PRISM: Concept-preserving Summarization of Top-K 
Social Image Search Results

Boon Siew Seah  
School of Computer Engineering  
Nanyang Technological University  
Singapore

Sourav S Bhowmick  
School of Computer Engineering  
Nanyang Technological University  
Singapore

Aixin Sun  
School of Computer Engineering  
Nanyang Technological University  
Singapore

hunting.bearz@outlook.com assourav@ntu.edu.sg axsun@ntu.edu.sg

ABSTRACT
Most existing tag-based social image search engines present search results as a ranked list of images, which cannot be consumed by users in a natural and intuitive manner. In this demonstration, we present a novel concept-preserving image search results summarization system called PRISM. PRISM exploits both visual features and tags of the search results to generate high quality summary, which not only breaks the results into visually and semantically coherent clusters but it also maximizes the coverage of the original top-k search results. It first constructs a visual similarity graph where the nodes are images in the top-k search results and the edges represent visual similarities between pairs of images. This graph is optimally decomposed and compressed into a set of concept-preserving subgraphs based on a set of summarization criteria. One or more exemplar images from each subgraph is selected to form the exemplar summary of the result set. We demonstrate various innovative features of PRISM and the promise of superior quality summary construction of social image search results.

1. INTRODUCTION
The rising prominence of image sharing platforms like Flickr and Instagram in the last decade has led to an explosion of social images. Consequently, the need for superior social image search engines to support efficient and effective tag-based image retrieval (TbIR) has become increasingly pertinent. Similar to traditional search engines, queries in a tag-based social image search engine are often short and ambiguous. As a result, search engines often diversify the search results to match all possible aspects of a query in order to minimize the risk of completely missing out a user’s search intent. An immediate aftermath of such results diversification strategy is that often the search results are not semantically or visually coherent. For example, the results of a search query "fly" (Figure 1) may contain a medley of visually and semantically distinct objects and scenes (hereafter collectively referred to as concepts) such as parachutes, aeroplanes, insects, birds, and even the act of jumping.

Image search results are typically presented as a ranked list of images often in the form of thumbnails (e.g., Figure 1). Such thumbnail view of ranked images enables end users to quickly glance through a set of images without browsing through them iteratively. However, it suffers from two key limitations. First, it fails to provide a view of common visual objects or scenes collectively. For example, the result images of "fly" query can be clustered by visual objects (e.g., aeroplane, insect) and activities (e.g., jump). Such organized image search results will naturally enable a user to quickly identify and zoom into a subset of results that is most relevant to her query intent. Second, a thumbnail view fails to provide a bird eye view of different concepts present in a query results. For instance, reconsider Figure 1. It will be beneficial to users if a suitable exemplar image from each type of concept can be selected to create a “summary” of the search results. This will enable a user to get a bird eye view of various key concepts associated with the results.

An appealing way to organize social image search results of a search query is to generate a set of image clusters from them such that images in each cluster are semantically and visually coherent and the clusters maximally cover the entire result set. Subsequently, at least one exemplar image from each cluster can be selected to generate an exemplar summary of the entire result set to give a bird eye view of different concepts in it. We advocate that such image clusters must satisfy the following desirable features.

- Concept-preserving. Each cluster should be annotated by a minimal set of tags generated from the images within to semantically\(^1\) describe all images in the cluster. Users therefore can easily associate the tag(s) with the images in a cluster at a glance. We refer to such a cluster as concept-preserving.

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-nd/3.0/. Obtain permission prior to any use beyond those covered by the license. Contact copyright holder by emailing info@vldb.org. Articles from this volume were invited to present their results at the 41st International Conference on Very Large Data Bases, August 31st - September 4th 2015, Kohala Coast, Hawaii.

Proceedings of the VLDB Endowment, Vol. 8, No. 12
Copyright 2015 VLDB Endowment 2150-8097/15/08.

\(^1\)We assume that the tags are high-level semantic concepts assigned by image uploaders or annotators.

Figure 1: [Best viewed in color] Sample query results.
where a set of images shares at least one concept (tag)\(^2\). For instance, in a concept-preserving "helicopter" cluster, a single "helicopter" tag is sufficient to represent all images in it and describe them semantically.

- **Visual coherence.** Images in a cluster must be visually coherent. Visually similar images must be clustered together and dissimilar images must be separated in different clusters.
- **Coverage.** The image clusters should cover as much of the result set as possible in order to maximize incorporation of all possible query intent. In other words, image clusters should represent majority of the original result images.

