Clickage: Towards Bridging Semantic and Intent Gaps via Mining Click Logs of Search Engines

Xian-Sheng Hua, Linjun Yang, Jingdong Wang, Jing Wang
Ming Ye, Kuansan Wang, Yong Rui, Jin Li
Microsoft Corporation, One Microsoft Way, Redmond WA 98052, USA
{xshua; linjuny; jingdw; v-wangji; mingye; kuansanw; yongrui; jinl}@microsoft.com

ABSTRACT

The semantic gap between low-level visual features and high-level semantics has been investigated for decades but still remains a big challenge in multimedia. When "search" became one of the most frequently used applications, "intent gap", the gap between query expressions and users' search intents, emerged. Researchers have been focusing on three approaches to bridge the semantic and intent gaps: 1) developing more representative features, 2) exploiting better learning approaches or statistical models to represent the semantics, and 3) collecting more training data with better quality. However, it remains a challenge to close the gaps.

In this paper, we argue that the massive amount of click data from commercial search engines provides a data set that is unique in the bridging of the semantic and intent gap. Search engines generate millions of click data (a.k.a. image-query pairs), which provide almost "unlimited" yet strong connections between semantics and images, as well as connections between users' intents and queries. To study the intrinsic properties of click data and to investigate how to effectively leverage this huge amount of data to bridge semantic and intent gap is a promising direction to advance multimedia research. In the past, the primary obstacle is that there is no such dataset available to the public research community. This changes as Microsoft has released a new large-scale real-world image click data to public. This paper presents preliminary studies on the power of large-scale click data with a variety of experiments, such as building large-scale concept detectors, tag processing, search, definitive tag detection, intent analysis, etc., with the goal to inspire deeper researches based on this dataset.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing - *abstracting methods, dictionaries, indexing methods.* Information Search and Retrieval - *retrieval model, search process, relevance feedback.*

MM'13, October 21–25, 2013, Barcelona, Spain.

http://dx.doi.org/10.1145/2502081.2502100 .

General Terms

Algorithms, Experimentation, Human Factors

Keywords

Image search, image understanding, dataset

1. INTRODUCTION

It was estimated that there were more than 6 billion photos on Flickr, at least 690 million videos on YouTube and 220 billion or more photos on Facebook. Content-aware multimedia managing, understanding, searching and consuming is a key technology to deal with the explosion of images and videos available on the Internet, desktops and mobile devices.

However, visual content analytical technologies have not been extensively exploited in real-world applications such as tagging, searching, advertising and recommendation. The primary barrier is that current media content understanding technologies are still not mature enough to bridge the "semantic gap" (the gap between the low-level features and high-level semantics) and the "intent gap" (the gap between the representation of users' query/demand and the real intent of the users), in terms of accuracy, scalability and coverage. The technologies are still not there to effectively power real-world applications, especially when the data volume is large and data complexity is high [7, 12].

There are three possible directions to bridge these gaps: 1) to develop more representative visual features from the pixels in images or video frames, 2) to invent better learning approaches and/or statistical models to model and learn the semantics, and 3) to collecting more training data with better quality. Considerable efforts on the first two approaches have been made in the past decades (though the "gaps" remain). For the third, the key barrier is the availability of high quality and large scale dataset.

Though there are already quite a few labeled image datasets available to model and learn semantics for visual content, such as ImageNet [5], the coverage (in terms of both labels and images) of those datasets are far from being able to infer semantics for real-world applications. A good dataset for this purpose should not only contain large-scale data, but also reveals the challenges in solving real-world problems, enables creative researches to solve the challenges, and verifies the right solutions that are able to power real-world applications. We argue that the click data from commercial search engines meets these requirements.

Table 1: Comparison of different image datasets.

