What Are the High-Level Concepts with Small Semantic Gaps?

Yijuan Lu[1], Lei Zhang[2], Qi Tian[1], Wei-Ying Ma[2]
[1]University of Texas at San Antonio
One UTSA Circle, San Antonio
Texas, 78249-USA
{lyijuan, qitian}@cs.utsa.edu

[2]Microsoft Research Asia
No. 49, Zhichun Road, Haidian District
Beijing 10080, P. R. China
{leizhang, wyma}Microsoft.com

Abstract

Concept-based multimedia search has become more and more popular in Multimedia Information Retrieval (MIR). However, which semantic concepts should be used for data collection and model construction is still an open question. Currently, there is very little research found on automatically choosing multimedia concepts with small semantic gaps. In this paper, we propose a novel framework to develop a lexicon of high-level concepts with small semantic gaps (LCSS) from a large-scale web image dataset. By defining a confidence map and content-context similarity matrix, images with small semantic gaps are selected and clustered. The final concept lexicon is mined from the surrounding descriptions (titles, categories and comments) of these images. This lexicon offers a set of high-level concepts with small semantic gaps, which is very helpful for people to focus for data collection, annotation and modeling. It also shows a promising application potential for image annotation refinement and rejection. The experimental results demonstrate the validity of the developed concepts lexicon.

1. Introduction

Recent years have witnessed a fast development of Multimedia Information Retrieval (MIR). Despite continuous efforts in exploring new MIR techniques, the semantic gap between the expressing power of low-level features and high-level semantic concepts is still a fundamental barrier. In order to reduce the semantic gap, a promising paradigm of concept-based multimedia search has been introduced into many practical search systems in the past few years. This paradigm focuses on modeling high-level semantic concepts, either by object recognition or image annotation. Among various approaches, the first step is to select a lexicon that is relatively easy for computers to understand, and then to collect training data to learn the concepts.

However, the problem of lexicon selection is usually either simplified by manual selection or totally ignored in most previous works. For example, researchers working on object classification and recognition manually defined a number of datasets, including UIUC [1], Caltech 101[2], Caltech 256 [3], and PASCAL [4]. When choosing concepts to construct these datasets, they implicitly favored those relatively "easy" concepts, although it is still very challenging to model them. Other researchers working on image annotation either simply use all the keywords associated with training images, including ALIPR[5], SML [6], or don't impose any limitation to the annotation vocabulary such as ESP [7], LabelMe [8], and AnnoSearch [9]. These approaches actually ignore the differences among keywords in terms of semantic gap.

There is no doubt these efforts make their unique contributions to the standardization of concept corpus thus letting the multimedia community focus ongoing research on a well-defined set of semantics. However, we argue that semantic gaps are actually not uniform in a low level feature space and it is inappropriate to ignore the semantic gap differences. For example, it is well acknowledged that modeling "Europe" is more challenging than modeling "sunset" due to the lack of an effective visual feature that can represent the concept of "Europe". Also, researchers usually choose color features to model concepts like "sunset", and choose local features to model concepts like "building".

Concepts with smaller semantic gaps are likely to be better modeled and retrieved than concepts with larger ones. But in current literature, very little research is found on quantitative analysis of semantic gap. *What are the well-defined semantic concepts for learning and how to automatically find them* are still open problems. This highlights a critical requirement for establishing an efficient way to "measure" the semantic gap thus finding those high-level concepts with small semantic gaps, which should be given high priority for data collection, modeling, and training.

Motivated by this, this paper focuses on two key problems: what are the high-level concepts with small semantic gaps and how to identify them? In other words, what semantic concepts should we focus on first to assure that they can be well modeled and easily annotated? We answer these questions by proposing a novel framework to automatically construct a concept lexicon from a large web-scale image dataset, which contains over 2.4 million

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

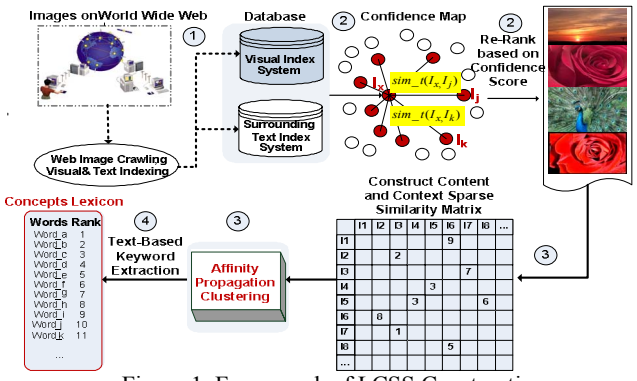


Figure 1: Framework of LCSS Construction

images collected from several online photo forum websites.