In this demonstration, we present a system called **prism**\(^2\) (concept-preserving social Image Search summarization) [7] that constructs high quality summary of top-\(k\) social image search results based on concept-preserving and visually coherent clusters which maximally cover the result set. Figure 2 depicts subsets of clusters constructed by **prism** for the query "fly". Each cluster is represented by minimal tag(s) shared by all images in it. Due to the concept-preserving nature, the images in a cluster form an equivalence class with respect to the tags. Consequently, any image in each cluster can be selected as an exemplar without loss of accuracy to facilitate generation of high quality exemplar summary of the result set. For instance, consider the "insect" cluster. Any image can be chosen as an exemplar to represent the "insect" concept.

Any query-specific image search results summarization presents several non-trivial challenges. The set of images to be summarized is not predetermined. Hence, the summarization method does not have the luxury of preprocessing the underlying images apriori. Additionally, simply leveraging traditional image clustering techniques may not generate high-quality summary due to the requirement that any summary must be concept-preserving and cover as many images as possible in the result set. To address these challenges, **prism** explores the concept space (i.e., tag space) to seek for visually coherent cluster of images. Specifically, it first constructs a visual similarity graph \(G\) where the nodes are images in the search results and the edges represent visual similarities between pairs of images. Then it optimally decompose \(G\) into a set of concept-preserving subgraphs based on the aforementioned desired features of image clusters. Particularly, images in each subgraph represents a concept-preserving cluster. Following that, **prism** performs a series of image set compression to simplify the subgraphs to form the final set of concept-preserving subgraphs. Lastly, one or more exemplar images from each subgraph is selected to form the exemplar summary.

2. RELATED SYSTEMS AND NOVELTY

One strategy to summarize image search results is by clustering tagged social images based on both visual and textual features as advocated by early fusion [6, 9] and late fusion [5] approaches. The former exploits the tags and visual content of the images jointly whereas the latter considers them independently. However, these techniques do not ensure that the generated summaries are concept-preserving and maximally covers the image results. Furthermore, unlike **prism**, most of these techniques do not associate each cluster with a tag concept for user interpretation and visualization. As such, one has to associate tag(s) to each image cluster as a post-processing step.

Another approach of image summarization is to find a set of exemplars that summarize the image set. In [3], a set of exemplars is identified using a sparse Affinity Propagation (AP) approach. Xu et al. [9] evaluates visual and textual information jointly to identify exemplar images. It extends the AP algorithm to support heterogeneous messages from visual and textual feature spaces. In contrast to **prism**, these approaches do not attempt to ensure that all other images can be properly clustered by their exemplars (and their tags) in a concept-preserving manner. Additionally, they do not ensure that the exemplars maximally cover the image set. Note that even for query-specific image categorization techniques provided by Web image search engines (e.g., Google Images (images.google.com)), where data associated with images are not as sparse as social images, there is little evidence whether they maximally cover the results. For example, consider the image categories generated by Google Images (Figure 3) for the query "fly". Despite having significantly larger datasets and richer set of web text annotations, these search engines still construct relatively limited variety of concepts. The concepts suggested by Google Images are mostly restricted to insects and cliparts, missing out other fly-related concepts such as the act of jumping, plans, helicopter, and birds.

3. SYSTEM OVERVIEW

**prism**\(^2\) is implemented using Java and Scala using the Play 2.0 framework (www.playframework.com). Figure 4 shows the system architecture of **prism** comprising of the following modules.

The **Indexer Module.** This module extracts query-independent tag features (e.g., tag relatedness, tag frequency, tag co-occurrence, etc.) from the underlying collection of social images \(D\). The relatedness between a tag \(t\) and its annotated image \(d\) is measured using neighborhood voting as described in [4]. Tag frequency of a tag \(t\) is the number of images annotated with \(t\). Tag co-frequency between two tags \(t_1\) and \(t_2\) is the number of images annotated by both \(t_1\) and \(t_2\). These two features are used to compute tag co-occurrences.

\(^2\)A prism can be used to break a beam of light up into its constituent spectral colors (the colors of the rainbow). Similarly, the **prism** system breaks the result image set into distinct image clusters.