	# of images***	# of labels	Construction	Coverage	Ext.**	Acc.**	Avail.**	Seg.**
MSRC	Small: 591	23	Manual	Low: 23 object categories	Low	High	Yes	Yes
Caltech101	Small: 9K	101	Search + Manual	Low: 101 object categories	Low	High	Yes	No
PASCAL 2012	Small: 27K	20	Search + Manual	Low: 20 categories	Low	High	Yes	No
Caltech256	Small: 30k	256	Search + Manual	Low: 256 object categories	Low	High	Yes	No
Corel	Small: 30K	98	Manual	Low: 98 categories	Low	High	Yes	No
LabelMe	Small: 30K	183	Crowdsourcing	Low: 183 categories	Medium	High	Yes	Yes
ImageCLEF	Middle: 250K	-*	Search Engine	Medium: English Dictionary	High	Low	Yes	No
NUS-Wide	Middle: 270K	5,018	Social Media	Medium: 5K generic tags	High	Medium	Yes	No
ESP	Large: 10+M	-*	Crowdsourcing (ESP)	Medium: generic manual tags	Medium	High	No	No
TinyImage	Large: 80M	75K	Search Engine	Medium: 75K from WordNet	High	Low	Yes	No
ImageNet	Large: 14M	22K	Search + Crowdsourcing	Medium: 22K synset (WordNet)	Medium	High	Yes	No
Clickture	Large: 40M	73.6M	Search Log	High: Queries of Users	High	High	Yes	No

* Not available in public literature.

** Ext. = Extensibility; Acc. = Label Accuracy; Avail. = Public Availability; Seg. = Segmented。

*** Using the largest number that we can find in literature.

Click data has been studied for some time in Web search area. But to the best of our knowledge, only a few limited research that is using large-scale click image data to bridge semantic and intent gap can be found in literature [10]. In this paper, we will discuss why and how large-scale click data can be used to bridge semantic gap and intent gap in a variety of image applications. We call this direction "Clickage", which means the use of large-scale click data to build the linkage among visual content, semantics and search intents. In [9], a large-scale click based image dataset, Clickture, is introduced. All the data used in this paper can be accessed at a central website (http://www.clickture.info). Table 1 shows a summary of commonly used image datasets and Clickture. More details of the unique features, construction processes, and properties of Clickture can be found in [9]. This paper will focus on experiments on Clickture to explore what this large-scale click-based dataset can provide and to show that the idea of "Clickage" is a promising direction to advance content-aware multimedia research.

The main purpose of this paper is not to present the "best" solutions to the research problems discussed below, but to introduce baselines for a variety of existing and new research topics by using the Clickture dataset, as well as to inspire deeper and more sophisticated studies based on this new real-world image dataset.

We will briefly review the properties of the Clickture dataset in Section 2. For the next three sections (3 to 5), we present three sets of exemplary research problems on the dataset, including query modeling, graph-based image analysis, and query intent analysis. Extensive experimental results will be reported and a dedicated "discussion" sub-section is included in each section. We conclude the paper in Section 6 with a highlight on a few additional research directions using the Clickture dataset.

2. THE NEW DATASET: WHAT AND WHY

To make the paper self-contained, in this section, we briefly review what are included in the Clickture dataset and then summarize the unique properties of this dataset. As aforementioned, more details can be found in [9].

2.1 The Datasets

The dataset, named Clickture, was sampled from one-year click log of a commercial image search engine. It consists of a big table with 212.3 million triads:

$$Clickture = \{\langle K, Q, C\rangle\}. \quad (1)$$

Table 2: Basic numbers of Clickture-Full and Clickture-Lite.

	Full	Lite
# of images	40M	1M
# of image/query pairs	212.3M	23.1M
# of unique queries	73.6M	11.7M
# of unique terms	4.0M	915K
# of clicks	405M	82.3M
# of queries per image (average)	5.3	23.1
# of images per query (average)	2.9	2.0
# of clicks per image	10.1	82.3

A triad $\{\langle K, Q, C\rangle\}$ means that the image "K" was clicked "C" times in the search results of query "Q" in one year (maybe by different users at different times). Image K is represented by a unique "key" which is hash code generated from the image URL, together with the original URL. Query "Q" is a textual word or phrase, and click count C is an integer which is no less than one. One image may correspond with to one or more entries in the table. One query may also appear in multiple entries triads that are associated with different images. There are 40 million unique (in terms of URLs) image keys, that is, images in the dataset, and 73.6 million unique queries (based on textual string comparison in lower case) in the Clickture.