Web images are usually associated with rich textual features, such as filename, title, alt text, and surrounding text [10]. These textual features are much closer to the semantics of the images than visual features. Especially when users upload their photos online, the input titles and comments that are assigned by the users are actually good semantic descriptions of the images. Therefore, in this paper, we particularly focus on the collected web images and fully utilize this useful information to identify images with small semantic gaps by the proposed confidence value. Then the top N images with the smallest semantic gaps are clustered and the key concepts learned from their surrounding texts are output as the produced concepts.

To our best knowledge, it is the first attempt that quantitatively and automatically identifies high-level concepts with small semantic gaps from a huge repository of well annotated photographs on the Web. Compared with the limited number of manually selected concepts, the proposed approach is potentially capable of constructing a well-defined lexicon customized for a given feature space, and is of great help for data collection and concept modeling.

In our design, we request two properties for the desired concept lexicon. (i) The words (concepts) in the lexicon should have high occurrence frequency within the descriptions of real-world images, which makes them commonly used concepts. (ii) The chosen concepts are expected to be visually and semantically consistent, that is, the images of these concepts have smaller semantic gaps, which make them moderately easy to be modeled for retrieval and annotation.

The contributions of this work can be highlighted as follows:

1) We quantitatively study and formulate the semantic gap problem and propose a novel framework including a definition of confidence value and an algorithm for dominant concept identification to automatically select visually and semantically consistent concepts.

2) The constructed lexicon shows its promising application potential for concept detection, automatic annotation, and multimedia information retrieval. As the chosen concepts are ranked based on their semantic gaps, researchers can either focus on modeling concepts which are visually and semantically more consistent, or concentrate on designing rejection strategies to reject those tough concepts with low confidence.

3) Although this work only studies lexicon construction in one feature space, the proposed framework is also helpful for feature space selection for any given concept. This will explicitly guide the research in concept modeling and provide a possibility for multimodality modeling.

The rest of the paper is organized as follows: Section 2 introduces the related work. Section 3 presents the lexicon construction procedure. Section 4 gives the comprehensive experimental results, and the conclusions and future work are given in Section 5.

2. Related Works

Lexicon selection and data collection are essential elements of concept-based image retrieval. Publicly available image databases, such as UIUC [1], Caltech-101 [2], Caltech-256 [3], and PASCAL [4], contain many manually selected concepts for category-level recognition. Recently, web-based annotation tools (ESP [7] and LabelMe [8]) provide a new way of building large annotated database by relying on the collaborative effort of a large population of users [11]. By playing games, players enter labels describing the content of images, from which, a lexicon can be collected. In 2006, MediaMill challenge concept data (101 terms) [12] and Large-Scale Concept Ontology for Multimedia (LSCOM) [13] containing about 1,000 concepts were proposed, both of which include a manually annotated concept lexicon established on broadcast news video from the TRECVID benchmark. The LSCOM was designed to satisfy multiple criteria of utility, coverage, feasibility, and observability.

Unfortunately, all the lexica described ignore the differences of semantic gaps among concepts and no automatic selection is executed.

3. Lexicon Construction

In this paper, we propose a confidence map to "measure" the semantic gap. After selecting images with small semantic gaps from a large-scale web-based database, concepts are mined from the descriptions of the clustered images.

The framework of the Lexicon of High-Level Concepts with Small Semantic Gaps (LCSS) construction procedure is shown in Figure 1. It contains four stages: (1) data collection, (2) confidence map construction, (3) affinity propagation clustering, and (4) text-based keyword extraction labeled by ①-④, respectively.

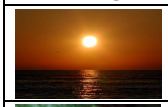
Image	Title	Descriptions
	Sea sunset	Sunset at the sea
	Red Rose	A rose in my garden taken June 8th 2002 (My other hobby is rose gardening)…
	The Falls	This is a waterfall that is about 3 miles from my house. It's called The Falls…

Figure 2: Example images and surrounding descriptions

3.1 Data Collection

About 2.4 million web images were collected from several online photo forum sites including Photosig.com, Photo.net, *etc.* The reason we chose these forum sites was that their photos have very high quality and rich textual information such as title and photographer's comments. As shown in Figure 2, these descriptions cover the content of the corresponding photos to a certain degree.

Semantic gap really depends on the low-level features. In this paper, a 64-dimensional global visual feature vector [14] is extracted for all 2.4 million images, which contains three different kinds of color features: 6 dimensional color moments in LUV color space, 44 dimensional banded auto-color correlogram in HSV color space, and 14 dimensional color texture moments.