\(^4\)All results related to Google Images are last accessed on June 14th, 2015.
using different measures (e.g., Jaccard coefficient, Pointwise Mutual Information, Pointwise KL divergence). The extracted data are then stored in a RDBMS.

The Image Search Module. This module encapsulates a standard TadIR search engine. Given a keyword query $Q$, it leverages the Index Module to retrieve the top-k images that best match $Q$ where $k$ is a user-specified number of desired images. Each image $i$ in the result set comprises of a $d$-dimensional visual feature vector representing visual content of the image and a set of tags $T_i$ representing concepts associated with the image by users. Note that the image retrieval algorithm is implemented on top of Lucene (lucene.apache.org) and is orthogonal to PRISM. In fact, any superior social image retrieval technique can be adopted for PRISM. Here, we adopt the framework in [8] for multi-tag queries.

The Visual Similarity Graph Constructor Module. Given the top-k result images of a query $Q$, this module constructs a visual similarity graph based on pair-wise visual similarity between images where each node in the graph is an image. To this end, we adopt cosine similarity to measure the visual similarity between any two images as follows: $Sim = L^{-1/2}A^{T}AL^{-1/2}$ where $A$ is the $n \times d$ matrix of image set visual features, $A^{T}A$ encodes the inner-product of the image feature vectors, and $L^{-1/2}$ is a $n \times n$ diagonal matrix that encodes normalization of each feature vector. Given the similarity matrix, the visual similarity graph $G = (V,E)$ is constructed as follows. Let $V$ be the set of images. We add an edge in $E$ between two images $i$ and $j$ if $Sim_{ij} > \delta$ where the weight of this edge is $Sim_{ij}$ and $\delta$ is the edge density threshold. Figure 5(i) illustrates a visual similarity graph.

The Concept Subgraphs Generator Module. Intuitively, PRISM formulates the summarization problem as the optimal decomposition of a visual similarity graph $G$ into a set of concept subgraphs from which exemplar images are drawn to create the summary. Given a set of tags $T$, a concept-preserving subgraph (concept subgraph for brevity), denoted by $C_T = (V_T, E_T, T)$, is a subgraph of $G$ induced by $V_T \subseteq V$. Every image in the subgraph shares the set of tags $T$, i.e., $T \subseteq T_i \forall i \in V_T$. We use concept subgraphs to model a set of images that preserves a set of concepts represented by $T$. That is, images in each concept subgraph represent a concept-preserving cluster. We can represent it in $G$ consicely by an exemplar node labeled with $T$. Figure 5(ii) depicts a set of exemplar nodes (represented by dashed circles) with labels ($T$) "surf", "beach", "sea", and "sun". These nodes represent the concept subgraphs induced by $\{v_1, v_2, v_3\}$, $\{v_8, v_9, v_{10}\}$, $\{v_4, v_5, v_6, v_7, v_9\}$, and $\{v_{11}, v_{12}, v_{13}, v_{14}\}$, respectively.

This module’s goal is to optimally decompose $G$ into concept subgraphs so that it can facilitate high quality summary construction. Specifically, a decomposition of $G$ generates a set of concept subgraphs $S = \{C_{T_1}, C_{T_2}, \ldots, C_{T_r}\}$ and a remainder subgraph $R$, such that the image set in $G$ is union of all images in $S$ and $R$. Each $C_{T_i} \in S$ can be represented by an exemplar node; the remainder subgraph $R$ represents the region of $G$ not covered by $S$ (i.e., $R$ is the subgraph induced by the set $V \setminus \bigcup_{C_{T_i} \in S} V_{T_i}$). For example, the visual similarity graph in Figure 5(i) is decomposed into $\{C_{surf}, C_{beach}, C_{sea}, C_{sun}\}$ and $R$ where $C_{surf}, C_{beach}, C_{sea}$, and $C_{sun}$ are represented by exemplar nodes "surf", "beach", "sea", and "sun", respectively, and $R = \{v_{15}, v_{16}\}$. Our decomposition allows overlap among subgraphs in $S$ (e.g., overlap between $C_{beach}$ and $C_{sea}$) and is guided by the following summarization objectives.

- **Visual coherence.** The visual coherence of $S$ is defined as:
  \[
  \text{coherence}(S) = \frac{1}{|S|} \sum_{C_{T_i} \in S} \sum_{e \in E_{C_{T_i}}} \frac{w(e)}{|E_{T_i}|}
  \]

  The $\text{coherence}(S)$ value reflects the average weight of visually similar images in each $C_{T_i} \in S$. Higher visual coherence means the images are more visually similar to each other.