Through users' click action during When doing image search, a user often clicks one or more images that are relevant to the query, thus mostly the query Q in the triad is relevant linked to the image K. In general, the bigger the click count C is, the higher probability that the corresponding query is relevant to the image. For convenience, we call Q a "clicked query" of Image K, and K a "clicked image" of query Q, and call $\langle K, Q\rangle$ a "clicked image-query pair", and the triad $\langle K, Q, C\rangle$ as "click data". We also call "clicked queries" of an image as "labels" of the image.

To enable the use of Clickture by a wide range of research organizations and individuals with different computing, networking, storage and programing capacities, a subset of Clickture images (1 million images and 11.7 million queries), is provided. We call this set Clickture-Lite and the full $40M$ dataset Clickture-Full (or in brief Clickture). The $1M$ images in Clickture-Lite are randomly sampled from the $40M$ image dataset (based on click frequency). Table 2 shows the basic numbers of Clickture-Full/-Lite, in which "terms" means individual words in a query. For example, query *white dog* contains two terms: *white* and *dog*.

Figure 1 shows the distributions of query lengths (i.e., number of words in a query) of Clickture (for clarity of the figure, only the percentage numbers for the Full set are shown). It is observed that most queries have 3 or 4 words (around 53% for both sets) and more than 83% queries have

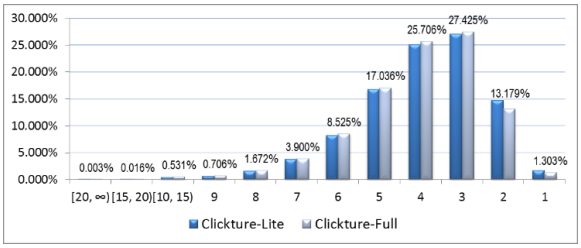


Figure 1: Query length distribution.

Table 3: Top 5 domains.

	Full	Lite
1	blogspot.com	blogspot.com
2	wordpress.com	wordpress.com
3	photobucket.com	photobucket.com
4	staticflickr.com	staticflickr.com
5	deviantart.net	fanpop.com

2 to 5 words. Less than 2% of the queries only have one word. Around 15% queries have six of more words, which maps to $1.7M$ and $11.3M$ queries in Clickture Lie and Full sets, respectively. These queries give us sufficient data to study the challenge of both long and short queries in image search.

Table 3 shows the top 5 domains where the image URLs come from. We can see that top domains are all from social media websites (including blog websites and photo sharing websites), which is accordant to the fact that social media now is receiving increasing attention from Internet users. At the same time, it also shows that Clickture provides sufficient data to facilitate researches on social media data.

2.2 The Advantages of Click Data

Most existing image datasets consist of limited number of images, which in general have relatively high accuracy but lower coverage in terms of both images and labels. Usually it only contains hundreds of object categories and is far from revealing the real-word complexities and meeting the needs of real-word applications, which needs to deal with millions of objects. Datasets based on social media sharing websites (for example, NUS-Wide [3], in which the user-input tags on Flickr were regarded as the labels of the images) typically contain big noises as well as the problem that the labels are often too generic because users generally tend not to spend too much time on labeling [5]. TinyImage [18] and ImageCLEF (the part for scalable image annotation) use the results returned by search engines directly [1], which are very noisy. ImageNet improved the label quality by using crowdsourcing labeling, which reaches 99.7% precision in average. ESP dataset [21] and LabelMe [19] are also constructed through crowdsourcing, one through ESP image labeling game and the other through volunteered online labeling. LabelMe only has 183 categories because manually labeling regions is still tedious even by leveraging online volunteers. Labels generated through ESP game are mostly "basic level" semantics as humans tend to label easily accessible semantics, although they are accurate. The labels in ImageNet are based on WordNet hierarchy, which have better coverage than ESP data. However, its overlap with the queries that people are using in daily search is very low. For example, it only covers 0.01% of the distinct queries in Clickture [9].