3.2 Visual-Textual Confidence Map

According to the second property of our concept lexicon, the images belonging to these selected concepts must have small semantic gaps. Very little research in the current literature can be found on how to analyze the semantic gap. Fortunately, semantic information is available for the web images from their rich context features, such as title, category, and photographers' comments. This input information actually describes images' semantics when users name and describe them. In this paper, we utilize the context information and define a novel Nearest Neighbor Confidence Score (NNCS) to evaluate the semantic gap between visual (content) and textual (context) features.

3.2.1 Nearest Neighbor Confidence Score

Viewing each image as a *K*-NN classifier, for a particular image I_x, we obtain its K nearest neighbors based on its visual feature. Assuming I_x and one of its neighbors I_i both have surrounding texts, we can measure their textual description's cosine similarity $sim\_text(I_x, I_i)$ using their textual features. Then the Nearest Neighbor Confidence Score of image I_x can be defined as

$$NNCS(I_x) = \frac{1}{K}\sum_{i=1}^{K} sim\_text(I_x, I_i) \qquad (1)$$

where I_i, $i = 1, \cdots, K$ are the K nearest neighbors of I_x in the visual feature space. Obviously, an image's *NNCS* can be interpreted as the coherence degree in both visual and textual spaces.

By definition, semantic gap is the difference between two descriptions by low-level features and high-level concepts. The basic idea of the proposed NNCS to measure the semantic gap is using the low-level content features of image I_x to search most visually similar images first. Next, their contextual (semantic) similarities are calculated. If they share common contextual information, we can conclude these visually similar images also have very similar semantics thus called content and context similar. This consistency shows that the low-level features of image I_x can express its semantic information well. Hence, the higher the *NNCS* value is, the smaller the semantic gap would be.

In our paper, we calculate *NNCS* Score with $K = 500$ [1] for all 2.4 million images and construct a large visual-textual confidence map. From this map, we can select candidate images with high *NNCS* for later concept exploration and lexicon construction. The most simple way[2] is to rank all images by their *NNCS* value and use a threshold to select the top N images. In our implementation, 36231 top images are selected due to its relatively large size and memory concern for the Affinity Propagation clustering algorithm described in Section 3.2.2.

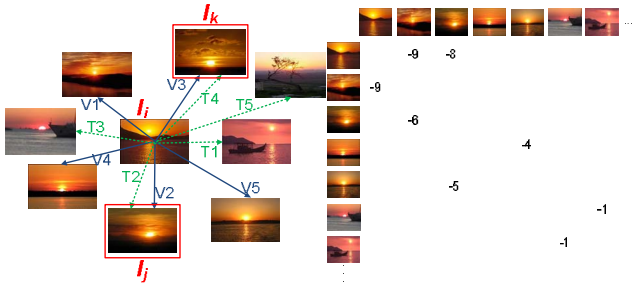
3.2.2 Clustering Using Affinity Propagation

After those candidate images with small semantic gaps are selected, the next step is to cluster these images and extract corresponding concepts information. We use a very recently proposed affinity propagation method [15] for clustering because it is fast for large scale data set and requires no *prior* information (e.g., number of clusters).

Different from traditional clustering methods, affinity propagation does not need specify and fix the number of exemplars (representative centers). It starts with the construction of a similarity matrix. By viewing each data point as a node in a network, this method recursively transmits real-valued messages along edges of the network until a good set of exemplars and corresponding clusters emerge. Affinity propagation has been successfully used

[1] K is chosen as a trade-off of both image coverage and computational complexity.

[2] We have tested with other selection methods, but simple thresholding gives quite reasonable candidate set and is very fast for implementation for large scale image dataset.



(a) (b)

Figure 3: (a) Content-Context *K*-Nearest Neighbor and (b) Content-Context Similarity Matrix

in face image clustering, genes detection, and sentence identification, *etc* [15].

In order to cluster content and context similar images together, we define and construct a content-context similarity matrix, based on content-context *K*-nearest neighbor (*KNN*-C2). Intuitively, image I_j is *K*NN-C2 of image I_i only if I_j is both visually and textually nearest neighbors of image I_i. An illustration example is given in Figure 3. I_i has K ($K=5$) visually and textually nearest neighbors v_1, v_2, v_3, v_4, v_5 and t_1, t_2, t_3, t_4, t_5. In this example, only image I_j and I_k are called *KNN*-C2 of image I_i since they are both visual and textual nearest neighbors of image I_i. The content-context *K*-nearest neighbors of image I_i can be represented by $kNN-C2(I_i)=\{I_j, I_k\}$.

