CASIS: A System for Concept-Aware Social Image Search

Ba Quan Truong
bqtruong@ntu.edu.sg

Aixin Sun
axsun@ntu.edu.sg

Sourav S. Bhowmick
assourav@ntu.edu.sg

School of Computer Engineering, Nanyang Technological University, Singapore 639798

ABSTRACT

Tag-based social image search enables users to formulate queries using keywords. However, as queries are usually very short and users have very different interpretations of a particular tag in annotating and searching images, the returned images to a tag query usually contains a collection of images related to multiple concepts. We demonstrate Casis, a system for concept-aware social image search. Casis detects tag concepts based on the collective knowledge embedded in social tagging from the initial results to a query. A tag concept is a set of tags highly associated with each other and collectively conveys a semantic meaning. Images to a query are then organized by tag concepts. Casis provides intuitive and interactive browsing of search results through a tag concept graph, which visualizes the tags defining each tag concept and their relationships within and across concepts. Supporting multiple retrieval methods and multiple concept detection algorithms, Casis offers superior social image search experiences by choosing the most suitable retrieval methods and concept-aware image organizations.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—Information Filtering

Keywords

Flickr, TagIR, Tag Concept, Tag Relation, Image Search

1. INTRODUCTION

The availability of social tags on various image sharing platforms (e.g., Flickr) raises the opportunity of building effective tag-based social image retrieval systems. In contrast to content-based image retrieval paradigm, which searches for visually similar images of a given query image, tag-based image retrieval (TagIR) enables users to formulate semantic queries using textual keywords through an interface similar to Web search engines. Interestingly, similar to general Web search queries, the queries for tag-based image search are usually very short, consisting of 2.2 tags on average for each search [13].

A short query, particularly a single-tag query, usually leads to a large number of potentially relevant images. More importantly, these images often represent multiple concepts because: (i) a single-tag has limited expressiveness in precisely defining the information need and (ii) social tags are noisy in nature as different users might have very different understandings of a particular tag in annotating and searching images. Figures 1(a) and 1(b) depict the search results of two sample single-tag queries, \( Q_{\text{rock}} \) and \( Q_{\text{sea}} \), respectively. Observe that the results of \( Q_s \) capture two different concepts. Specifically, the highlighted images (in green border) are about rock music while the other images are related to stone rock or mountain. Similar phenomenon is observed for \( Q_{\text{sea}} \). Although most images returned for \( Q_s \) are good matches to the query, the highlighted ones are mainly related to ship and the remaining ones deal with seascape (beaches, sky, clouds, etc.). This multiple concepts phenomenon might be attributed to the ambiguity of the query itself (e.g., rock) or the different aspects of the query (e.g., sea), among other reasons. Importantly, a user is usually interested in only one of the concepts and not all of them. Consequently, displaying images representing multiple concepts in a single ranked list may adversely affect superior image search experiences.

In this demonstration, we present a social image search system called Casis (concept-aware social image search), which exploits the notion of tag concept (or simply concept) to address the aforementioned problem. The intuition is that a specific meaning or aspect of a short (particularly single-tag) query can be well described by a group of highly related tags. Each such group of tags is referred to as a tag concept. Accordingly, images matching the query can be organized into groups, each of which matches one tag concept. For instance, Figure 2 depicts the reorganization of the search results of \( Q_{\text{rock}} \) and \( Q_{\text{sea}} \) in Casis according to their corresponding tag concepts. Note that, Casis supports queries consisting of single or multiple tags. In this paper, we mainly use single-tag queries for illustration purpose.
Intuitively, for a given query, Casis retrieves a set of images best matching the query mainly based on the relevance scores between the query tag(s) and the images. This list of images serves as the initial search results to be presented to the user. From the initial results, the co-occurring relationships among the frequent tags are identified to form a Tag Relation Graph (TRG). A node in a TRG is a tag and the weight of an edge between two tags represents the strength of the co-occurrence of the two tags. Figure 5(a) gives an example TRG for $Q_{sea}$. A graph-cut algorithm is then applied to (softly) cut the TRG to form a Tag Concept Graph (TCG). A TCG is a graph with meta-nodes, each representing a group of tags. Tags in each group, linked by co-occurrence relationships, jointly represent a tag concept. The links across meta-nodes indicate the relationships between tag concepts. Note that two tag concepts associated with a query may be unrelated (e.g., rock music and rock stone) or related (e.g., ship and seascape) to each other. The concepts may also overlap if they share some identical tags. A tag concept may even subsume another by containing all the latter’s tags as semantically concepts can be defined at different levels of abstraction. For visualization, TRG and TCG are superimposed to illustrate the relationships between tags and tag concepts (See Figure 5).