Figure 2: Random examples in click data

The primary shortcoming of existing image datasets is the low coverage of real-world semantic to reveal and verify the real-world challenges and complexities, thus the researches based on those datasets still have a large gap from being capable of powering real-world applications. On the contrary, Clickture was constructed by a totally different approach: using search engine click logs. As a byproduct of commercial image search engines, and with the increase of the image indexed as well as the number users that are using image search, Clickture can potentially grow both in amount of data collected and its coverage, and be extended unlimitedly. It also reflects common users' searching and consuming interests and covers the semantics (textual queries) that people desire to search in daily life. Labels in Clickture are more accurate than datasets that are using search results from search engines or user-input tags in Flickr, though less accurate than manual labeling. Figure 2 shows a few random exemplary images with top three clicked queries. We observe that the quality of the label is generally high, though noises can also be observed. Figure 3 shows the top 100 clicked images of query *chair*, which contains a variety of chairs with diverse appearances.

Figure 3: Top 100 clicked images of query "chair".

3. EXAMPLARY RESEARCH (I): QUERY MODELING

In this section, we will use Clickture as "labeled" data for semantic "concept" (a.k.a., "queries" in this paper) modeling and classification and then use the classification results to improve the ranking of image search. Throughout the section, for each image in the dataset, we use a 1000-dimensional feature vector, which includes color histogram, color wavelet, gradient histogram, face, etc.

3.1 Top Query Modeling

In this experiments, we evaluate statistical models learned from Clickture for queries with relatively larger number of clicked images. For Clickture-Lite, we filter out queries with less than 100 clicked images and form a "top query list", which contains 6891 queries. We also use this list for the Clickture-Full. Table 4 shows the distribution of the number of clicked images of these queries in the Full and Lite datasets. A few examples can be found in Figure 4 and 6.

Table 4: Number of clicked images for top queries.

# of images	# of queries (Lite)	# of queries (Full)
[100, 200)	4955	115
[200, 300)	1081	558
[300, 500)	596	1748
[500, 1000)	216	2695
[1000, infinity)	43	1775
Total	6891	6891

Table 5: Top query modeling performance on Clickture-Lite.

Model	Acc.	P. Prec.	P. Recall	N. Prec.	N. Recall	T. Train	T. Test
SVM (n=1)	0.7710	0.7690	0.7822	0.7818	0.7598	14.61	4.33
SVM (n=2)	0.7665	0.7633	0.7842	0.7823	0.7488	12.74	3.78
FRC (n=1)	**0.7846**	**0.7816**	**0.7931**	**0.7923**	**0.7760**	10.22	3.43
FRC (n=2)	0.7646	0.7610	0.7788	0.7729	0.7503	6.41	2.15
PM (n=1)	0.7670	0.7604	0.7833	0.7798	0.7507	2.21	0.74
PM (n=2)	0.7537	0.7480	0.7732	0.7707	0.7342	1.94	0.65
AP (n=1)	0.7367	0.7346	0.7539	0.7564	0.7195	1.13	0.61
AP (n=2)	0.7190	0.7184	0.7453	0.7482	0.6926	1.12	0.60
LR (n=1)	0.7734	0.7678	0.7869	0.7837	0.7598	3.27	0.24
LR (n=2)	0.7627	0.7572	0.7793	0.7764	0.7461	2.63	0.19

Acc.: Accuracy; P.: Positive; Prec.: Precision; T.: Time Cost (in machine-hour); FRC: Fast Rank Classification; PM: Parameter Mixer; AP: Averaged Perception; LR: Logistic Regression.

Table 6: Top query modeling performance on Clickture-Full.