Based on the *K*NN-C2, we construct a 36231×36231 content-context similarity matrix (CCSM) P as follows (Figure 3):

$$P_{ij}=\begin{cases} sim(I_i,I_j) & for\ I_j \in KNN-C2(I_i) \\ -\infty & for\ I_j \notin KNN-C2(I_i) \end{cases} \quad (2)$$

Where

$$\begin{aligned} sim(I_i,I_j) &= (1-\lambda)\cdot sim\_visual(I_i,I_j)+\lambda\cdot sim\_text(I_i,I_j) \\ &= -(1-\lambda)\cdot dist\_visual(I_i,I_j)-\lambda\cdot dist\_text(I_i,I_j) \end{aligned} \quad (3)$$

This CCSM matrix describes the similarity of visual content and textual context between any two images I_i and I_j, $i,j=1,2,\cdots,36231$. When I_j is *K*NN-C2 of I_i, P_{ij} is set to their content and context similarity, which equals to the summation of negative Euclidean distance of their visual features and cosine distance of textual features. Otherwise, P_{ij} is set to $-\infty$.

P_{ij} indicates how well an image I_i is suited to be the exemplar for image I_j. For all images, we assume that they are equally considered to be exemplars. Hence the preferences [15] are set to a common value - the median of P_{ij}. It should be noted that P_{ij} is not necessarily equal to P_{ji}. Fortunately, affinity propagation can be applied to non-symmetric similarity matrix.

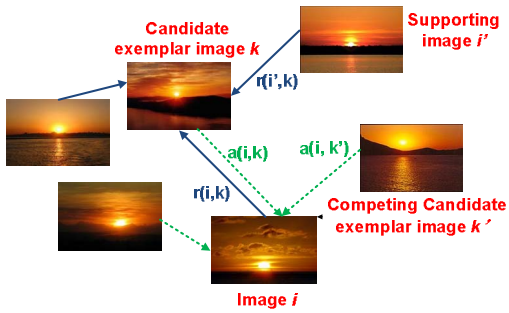


Figure 4: Message passing between images

Affinity propagation is a message-passing algorithm. Two kinds of messages: "responsibility" $r(i,k)$ and "availability" $a(i,k)$ are exchanged between images i and k.

Shown in Figure 4, the "responsibility" $r(i,k)$, sent from image i to candidate exemplar image k, reflects the accumulated evidence for how well-suited image k is to be the exemplar for image i, taking into account other potential exemplars. The "availability" $a(i,k)$, sent from candidate exemplar image k to point i, reflects the evidence for how appropriate for image i to choose point k as its exemplar, considering the support from other images that image k should be an exemplar.

At the beginning, the availabilities are initialized to zero: $a(i,k)=0$. Then, the responsibilities and availabilities are updated iteratively using the following two rules, which let all candidate exemplars compete for ownership of an image and gather evidence from images to support each candidate exemplar.

$$r(i,k) \leftarrow s(i,k)-\max_{k's.t.k'\neq k}\{a(i,k')+s(i,k')\} \quad (4)$$

$$a(i,k) \leftarrow \begin{cases} \min\left\{0, r(k,k)+\sum_{i's.t.i'\notin\{i,k\}}\max\{0,r(i',k)\}\right\} & i\neq k \\ \sum_{i's.t.i'\neq k}\max\{0,r(i',k)\} & i=k \end{cases} \quad (5)$$

After a fixed number of iterations, a good set of exemplars and corresponding clusters emerges. For image i, the image k that maximizes $a(i,k)+r(i,k)$ will be identified as its exemplar. If $k=i$, image i itself is an exemplar. The corresponding clusters are constructed by connecting each image to the exemplar that best represents it.

3.3 Text-based Keyword Extraction

After candidate images are well clustered, a text-based keyword extraction (TBKE) is proposed to extract keyword (concept) information from these clusters.

Given a cluster C_i in the cluster pool C, the text-based keyword extraction is to find the most representative

keywords by ranking all related keywords in this cluster. The related keywords are the ones that appear in the title or surrounding descriptions of images belonging to C_i. To be specific, the set of the related keywords of cluster C_i is denoted as W_i. And the relevance score of a keyword k_j to cluster C_i is denoted as $Score\_r(k_j, C_i)$.

Many different strategies could be applied to calculate $Score\_r(k_j, C_i)$. In paper [10], they show that an *if-ikf* strategy (image frequency-inverse keyword frequency) performs well when it finds keywords from surrounding texts of an image. Enlightened by the *if-ikf* strategy, we extend this strategy from image to cluster, defined as follows:

$$Score\_r(k_j, C_i) = \begin{cases} \dfrac{occurence(k_j, C_i)}{\ln(|W_i|+1)} & otherwise \\ 0 & W_i = \Phi \text{ or } k_j \notin W_i \end{cases} \quad (6)$$

where $occurence(k_j, C_i)$ denotes the number of keyword k_j in title or descriptions of images belong to cluster C_i.

