PivotBrowser: A Tag-Space Image Searching Prototype

Xiaoyan Li, Lidan Shou, Gang Chen, Xiaolong Zhang, Tianlei Hu, and Jinxiang Dong
College of Computer Science, Zhejiang University
Hangzhou, P.R.China 310027
kricel_lee@yahoo.com.cn, {should,cg,xiaolongzhang,htl,djx}@zju.edu.cn

ABSTRACT

We propose a novel iterative searching and refining prototype for tagged images. This prototype, named PivotBrowser, captures semantically similar tag sets in a structure called pivot. By constructing a pivot for a textual query, PivotBrowser first selects candidate images possibly relevant to the query. The tags contained in these candidate images are then selected in terms of their tag relevances to the pivot. The shortlisted tags are clustered and one of the tag clusters is used to select the results from the candidate images. Ranking of the images in each partition is based on their relevance to the tag cluster. With the guidance of the tag clusters presented, a user is able to perform searching and iterative query refinement.

Categories and Subject Descriptors: H.4 [Information Systems Applications]: Miscellaneous

General Terms: Algorithms, Design

Keywords: Tag, Inconsistency, Ambiguity, Relevance

1. INTRODUCTION

Tagging based search systems are known to be prone to semantic errors or limitations [2]. To name a few: Different users may use different tags (maybe synonyms) to describe the same object, causing *inconsistency* in tagging; The existence of polysemy (single term having multiple meanings) in a query causes *ambiguity*, and the query is often hard to refine; The distribution of the tags being used is usually *skewed* and has the long-tail characteristic. Therefore, on one hand, images with rare tags cannot be easily found. On the other hand, queries with rare tags may need to be expanded to larger scopes. We propose PivotBrowser, an iterative searching and refining prototype for tagged images. PivotBrowser employs a novel tag-based structure called *pivot* to address the above problems. Our approach is different from a previous work on social tag clustering [1] as we can handle both synonymy and ambiguity.

2. THE PIVOT BROWSING SCHEME

To introduce the concept of pivot, we first give the definition of *tag atom* based on the availability of a *tag thesaurus*. The tag thesaurus contains lexical relevance information for all tags, such as the synonyms ("flower, bloom, blossom"), the spelling variations (plural, abbreviation, etc.), and the other highly relevant terms ("film" vs. "movie"). A good example of tag thesaurus is the one used in the WordNet[3]. A *tag atom* $\widehat{A}$ is a set of tags that satisfy the following requirements: (1) If a tag atom $\widehat{A}$ contains a tag t, it must also contain all lexically relevant tags of t as defined in the thesaurus; (2) For any two tags in $\widehat{A}$, t_1 and t_2, they must be lexically relevant to each other. It is important to note that one tag may possibly appear in multiple tag atoms as it can have more than one lexical meaning in the thesaurus. Therefore, given a universe of tags $\{t_i\}$, we can precompute an inverted list for all possible tag atoms based on the tag thesaurus. Each entry in the inverted list is like following

$$< t_i, \text{id of } \widehat{A}_{i,1}, \text{id of } \widehat{A}_{i,2}, \cdots >,$$

where each tag atom $\widehat{A}_{i,j}$ contains tag t_i. We refer to this inverted list as the Tag Atom Inverted List (TAIL).

A *pivot atom* of tag t_i, denoted as $PA(t_i)$, is defined as the union of all tag atoms which contain t_i (or those in the same entry of t_i in TAIL). A *pivot* of n tags, $P(t_1, t_2, \ldots, t_n)$, is defined as the set containing all pivot atoms of its tags,

$$P(t_1, t_2, \ldots, t_n) = \{PA(t_m) | m = 1, \ldots, n\}.$$

An n-tag set $\{t_{j_1}, t_{j_2}, \ldots, t_{j_n}\}$ is said to be *supported* by pivot $P(t_1, t_2, \ldots, t_n)$, if $t_{j_m} \in PA(t_m)$ $(m = 1, 2, \ldots, n)$. Based on these definitions, we can enumerate all tag sets supported by a pivot. These tag sets are supposed to be semantically similar to each other.

2.1 The precomputation

Before any user interaction, we need to precompute some data structures required during pivot browsing. (1) First, given the universe of all tags in the image database, we generate the TAIL based on the thesaurus. (2) Second, we generate an inverted index for the image database, where each entry contains an image ID list for a tag. If each image is regarded as a document, for each key (tag) of the inverted index, we check the length of its image ID list and compute the inverse document frequency (IDF) for each tag. (3) Third, we compute a tag-to-tag affinity matrix for all tags in the database. The tag affinity metric that we use is the well-known Jaccard coefficient over the entire image database:

$$\text{aff}(t_1, t_2) = \frac{|ImageSet(t_1) \bigcap ImageSet(t_2)|}{|ImageSet(t_1) \bigcup ImageSet(t_2)|},$$

where $ImageSet(t)$ indicates the set of images having tag t.