Model	Acc.	P. Prec.	P. Recall	N. Prec.	N. Recall	T. Train	T. Test
SVM (n=1)	0.7811	0.7861	0.7897	0.7953	0.7722	22.72	6.73
SVM (n=2)	0.7875	0.7870	0.7993	0.7999	0.7752	17.83	5.28
FRC (n=1)	**0.8246**	**0.8215**	**0.8317**	**0.8299**	**0.8172**	21.14	7.10
FRC (n=2)	0.8124	0.8093	0.8216	0.8181	0.8028	15.15	5.09
PM (n=1)	0.8030	0.7966	0.8156	0.8115	0.7901	5.09	1.71
PM (n=2)	0.7920	0.7849	0.8080	0.8038	0.7757	3.35	1.13
AP (n=1)	0.7745	0.7700	0.7868	0.7859	0.7618	1.37	0.74
AP (n=2)	0.7632	0.7586	0.7825	0.7815	0.7434	1.26	0.68
LR (n=1)	0.8057	0.7997	0.8173	0.8134	0.7939	7.42	0.55
LR (n=2)	0.8094	0.8008	0.8259	0.8216	0.7925	5.25	0.41

Abbreviations are the same as Table 5.

In the first set of experiments, we regard clicked images with click count equal to or greater than n as "real" positive samples for the corresponding query in the top query list. The larger the n is, the more accurate the training data is. The tradeoff is that there are fewer training samples satisfying the accuracy criteria. We selected negative samples by randomly sampling from the rest of the dataset, with the images that do not contain the target query in the clicked query list.

(a) "backsplash designs"

(b) "bunk beds"

(c) "lifted trucks"

Figure 4: Top query modeling examples (upper row: training data; lower row: top predicted positive).

Five learning algorithms are applied on the top query dataset in Table 4, with 6891 top queries with their clicked images: Linear SVM [16], Fast Rank Classification (Boosting Decision Trees) [6], Parameter Mixer [15], Averaged Perception [4] and Logistic Regression [2]. Two-fold cross validation is used to evaluate the performance of the algorithms. Parameters of those algorithms are fine-tuned on a randomly sampled query set with 50 queries. All the learning and prediction are carried out on a low-end cluster with 12 machines (Quad-Core 2.1 GHz CPU, 8G RAM, Windows Server 2008).

Table 5 and 6 shows the performance numbers (accuracy, positive precision/recall, and negative precision/recall) of those five algorithms on Clickture-Lite and -Full respectively, including training and test time costs of each experiment.

From the two tables, we can see Fast Rank Classification perform the best in all metrics. Moreover, when is set to 1, the results are almost always better than those when $n = 2$, which shows that more data generates better results though at the same time more noises are brought in.

Figure 4 shows the randomly selected three queries with accuracies larger than 0.95, on the Clickture-Full dataset when $n = 1$, using the best learning algorithm in Table 6 (i.e., Fast Rank). The top ten positive training samples (according to click counts) are shown in the first row and the top ten images from the test part which have the highest prediction confidence (though may not be correct) are shown in the second row.

3.2 Image Annotation by Query Modeling

In the above experiment, we used clicked images as both training data and ground-truth for evaluation. In this section, we apply the models learned in Section 3.1 to annotate a different set of images.

From the top 6891 queries, we randomly select 100 queries with accuracy higher than 0.90, and then for each query model, top 50 search results of the corresponding query from a search engine are used for evaluation. The 100 best classifiers (from the aforementioned five algorithms, retrained using all the click data) are applied on 5K images. The accuracy, positive precision/recall, and negative precision/recall are 0.8412, 0.8292, 0.8590, 0.8541 and 0.8233, respectively. Figure 5 shows a few prediction results.

Figure 5: Examples of image annotation by top query modeling. The query is "cars". Images with blue, green and orange borders mean: true positive, true negative and false negative, respectively. No false positive is found in this example query.