Similarly, for the whole cluster pool C, we denote W as the combination of all W_i, *i.e.* $W = \bigcup_i W_i$. For each keyword k_j in W, its relevance score to the whole cluster pool C can be denoted as $Score(k_j)$. It is the summation of the relevance scores of k_j to each $C_j, C_i \in C$. $Score(k_j) = \sum_{C_i \in C} Score(k_j, C_i)$. The assumption is that if the k_j is a representative word for many clusters, it would be also an important keyword for the whole cluster pool.

Once the relevance score $Score(k_j)$ of each keyword k_j in W is obtained, we can select the top ones to make the final concepts lexicon. Because of the limited space, we only list the top 50 keywords in Table 1 grouped in five categories: scene, object, color, season, and others. The full concept lexicon can be downloaded from (left blank for blind review).

Table 1. Top 50 keywords in the LCSS lexicon

Category	Concepts
Scene/Landscape	sunset , sky, beach, garden, lake, sunflow, water , firework, cloud, moon, sunrise, mountain, city, river, snow, rain, home, island
Object	flower, rose, butterfly, tree, bee, candle, bridge, leaf, eye, tulip, orchid, house, peacock, window, glass, bird, rock
Color	blue, red, yellow, green, pink, purple, orange, dark, golden
Season	fall, spring, summer, autumn
Others	small, wild

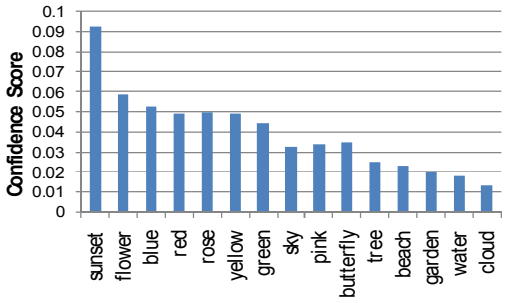


Figure 5: Distribution of average confidence value. *x*-axis represents top 15 keywords (from left to right) in LCSS.

4. Experiments

It is not a trivial task to evaluate LCSS. In this paper, we design two experiments to evaluate the validity of this list. First, we construct a confidence map for some concepts selected from the LCSS and compare their average confidence score. Secondly, we apply this developed LCSS list on image annotation refinement. The superior performance demonstrated that this concept lexicon provides a reliable and effective list of concepts with small semantic gaps.

4.1 Confidence Map

One way to evaluate the LCSS is to calculate the confidence score of images labeled with the keywords in the list. Intuitively, the images labeled with top keywords should have higher confidence value than images labeled with lower ranked keywords.

Hence, in our first experiment, we select top 15 keywords from the LCSS list. For each keyword *w*, we randomly selected 500 titled photos with this keyword from 2.4 million web images database. It is our assumption that the image is labeled with the keyword if the keyword appears in its title, then we calculate the average nearest neighbor confidence score of photos labeled with the same keyword.

The distribution of the average confidence score for each keyword is shown in Figure 5. It can be seen that the confidence value decreases similarly to the keyword rank's depreciation. This figure clearly demonstrates that the image labeled with the top words have higher confidence value.

4.2 Image Annotation Refinement

As mentioned in Section 1, the developed LCSS can be applied to help refine and re-rank the annotated keywords, which is called *image annotation refinement*.

In this section, we apply the lexicon on the University of Washington (UW) dataset to refine the annotation results obtained by the search-based image annotation algorithm [10]. Two different refinement strategies are shown in Figure 6.

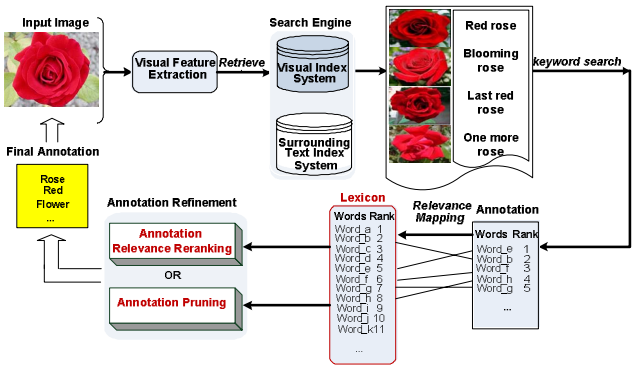


Figure 6: Image annotation refinement scenario

UW dataset is a popular content-based image retrieval database, which has been used in many cited work and is downloadable from (http://www.cs.washington.edu/research/imagedatabase/groundtruth/). For each image, there are about 5 manually labeled ground truth annotations. In total, it contains 1,109 images and more than 350 unique words. In our evaluation, we strictly use the annotations of UW as the ground truth and all synonyms and non-appearing correct annotations are assumed incorrect.