2.2 Pivot browsing

The interactive pivot browsing is an iterative process consisting of the following three phases:

(1) First, when a user issues a query Q containing tags $\{q_1, \ldots, q_n\}$, the system looks up the TAIL to find the tag atoms for each query tag q_i. By merging the tag atoms for each query tag, we obtain a pivot $P(Q) = \{PA(q_1), \ldots, PA(q_n)\}$. For each tag set Q' supported by $P(Q)$, we look up the inverted index of the image database to find images where all tags in Q' co-occur. These images are saved as a candidate image set I_{can}, and all tags associated with them are saved (except for the query tags in Q) as a candidate tag set T_{can} for further consideration in the subsequent phases.

(2) Second, all candidate tags in T_{can} will undergo a selection pass, and the top-K candidates relevant to $P(Q)$ will be obtained. Meanwhile, the relevance value of each tag is saved as its weight. The tag selection method will be discussed in section 2.3.

(3) Third, the K tags in the output of the previous phase will be clustered on the fly using a graph-partitioning algorithm as proposed in [4]. The affinity metric for clustering is based on the precomputed tag-to-tag affinity values. These K tags, grouped in their clusters, will then be presented to the user for a new round of tag selection/refinement. Meanwhile, one of the tag clusters (by default the most compact one) will be used to select the images in I_{can} – only candidates with tags which appear in the cluster are selected. Ranking of the output images is based on the relevance between the tag vector of each image and the weighted vector of the cluster. The latter can be obtained from the results of phase 2. A user can certainly choose another tag cluster for image selection and browsing. If a user subsequently adds a new tag to or removes an old one from the query, the pivot browsing process enters the next iteration (goto phase 1).

2.3 Selecting pivot-relevant tags

The top-K tags are selected from the candidate tag set T_{can} based on their relevance to the pivot. The relevance between a candidate tag t and the pivot $P(Q)$ is computed using the tagging statistics as following

$$rel(t, P(Q)) = co(t, P(Q)) \cdot IDF(t)$$

where $co(t, P(Q))$ is the number of co-occurrences of t and any tag set Q' supported by $P(Q)$, on the entire image database. As any image associated with Q' must appear in I_{can}, we can expedite the co-occurrence computation by computing the number of images having tag t in I_{can}. The IDF values can be obtained from the precomputation.

3. RESULTS AND CONCLUSION

We implement the PivotBrowser prototype, and evaluate its query performance on a dataset containing 523746 tagged images randomly downloaded from Flickr. The tag universe of the dataset contains 427482 unique tags. Figure 1 shows the PivotBrowser interface with the search results for query of tag "window". The top-right region presents the six clusters of the K tags relevant to the pivot. To give a few examples, $cluster_1 = \{view, airplane, condo \ldots\}$, $cluster_2 = \{store, fashion, display, shopping \ldots\}$, and $cluster_3 = \{white, green, red, nikon, canon, blue \ldots\}$. The images shown are the topmost results of $cluster_1$, which confirm the implied semantics of "sight-view from window" in the cluster.

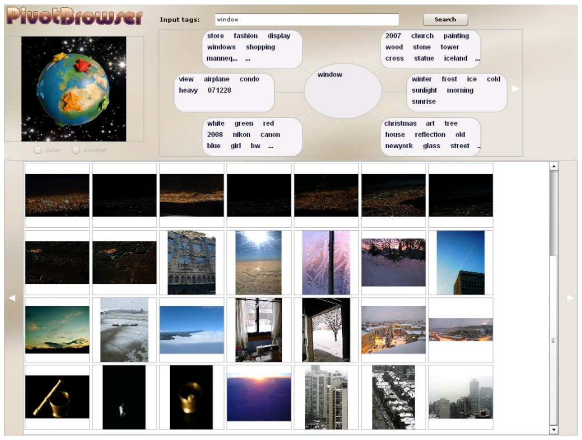
PivotBrowser
Input tags:
window
Search
store fashion display windows shopping manneq... ...
2007 church painting wood stone tower cross statue iceland ...
view airplane condo heavy 071228
window
winter frost ice cold sunlight morning sunrise
white green red 2008 nikon canon blue girl bw ...
christmas art tree house reflection old newyork glass street ...
color
wavelet

Figure 1: A Search Result Page for query "window"

We perform 200 unique queries on the prototype. Each query is executed for 100 times. Table 1 presents the average CPU time for selecting the candidate image set on the inverted index of the image database (SelectI), generating the top-K relevant tags (SelectT), clustering the K tags (ClusterT), and ranking the results (Rank). The time for creating a pivot is negligible. The results in the table reveal that the query cost is dominated by the selection on the inverted index of the image database.

	SelectI	SelectT	ClusterT	Rank
CPU Time (ms)	535.3	15.5	97.5	78.1

Table 1: Run-time CPU Cost in Each Phase

In conclusion, the pivot browsing scheme realizes effective query expansion and image searching in the tag-space at a low expense of computation and storage. Therefore, it can help users to find the intended results more effectively compared to conventional methods. We believe that pivot browsing can potentially become a general tag-space search paradigm not only limited to images.

For future work, we would conduct a usability study on PivotBrowser. We would also consider a comprehensive study on incorporating visual feature comparison, and other tag selection and clustering strategies into it.

4. REFERENCES

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

[2] S. A. Golder and B. A. Huberman. Usage patterns of collaborative tagging systems. *J. Inf. Sci.*, pages 198-208, 2006.

[3] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235-244, 1990.

[4] S. White and P. Smyth. A spectral clustering approach to finding communities in graphs. In *SDM*, 2005.