3.3 Rank Learning

We investigate how Clickture can be used to improve image search ranking. The development set in the MSR-Bing Image Retrieval Challenge 2013 [9] is used for testing, which contains around $80K$ image-query pairs. The task is to estimate the relevance of the image and the query for each query-image pair, and then for each query, we order the images based on the prediction scores returned by the trained classifiers.

As in [9], Discounted Cumulated Gain (DCG) [11] is applied to measure ranking results against the manually labeled ground-truth,

$$DCG_p = \alpha_p \sum_{k=0}^{p} \frac{2^{rel_k} - 1}{\log_2(k+1)} \quad (2)$$

where $p = 25$, $\alpha_p = 0.01757$, rel_k is graded relevance score of the result in position k in the labeled dataset, and $rel_k = \{Excellent = 3; Good = 2; Bad = 0\}$. And last, the average of the DCGs on all the queries is the final evaluation result.

In this experiment, we discard queries in the development set that do not intersect in the Clickture. In total 352 and 546 queries intersect with Clickture-Lite and -Full, respectively. We randomly select 100 shared queries from these two intersections for evaluation. Fast Rank Regression [6] is used to learn a ranking function, and all the clicked images of the selected queries are used as training data. Two settings were tried: Setting 1: regarding the ranking label of all clicked images as "relevant" and randomly selected negative samples as "irrelevant"; Setting 2: regarding images with click count greater, equal, less than m as "Excellent", "Good", and "Average", respectively and negative samples as irrelevant (m is set to 3 in this experiment).

For Setting 1, DCG numbers on the Full and Lite sets are 0.474 and 0.482, respectively. The DCGs increased to 0.483 and 0.493 for Setting 2, which shows more training data generates better results, and grading click counts also further improve the DCG numbers. In Section 5, we will see that query intent estimation from click data will improve these results.

3.4 Discussion

3.4.1 Applications of Query Modeling

Experiments in this section showed promising results on annotating images with queries. When the number of queries (labels) is small, the annotations cannot improve the search quality in any significant way due to only a small portion of the images in the index will be impacted by those classifiers, as well as that sufficient number of Web images in general already have textual meta-data that are relevant to those categories. However, those small number of classifiers may significantly assist personal photo search because personal photos in general are lack of text descriptions. In Section 5, we will show that small-scale classification helps general web image search when the categories are sufficiently broad and when they are combined with query intent estimation.

3.4.2 Large-Scale Query Modeling

In 3.3, we only evaluated queries that can be found in the training datasets, and only trained classifiers for those top queries. In real search systems, this will not improve the search quality too much as it cannot cover queries that are not in the top query list. Two possible approaches can be applied: query modeling and n-gram modeling. The most difficult aspect here is handling the scalability issue, that is, training large-number of classifiers efficiently and predicting labels over large-number of labels in real-time [9, 8].

3.4.3 Completeness of Clicked Queries

The primary purpose of "query modeling" is to use click data to connect image content with semantics. The advantage of using click data is that it conveys a very wide range of "semantics" through a variety of "queries". Though those "queries" are not a "complete" list of textual descriptions of the image content, we believe it covers most "semantics" that average users care in their daily search. At the same time, the study of the completeness of the clicked query list of an image is a separate research topic which is very interesting and challenging.

3.4.4 Modelability of Queries

We first tried to model "query" with large number of clicked images and also discussed possible approaches to model every queries and n-grams with clicks. However, not every query can be modeled through the low-level features that we extracted and the statistical models that we choose. For example, the 10 queries with the worst accuracy (less than 0.61) under the best experimental setting in Table 6 are: *images quotes, crazy, s* (letter "s"), *image, cool stuff, pictures of stuff, image of art, funny animated pictures, fun,* and *Chinese.*

Figure 6 shows a few exemplar clicked images for some of these queries. We can hardly find any "patterns" to model. It is also difficult to design specific features to model them. We plan to investigate on identification of those tough to model queries, and investigate their feature discovery and selection in a future work.