Search-based image annotation (SBIA) is a very recent annotation algorithm [10]. SBIA first uses the 64 color features to retrieve visually similar images. Next it applies a keyword search to obtain a ranked list of candidate annotations from the surrounding texts of the retrieved image.

Given a query image, SBIA is first employed to obtain a set of candidate annotations. Then, uses the developed LCSS to refine the annotations by re-ranking the candidate annotations and reserves the top ones (Figure 6). In our experiments, two new refinement strategies are used: *annotation relevance re-ranking* and *annotation pruning*.

4.2.1Annotation Relevance Re-Ranking

For each keyword appearing in the annotations of an image, for example $word\_a$, its relevance score, which reflects the relevance between the image and keyword can be calculated as:

$$Score\_r(word\_a) = \frac{1}{\ln(1+i)} \quad (7)$$

i is the rank of $word\_a$ in the annotation of the image. The assumption is that the keyword with top rank is more important to the image thus having larger relevance score.

Similarly, in the constructed lexicon, a static score of each keyword could also be defined to reflect its appropriateness as an effective annotation. It should be noted that our word lexicon LCSS is constructed over the 2.4 million web images, which is independent from the UW dataset. Therefore, the static score is independent of the target images.

$$Score\_s(word\_a) = \begin{cases} \frac{1}{\ln(1+j)}, & word\_a \in LCSS \\ 0, & word\_a \notin LCSS \end{cases} \quad (8)$$

j is the rank of the $word\_a$ in the lexicon list. If $word\_a$ does not appear in the lexicon list, its static score is defined as 0.

The final score of the keyword could be calculated as a weighted combination of the relevance score and the static score as follows:

$$Score\_c(word\_a) = Score\_r(word\_a) + \alpha \cdot Score\_s(word\_a) = \frac{1}{\ln(1+i)} + \alpha \cdot \frac{1}{\ln(1+j)} \quad (9)$$

In our experiments, α is set to 10 empirically from various tests.

From formula (9), we can see if one keyword is ranked high in the annotation but appears low in the LCSS, its final rank will decrease. Similarly, if one lower ranked keyword is within the top keywords of the LCSS, it will be ranked higher in the final annotation.

4.2.2Annotation Pruning

Annotation pruning is an alternative to annotation relevance re-ranking. The difference from annotation relevance re-ranking is that irrelevant annotations that do not appear in the lexicon are pruned. The basic assumption is that highly correlated annotations should be reserved and non-correlated annotation should be removed.

Since the original UW ground truth annotations include both keywords and phrases, in our experiments, we define two evaluation levels to evaluate the annotation performance: *phrase*-level and *term*-level. In the phrase-level, an annotation is considered to be correct if and only if it is a ground truth annotation of the target image. In the term-level, both the ground truth annotation phrases and the result annotation phrases are divided into separate words. If there is more than one same word in the annotations of an image, only one is reserved. An annotated keyword is considered to be correct if and only if it appears in the ground truth annotation of the target image. The *Precision* and *Recall* are defined as follows:

$$precision = \frac{1}{n}\sum_{k=1}^{n} \frac{\text{number of correctly annotated phrases (terms) in image } I_k}{\text{number of annotated phrases (terms) in image } I_k} \quad (10)$$

$$recall = \frac{1}{n}\sum_{k=1}^{n} \frac{\text{number of correctly annotated phrases (terms) in image } I_k}{\text{number of ground truth phrases (terms) in image } I_k} \quad (11)$$

n is total number of the test images. In our experiments, the number of SBIA's annotation keywords is restricted to be no more than 10.

4.2.3Size of Lexicon

Obviously, the size of lexicon is a crucial parameter in the annotation refinement. It decides how many keywords will be used to refine or prune the final annotation. Let's denote it by s. To facilitate the further evaluations, s is first determined by comparing the annotation performance with