The benefit of Casis is two-fold. First, the search results in Casis are better organized into groups where each group of images is relevant to a more cohesive tag concept. Note that an end user may not always be aware of the existence of such concepts due to the collaborative nature of social tagging. Different users may use the same tag for different purposes and in very different manners. Hence, the tag concepts provide a superior mechanism to browse the search results. Second, by automatically identifying the tag concepts, the original tag query can be enriched by adding the concept-specific tags. As the information need is now much more clearly defined, accurate image search results are expected for each concept-enriched tag query. Note that in [10] we have empirically demonstrated that the search accuracy of a single-tag query is largely affected by the choice of tag relatedness measure (i.e., the effectiveness of a tag in describing its annotated image) and the matching model between the query tag and an image’s tag (see Section 2 for more details). However, for a multi-tag query, the presence of all query tags in an image largely guarantees a very good match. Thus, Casis facilitates reﬁnement of the original query by adding concept-speciﬁc tags as well as retrieval of concept-relevant images within the initial search results of the query.

Clustering search results and presenting the results in groups have been studied in traditional Web search setting (see [2] for a survey). However, there are at least two key differences between Web clustering engines and concept-aware TasiIR. Firstly, there exists a fundamental difference in the way users perceive text and image data. In Web search engines, a user needs to read the title/snippet of a hit to judge its relevance, one by one. In contrast, a glance at image thumbnails could easily tell a user whether a set of images are relevant. Secondly, Web search results and social images are represented in very different feature spaces which calls for different clustering techniques. These two key differences pose new challenges toward realizing concept-aware social image search. Particularly, it is important for Casis to provide an intuitive user interface to facilitate quick judgment of relevance as well as the ﬂexibility in browsing multiple tag concepts.

### 2. RELATED WORK

#### Image Search Results Clustering

Most approaches for clustering image search results exploit low-level visual features [3]. However, these approaches suffer from two problems: (a) semantic gap between the low-level features and high-level semantics and (b) low efﬁciency due to curse of dimensionality. $i$Group [12] took a step towards addressing these limitations by exploiting textual features such as image captions, snippets, surrounding texts. The clustering is then accomplished by combining both visual features and textual features. In the context of social tagged images, shared nearest neighbors algorithm (snn) was applied to cluster images in a collection using both tag features and visual features [5]. In Casis, we aim to detect the concepts associated with a tag query and group the images according to the detected concepts. To be detailed in the next section, Casis is ﬂexible enough to easily accommodate different clustering algorithms (snn should be one of them).

#### Tag-based Image Browsing

Using tags in searching and browsing social images has been explored from multiple dimensions. The systems presented in [4] and [1] explore temporal and spatial dimensions, respectively. Other systems utilize multi-faceted browsing to produce coherent image groups. The facets can be manually speciﬁed [11] or automatically detected using WordNet [7] or Wikipedia [6]. MediaFaces [11] extracts location, celebrity, movie entities from pre-determined sources such as Yahoo!Travel, GeoPlanet, Wikipedia, etc. A query is then mapped to a stored entity whose related facets are then used to produce groups of semantically related images. Olive [7] extracts concepts from WordNet nouns for faceted browsing. WordNet, however offers relatively low coverage for the fast-changing keywords such Flickr tags. To address this problem, Wikipedia is considered in [6] for improving the coverage which is then used in TagExplorer [8], a faceted browsing system for Flickr photos. Different from faceted browsing where the facets are often pre-determined, Casis detects tag concepts automatically in real-time from the tagged images.