A large number of queries in Clickture only have very limited number of images, which are very difficult to be well modeled. We plan to investigate the modelling of such query by leveraging other semantically-similar queries and returned images in our future work as well.

3.4.5 Handling Noises

Though clicked queries are in general relevant to the corresponding image, noises exist in Clickture. Some images may be clicked by mistake as they attracted users' attention for whatever reason (for example, very unique or strange images, even though they are not relevant to the current query). Clicked labels may also contain typos. As search engines typically are tolerant to "typos", they show indexed im-

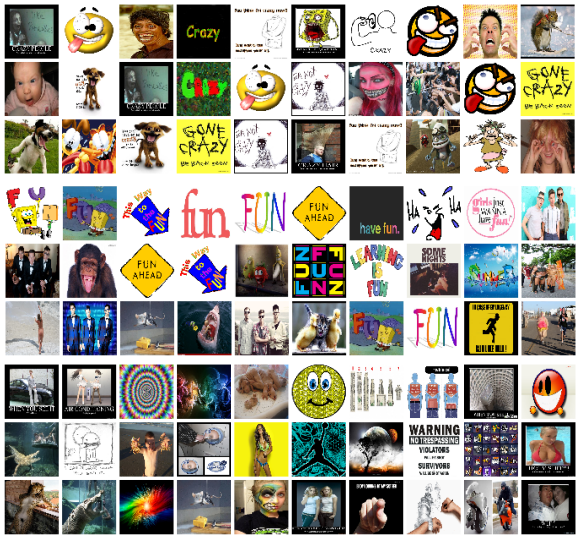


Figure 6: Queries that are difficult to model (top: "crazy"; middle: "fun"; bottom: "cool stuff").

ages of the (automatically) corrected query to users. Therefore, some clicked images may be associated with queries with typos. The query count is a good indicator for the confidence or relevance of the label to the image. By grouping multiple visually duplicate and/or similar images into a cluster, it is possible to reduce the noises in the clicked queries. Reduction of noise will be discussed more in Section 4.

3.4.6 *Leveraging Correlations among Queries*

Quite a few similar clicked queries will appear for "popular" images (images with a large number of clicked queries). For example, "cat", "cats", "cat picture", "image of cats", "image of cat", "cats images", "kitten", "image of kitten", etc., may be clicked queries of the same image. A preprocessing step of removing non-essential phrases, such as "picture", "image", "image of", and "picture of" from the query labels, will combine clicked images of semantically-equal queries into one single set.

In addition, if the "correlations" of queries can be discovered, for example, if we know "labradoodle dog" is a kind of "dog", and "dog" is a kind of animal, the clicked images of those queries can be leveraged upon each other to enhance query modeling. WordNet provides such an ontology. The shortcoming is that it only covers a limited number of queries. Researchers have also found ways to automatically create term ontologies, such as Needle Seek [27], which can be applied here.

4. EXAMPLARY RESEARCH (II): GRAPH-BASED IMAGE UNDERSTANDING

In this section, we will build a similarity graph on the 40 million Clickture-Full dataset and then use it to do "tag processing", image search and image annotation. Experimental results and discussion remarks will be presented at the end of this section.

4.1 Building Image Graph and Image Index

We represent each image with a 768-dimensional float vector. Mixture of Gaussian models are used to generate 12 representative colors in the color space [22], and for each RGB color, we compute a probability histogram indicating its similarity to the 12 colors. For each image, we divide it into a 9×9 grid. A color histogram is generated for each 2×2 cells. As a result, we get 64 histograms, from which a 768-dimensional feature vector is produced by concatenation.

As in [23], a hierarchical partition tree partitions the whole data set into one million clusters, and then a similarity graph is built for the one million cluster centers. Each cluster center is assigned with a unique ID and then regarded as a "visual word" to describe the image. We build up an inverted index structure that maps each cluster ID (a.k.a. visual word) to its member images in Clickture-Full.