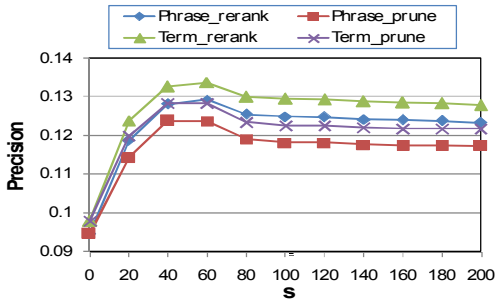


Figure 7: Annotation precision of different sizes of lexicon

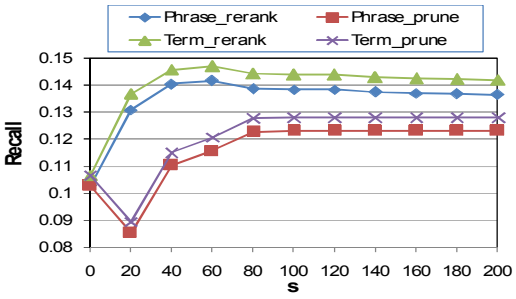


Figure 8: Annotation recall of different sizes of lexicon

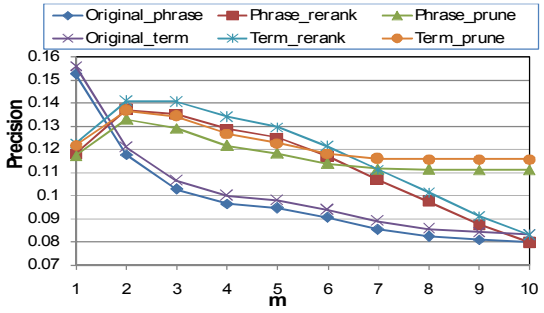


Figure 9: Annotation precision of different sizes of m

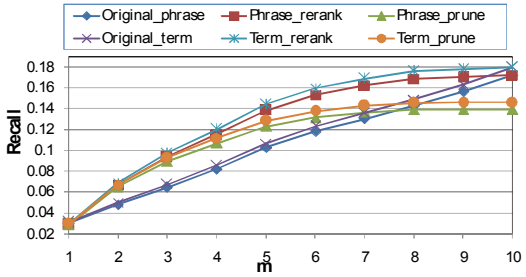


Figure 10: Annotation recall of different sizes of m

different *s*. The number of annotation results *m* is fixed to 5 since there are about 5 manually labeled ground truth annotations in the UW dataset. The *Precision* and *Recall* are shown in Figure 7 and Figure 8, with *s* changed from 0 to 200. When *s* is set to 0, it is the original annotation without refinement.

From these two figures, we can find that *Precision* and *Recall* exhibit similar varying characteristics. The refinement distinctively improves original annotation's precision and recall when *s* becomes larger. And the performance of refinement keeps stable when *s* is equal or larger than 100, which means the most annotation words of UW dataset fall into the first 100 keywords in the LCSS list. Therefore, *s* is set to be 100 in the following evaluations.

4.2.4Size of Annotation

In Figure 7 and 8, we can see that the annotation refinement consistently improves the performance when *m* is 5. In order to test the effect of different *m*, in our second experiment, *m* is varied from 1 to 10. The corresponding *Precision* and *Recall* are shown in Figure 9 and Figure 10, respectively. Three observations can be drawn from the results. Firstly, when *m* is ranging from 3 to 7, the *Precision* and *Recall* of refined annotation (annotation re-rank and pruning) are improved most. When *m* hits 10, the *Precision* and *Recall* of annotation re-rank becomes same as the unrefined one since all annotation words are counted for both methods. In addition, the *Precision* and *Recall* of the annotation pruning remains same especially while *m* is larger than 7. It means that most of top 7 annotation words of SBIA fall into the *LCSS* list. Secondly, the absolute values of term-level evaluation are better than that of phrase-level, because SBIA is a term-based annotation algorithm. Thirdly, the performance of annotation re-rank is better than pruning. The reason is that pruning method filters out some possible correct annotations.

4.2.5Comparison with Other Lexica

In our last experiment, we studied three different lexica of semantic concepts, where each set is larger than the previous one.

LCSS: Our developed lexicon list of concepts with small semantic gaps. To be consistent with previous experiments, we still use the top 100 keywords for annotation refinement.

LSCOM: Large Scale Concepts Ontology for Multimedia [13], a standardized lexicon established on broadcast news video from TRECVID benchmark. The largest word list, which contains 858 terms, can be downloaded from http://www.ee.columbia.edu/ln/dvmm/lscom/.

WordNet: A very large lexical database of English. Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Because most of image annotation words are nouns and adjectives, for this study, we use a total of 100,303 nouns and adjectives terms from WordNet 2.1 [16].

Since LSCOM and WordNet do not have the word rank, in this experiment, we only compare their annotation pruning performance of SBIA results on the UW dataset. From Figure 11 and Figure 12, it is clear that LSCOM and WordNet do not improve the annotation precision at all and perform much worse than LCSS. The reason is that

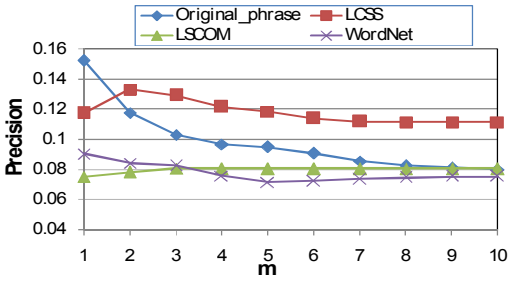


Figure 11: Annotation precision of different lexica

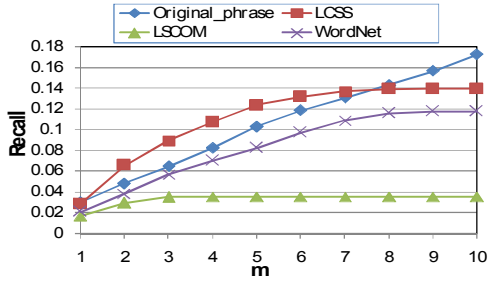


Figure 12: Annotation recall of different lexica

many correct annotations are not included in these two lexica thus are pruned. It validates that these two lexica are not good concepts corpus with small semantic gap for web-scale image annotations. Instead, LCSS demonstrates to be more effective for image annotation refinement.