- **Distinctiveness.** Intuitively, a pair of exemplar nodes that represent two disjoint subgraphs is more informative that a pair that represent identical subgraphs. We quantify this objective with the distinctiveness measure as follows.
  \[
  \text{distinctiveness}(S) = \frac{|\cup_{C_{T_i} \in S} V_{T_i}|}{\sum_{C_{T_i} \in S} |V_{T_i}|}
  \]

- **Coverage.** A set of concept subgraphs $S$ that well represents $G$ is preferable. We use the notion of coverage to measure this. Intuitively, it quantifies how many images from the image set $V$ appears in $S$. Formally, it is defined as:
  \[
  \text{coverage}(S) = \frac{|\cup_{C_{T_i} \in S} V_{T_i}|}{|V|}
  \]

Note that $\text{coverage}(S)$ is 1 if all images in $V$ are selected in $S$.

This module implements a weighted minimum $k$-set cover-based strategy [2] to find an optimal set of concept subgraphs $S$ such that $\text{coherence}(S)$, $\text{coverage}(S)$ and $\text{distinctiveness}(S)$ are maximized. Since the problem is NP-hard, an $H_k$-approximation greedy algorithm, where $H_k = \sum_{i=1}^{k} \frac{1}{i}$ is adopted towards this goal [2]. It includes a cost model that incurs a weight (i.e., cost) every time a subgraph is added as concept subgraph or as remainder subgraph. For each concept subgraph, it incurs a visual incoherence cost, the inverse of visual coherence of a concept subgraph, for choosing visually incoherent images (maximize $\text{coherence}(S)$). For each remainder subgraph, it incurs a remainder penalty cost for choosing large remainder subgraphs (maximize $\text{coverage}(S)$). Given the cost model, it finds the minimum weight (cost) of subgraphs needed to cover $V$, penalizing redundant subgraphs that add little to the summary since every subgraph added incurs a cost (controlling $\text{distinctiveness}(S)$). Note that state-of-the-art graph clustering techniques (e.g., [6]) cannot be directly leveraged by this module to identify these concept subgraphs as they do not preserves concepts.
Panel 1

S prism a compressed set provides more concise view of the result set. In {summary of the search result, then
T share conceptual similarity. For example, assume that S images have at least one common concept) are contracted as they
set and replaces them with contraction of pairs concept subgraphs. The
contraction of pairs \( C_{T_1} \) and \( C_{T_2} \) removes both subgraphs from the
set and replaces them with \( C_{T_1 \cup T_2} = \langle V_{T_1} \cup V_{T_2}, E_{T_1} \cup E_{T_2} \rangle \). Note
that only those pairs that share a non-empty set of concepts (i.e., all
images have at least one common concept) are contracted as they
share conceptual similarity. For example, assume that \( S \) contains
two subgraphs with \( T^1 = \{ \text{boat}, \text{sail}, \text{rock} \} \) and \( T^2 = \{ \text{rock}, \text{cliff} \} \).
Then these two subgraphs can contracted into a larger subgraph
sharing the \( \{ \text{rock} \} \) concept. Observe that if a user wants a detailed
summary of the search result, then \( S \) is most appropriate for
generating exemplar summaries. If a broader overview is desired, then
a compressed set provides more concise view of the result set. In
prism, by default we use \( S^1 \) to create the exemplar summary.

The Compressor Module. The preceding module generates an
optimal collection of concept-preserving clusters without constraining
each cluster size. This is beneficial as it enables us to select the
“best” combination of clusters with highest visual coherency. On the
other hand, there is a lack of control over the summary granularity if each concept subgraph in the constructed \( S \) is used for creating the exemplar summary (detailed in the Exemplar Summary Constructor Module) as \( S \) may contain too finely-grained clusters for presentation to users. We assume that a user expects a summary at a particular summary granularity. For instance, if a user wants a broad overview of the search result, then a summary of 5 exemplars may be preferable to a summary of 50 exemplars. On the other hand, if a user prefers a detailed summary, then the summary with 50 exemplars is better.