#### Tag-based Image Searching

Concept-aware social image search relies on an effective TasiIR ranking method to retrieve the images that best match the query before these images can be grouped into different concepts. In Casis we adopt the TasiIR framework consisting of five orthogonal dimensions for ﬂexibly deﬁning a speciﬁc TasiIR ranking method [10]. The ﬁve dimensions include tag relatedness for measuring the degree of effectiveness of a tag describing the tagged image, tag discrimination for quantifying the degree of discrimination of a tag with respect to the image collection, tag length normalization analogous to document length normalization in Web search, tag-query matching model for computing matching score between an image tag and a query tag, and query model for rewriting tag queries. A systematic evaluation of hundreds of TasiIR methods is reported in [10]. In [9], we show that tag concepts can be used to improve image tag recommendation.

### 3. SYSTEM OVERVIEW

In this section, we give an overview of the Casis system. Figure 3 depicts the system architecture. The tagged image retriever module retrieves a relatively large number of images for further processing by the initial results processor. The latter delivers the
3.2 Tagged Image Retriever Module

Given a query $Q$ containing one or more query keywords, this module retrieves top-$N$ images that best match $Q$ where $N$ is the predefined size of initial search results (default setting $N=5000$). Most TdIR algorithms can be adopted by this module, and in our implementation, we adopt the framework in [10] for supporting various settings in TdIR (see Section 2). This module relies on two databases, the tagged images and the tag index where the tag features are extracted by Global Tag Feature Extractor. The indexes are implemented using MySQL$^4$ and Lucene$^5$, respectively. The images in the initial search results are then indexed in an in-memory Lucene index for further processing. This in-memory index will be used to search for concept-specific images.

Global Tag Feature Extractor. This component extracts query-independent tag features (e.g., tag frequency, tag relatedness, tag associations, etc.) from the collection of social images. We briefly discuss a subset of features used in Casis. 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 associations (or co-occurrences) using different measures (e.g., Jaccard coefficient, Pointwise Mutual Information, Pointwise KL divergence). To support the framework in [10], the tag relatedness between an image and any of its annotated tags is computed and stored. The default tag relatedness is by neighbor voting. A tag $t$ receives a high relatedness score towards image $d$ if many of the nearest neighbors of $d$ by visual similarity are also annotated by $t$.

3.3 Initial Results Processor Module

This module is responsible for constructing query-dependent tag relation graph and tag concept graph. We detail the three major components in this module.

Local Tag Feature Extractor. This component first extracts the tags that are most related to the query tags from the initial results. Hence the tags and their features extracted are query-dependent. The features for the tags include tag frequency and tag co-frequency for constructing tag relation graph. Given that there can potentially be a large number of distinct tags associated with all the images in the initial results, the key issue here is to determine which tags to consider for subsequent processing.

Let $T_d$ be the set of tags associated with an image $d$. For each image, we extract its top-$r$ related tags ordered by tag relatedness, where $r = \max(\tau, \lceil |T_d|/\rho \rceil)$. Both $\tau$ and $\rho$ are configurable in Casis ($\rho = 0.1$ and $\tau = 4$ by default). In other words, for each image, we want to consider a reasonably small set of tags that best visually describe the image. The main reason is that tags are noisy in nature and as a result many of them do not effectively describe the images. On the other hand, some images may not be well tagged and $|T_d|$ can be very small; $\tau$ is introduced to avoid a very small $r$. The extracted tags are then considered as candidate tags for constructing the tag relation graph, and subsequently tag concept extraction.

Tag Relation Graph Constructor. This component constructs the tag relation graph (tag) from the initial results. The nodes in a tag are the most frequent tags among the candidate tags identified by the local tag feature extractor with respect to the tag frequency in the entire dataset. Let $D_Q$ be the initial results for query $Q$, and $D$ be the entire dataset. The normalized relative frequency of a tag $t$ is $f(t) = \frac{\sum_{d \in D_Q} \mathbb{1}(d \in D \land t \in d)}{|D_Q|}$, where $d \in D_Q$ denotes the number of images in the initial results for query $Q$. If a query has fewer than $N$ matching images, then all matching images are retrieved.