5. Conclusion and Future Work

In this paper, we have presented an innovative framework to automatically construct a concept lexicon with small semantic gap from a large photo dataset.

Our major contributions are: (i) This work sheds some light in answering the question "what specific concepts have small semantic gaps?" Among the hundreds or even thousands of multimedia concepts, it is the first of its kind to be designed for which semantic concepts should be focused first for data collection and modeling. (ii) It also provides a candidate pool of good semantic concepts to annotate other image datasets, thus it can be used for annotation refinement and rejection. (iii)This lexicon list will have many potential applications in concept detection, query optimization and multimedia information retrieval.

It should be also noted that these concepts are related to low-level visual features. Except for the current color features used in this work, in the future we will investigate texture feature, shape feature and SIFT feature, *etc.,* to construct feature-based lexica. The family of these lexica will provide more options to different modeling methods given specific feature and could also be fused or combined to further refine annotation results based on various features.

Although a number of semantic concepts have been developed for MIR, some questions still remain to be answered in the future, *e.g.*, how many semantic concepts are necessary [17]? Which features are good for image retrieval with specific concept? Our work is a first step in answering these questions.

6. Acknowledgment

This work is supported in Part by ARO Grant W911NF-05-1-0404, and by DHS Grant N0014-07-1-0151. We also thank Dr. Changhu Wang for valuable discussions.

7. References

[1] S. Agarwal, and D. Roth. Learning a sparse representation for object detection. *Proc. of European Conference of Computer Vision (ECCV)*, Copenhagen, Denmark, 2002.

[2] L. Fei-Fei, R. Fergus, and P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. *IEEE CVPR Workshop on Generative-Model Based Vision*, 2004.

[3] G. Griffin, AD. Holub, and P. Perona. The Caltech-256. *Caltech Technical Report*.

[4] http://www.pascal-network.org/challenges/VOC.

[5] J. Li, and J. Z. Wang. Real-time computerized annotation of pictures. *Proc. of ACM Multimedia*, 2006

[6] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos, Supervised Learning of semantic classes for image annotation and retrieval. *IEEE PAMI*, 29(3):394-410, 2006.

[7] L. von Ahn, L. Dabbish. Labeling images with a computer game. *Proc. of ACM Conf. of Humor Factors Computing System*. 2004.

[8] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, LabelMe: a database and web-based tool for image annotation. *Tech. report*, MIT, AI Lab Memo 2005.

[9] X. J. Wang, L. Zhang, F. Jing, and W. Y. Ma. AnnoSearch: image auto-annotation by search. *Proc. of IEEE Conf. CVPR*, New York, June, 2006.

[10] C. Wang, F. Jing, L. Zhang, and H. J. Zhang. Scalable search-based image annotation of personal images. *Proc. of the 8th ACM international Workshop on Multimedia information Retrieval*, Santa Barbara, CA, USA, 2006. 10.

[11] J. Ponce, T. L. Berg, M. Everingham, and D.A. Forsyth, *et al*. Dataset issue in object recognition. *Toward Category-Level Object Recognition*, *LNCS* 4170:29-48, 2006.

[12] C. G. Snoek, M. Worring, J. C. van Gemert, J. M. Geusebroek, and A. W. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. *Proc. of ACM Multimedia*, 2006.

[13] C. M. Naphade, J. R. Smith, J. Tesic, and S. F. Chang, *et al.* Large-scale concept ontology for multimedia. *IEEE MultiMedia*, 13(3):86–91, 2006.

[14] L. Zhang, Y. Hu, M. Li, W. Ma, and H. Zhang. Efficient propagation for face annotation in family albums. *Proc. of ACM Multimedia*. New York, NY, 2004.

[15] B. J. Frey, and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972-976, 2007.

[16] G. A. Miller. WordNet: A lexical database for English. *Communication of ACM*, 38(11): 39–41, 1995.

[17] A. Hauptmann, R. Yan, and W. H. Lin. How many high-level concepts will fill the semantic gap in video retrieval? *Intl. conf. on image and video retrieval (CIVR)*, 2007.