The Compressor module addresses this issue by aggregating con-
cept subgraphs iteratively to build a multilevel compression scheme
at varying summary granularity. Given the initial \( S \), it constructs a
list \( \{ S, S^1, S^2, \ldots , S^i \} \) such that \( \forall i, j, |S^i| > |S^j| \) if \( i < j \). Each
\( S^i \) is called a compressed concept subgraph set of \( S \). Each successive
set \( S^{i+1} \) is a compressed representation of its predecessors \( (S^i) \) and is constructed by contracting pairs of concept subgraphs. The
contraction of pairs \( C_{T_1} \) and \( C_{T_2} \) removes both subgraphs from the
set and replaces them with \( C_{T_1 \cup T_2} = \langle V_{T_1} \cup V_{T_2}, E_{T_1} \cup E_{T_2} \rangle \). Note
that only those pairs that share a non-empty set of concepts (i.e., all
images have at least one common concept) are contracted as they
share conceptual similarity. For example, assume that \( S \) contains
two subgraphs with \( T^1 = \{ \text{boat}, \text{sail}, \text{rock} \} \) and \( T^2 = \{ \text{rock}, \text{cliff} \} \).
Then these two subgraphs can contracted into a larger subgraph
sharing the \( \{ \text{rock} \} \) concept. Observe that if a user wants a detailed
summary of the search result, then \( S \) is most appropriate for
generating exemplar summaries. If a broader overview is desired, then
a compressed set provides more concise view of the result set. In
prism, by default we use \( S^1 \) to create the exemplar summary.

The Exemplar Summary Constructor Module. This module
selects one or more exemplar images (by default we chose three
images) from each summarized concept subgraph to form the ex-
emplar summary (Figure 5(iii)). Note that since the set of images in each concept-preserving cluster forms an equivalence class with respect to its concept set, any image in the set can be selected as an exemplar to associate with the concept.

The PRISM GUI Module. Figure 6 depicts the user interface
of PRISM using the query "art". It consists of two panels. A user
issues a tag query by keying keyword(s) in Panel 1. Clicking on the
"Spanner" icon in Panel 1 will invoke the configuration dialog box
to set various parameters (e.g., desired number of k images, edge
density threshold, summary granularity, etc.). Once the query is
processed, the top-k images are displayed as a visual summary in
Panel 2. Item 1 in this panel provides an overview of the statistics
related to the summary. For example, it shows the number of im-
ages represented by the exemplar summary, the number of concept-
preserving clusters retrieved, and the number of unique images re-
presented by the summary. Panel 3 displays the exemplar summary as
horizontal blocks of images where each block is labeled with \( T_i \)
and represents the exemplar images of a concept-preserving clus-
ter \( C_{T_i} \). For instance, Item 2 points to the exemplar images of the
painting cluster. A user may click on a block to reveal a popup
(Item 3) describing various statistics pertaining to the images within
the cluster such as (a) number of images within a cluster and (b) top
ranked tags most closely associated with the cluster. Additionally,
the user has the option to view, in a separate pane, all images within
the cluster by following a link.

4. DEMONSTRATION OVERVIEW

Our demonstration will be loaded with the nus-wide dataset [1]
containing 269,648 images from Flickr\(^5\). We aim to showcase the
functionality and effectiveness of the prism system in summariz-
ting top-k query results. Example queries will be presented. Users
can also write their own ad-hoc queries through our gui. A video
of prism is available at https://www.youtube.com/watch?v=dh1A0YzC3I&feature=youtu.be.

One of the key objectives of the demonstration is to enable the
audience to interactively experience the proposed search results
summarization framework in real-time. Through our gui, the user
will be able to formulate search queries (Panel 1) and browse the
exemplar summary of the top-k results (k can be specified by the user)
generated by prism (Panel 2). Going a step further, the user may
click on the exemplar images of any concept-preserving cluster
which will allow her to view immediately all images in the cluster
as well as information about relevant tags. Additionally, by set-
ting different values for k and varying summary granularity, she can
view changes to the exemplar summaries.

Acknowledgements: Boon Siew Seah and Aixin Sun were sup-
sported by Singapore MOE AcRF Tier-I Grant RG13/10.

5. REFERENCES

web image database from National University of Singapore. In ACM CIKR,
2009.


[6] M. Rege, M. Dong, J. Hua. Graph theoretical framework for simultaneously
integrating visual and textual features for efficient web image clustering.


\(^5\)Note that we use the popular nus-wide dataset instead of any other larger social image
collection because its size does not impact the performance of prism as the focus here
is to summarize top-k results regardless of the image retrieval process.