$^4$http://www.mysql.com/

$^5$http://lucene.apache.org

---

3.1 User Interface

Recall that the user interface design is crucial in addressing the challenges in concept-aware TdIR. Figure 4 depicts the main gui of Casis using the query $Q$(sea). It consists of three panels. User issues a tag query by keying keyword(s) and the desired number of $h$ image hits in the search panel (Panel 1). Clicking on the Casis label (purple color) will invoke the configuration dialog box to set various parameters. Once the query is processed, the top-$h$ image hits are displayed in the first tab (labeled “All”) in result panel (Panel 2). When we mouse over each image, its associated tags are displayed sorted by descending order of tag relatedness. Clicking on tab id on the top-left corner of the result panel displays the top-$h$ image hits for the concept id. For example, in Figure 4, the image hits for the tag concept ship, boat are displayed in Panel 2. The corresponding tag concept id is highlighted in the tag concept graph visualized in Panel 3, showing the relationship between tags within and across the concepts. Mouse hovering on the other parts of the tcg brings up other concept ids. Double-clicking a tag in the graph refines the query by adding the tag into the query box in Panel 1. The tcg visualization is implemented using Prefuse package$^2$ allowing color-coded tag concepts with overlapping and containment relationships. Mouse hovering a large concept may reveal sub-concepts contained in it. Tag concepts are visualized using different background colors.

---

$^2$http://prefuse.org/
set $S$ annotated by $t$. The number of nodes in a \textsc{trg} is $n \times \log(|D_0|)$ where $n$ is configurable and set to $1.5$ by default.

An edge between two tags in a \textsc{trg} can be computed using any first-order co-occurrence measure. \textsc{Casis} also supports second-order co-occurrence. Specifically, for each tag in the \textsc{trg} node set, we get its top-$\ell$ most associated tags within the candidate tags by a select-

The second-order co-occurrence between two tags is computed by cosine similarity between two such vectors.

With second-order co-occurrence, edge weights between any pair of nodes in a \textsc{trg} are likely to be non-zero, making the \textsc{trg} almost a complete graph. Thus, the edges are further filtered using $k$-nearest neighbor and $\epsilon$-neighborhood models. The $\epsilon$-neighbor model keeps top-$\ell$ highest weight edges for each node. The $\epsilon$-neighborhood model prunes all edges whose weights are below a threshold $\epsilon$. In \textsc{Casis}, we set $k = 5$ and $\epsilon$ equals to the median of all edge weights.

**Tag Concept Graph Constructor.** Finally, this component detects tag concepts from the \textsc{trg} using an existing graph-cut or clustering algorithm. Notably, this component is designed to seamlessly incorporate different clustering algorithms (both hard and soft clustering) with different configurations, allowing users to explore different possible concepts. \textsc{Casis} currently supports five clustering algorithms, namely, \textsc{Border Flow} ($bf$), \textsc{Chinese Whisper} ($cw$), \textsc{K-Nearest Neighbor} ($knn$), \textsc{Affinity Propagation} ($ap$), and \textsc{Markov Clustering} ($mcl$) (mainly based on the Cluster Visualization Kit\(^6\)). As it is unrealistic to predict the number of concepts in a search, we choose those clustering algorithms which do not require the number of pre-defined clusters. Among them, \textsc{bf}, \textsc{cw}, and \textsc{mcl} support soft-clustering enabling overlapping and subsumed tag concepts. Figures 5(b), 5(c), and 5(d), show the concepts detected for $Q_{\text{sea}}$ by \textsc{knn}, \textsc{mcl}, and \textsc{ap}, respectively. The concepts detected by \textsc{bf} are shown in Figure 4. Note that, the query tag (i.e., \textsc{sea}) is not shown in \textsc{trg} or \textsc{tcg} as it is related to all tags and all concepts.

For each detected concept, its tags are then used to refine the query to get the concept-specific results from the initial results, and then display them under a numbered tag (id) in the result panel. That is, \textsc{Casis} presents both the tags (and their relationships) defining each concept and the corresponding image results for the concept.

**5. REFERENCES**


---

\(^6\)http://borderflow.sourceforge.net/