIT Press (1999)