For each image in Clickture-Full, we then search over the inverted index to obtain the nearest 100 images to the query image as the initial similar images. Finally, we re-rank those images using the Euclidean distance between the original features of the images and the query image [23]. By this way, we have generated a 100-NN graph on the $40M$ Clickture-Full dataset.

4.2 Tag Processing

Here "tag" means a clicked query of an image in Clickture-Full dataset. As aforementioned, noises exist in the dataset and the clicked queries are not complete, tag processing will reduce the noises as well as adding more tags to the images.

4.2.1 *Tag Ranking*

Tag ranking is to rank the tags of an image according to their relevance or importance to the image [14, 13]. Different from the work in [14, 13], tag ranking here not only ranks existing tags of the image, but also enriches tags from its neighbors, and this can be done in a unified way, as below.

Suppose the nearest neighbor list of query image x_0 is $\{\langle x_i, d_i\rangle, 0 \le i \le m\}$, among which x_i is an image, d_i is the distance of x_i to the query image, with $d_0 = 0$. Then a tag t's weight to x_0 is

$$w_t = \sum_{i=0}^{m} ClickCount(x_i, t) \cdot e^{-\alpha d_i} \quad (3)$$

where $ClickCount(x_i, t)$ is the click number of image x_i under query t, and α is a parameter to adjust the importance of image distance.

In equation (3), click count is used to indicate the confidence of a tag. To avoid to be dominated by images with very high click counts, we clip click counts to T ($T = 25$ in this paper). At the same time, to suppress the possible noises in neighbors, when computing the weight of each tag, we only use the nearest M candidates that have that tag. For each image, besides re-weighting the existing tags of the target image, we enrich at most 5 tags from its neighbors (M is set to 20 here).

4.2.2 *Definitive Tag Generation*

In many scenarios, it is beneficial to generate one most representative tag for the image. We call this tag the *definitive* tag of the image. For example, in "search by image" scenario, a "best guess" of the user uploaded image is to show the definitive tag. Moreover, "search by image" can then search over the image database using both the definitive tag and the query image's visual features, which will lead to improved quality of the search results comparing with searching by only visual features.

Based on "tag ranking" in 4.2.1, the top one tag of the image can be regarded as "definitive tag". A more sophisticated approach can take into account the balance of "generality" and "specificity" of tags to find a more representative tag. For example, "dog" is a correct tag for an image of a labradoodle dog, but "labradoodle dog" is more specific thus a more informative tag than the generic tag "dog".

4.3 Annotation by Nearest Neighbor Search

After tag processing over the graph, the tag quality of the images in Clickture is greatly improved. Then, can we use this "graph" to annotate images that are not in Clickture? In Section 3, we have shown that this can be done by building statistical models for queries and then applying the learned models on the image to get annotations. However, when we have built a large image graph with quality tags, a different approach can be employed.

Actually, this can be done easily by using the same approach of tag processing, where the difference is that the query image does not have any initial tag. First, we extract visual words using the approach mentioned in Section 4.1 from the "out-of-database" image, then get a list of nearest neighbors from the inverted index. And last, use equation (3) to get the tags for the target image.

4.4 Similar Image Search

To search for similar image of a given image, we first find its nearest neighbors based on only image features (that is, visual words matching through the inverted index and then re-ranking based on the original image features). This is the baseline results of similar image search.

4.4.1 Re-Ranking by Tag Filtering

When the query image is in the database and it has some tags, we can improve the similar image search result simply by applying a tag filter. After the baseline image list is obtained, we check each of the image on whether it shares a common tag with the query image. We move all images that do not share a common tag with the query to the end of the image list.

4.4.2 Re-Ranking by Tag Ranking

Using the equation of tag ranking, we can re-rank the baseline "similar image list" by

$$Sim(x_i) = e^{-\alpha d_i} \sum_{j=1}^{k} w_{t_j} \cdot ClickCount(x_i, t_j) \qquad (4)$$

where $w_(t_j)$ is the weight of tag t_j to image x_i, $1 \le j \le K$. In (4), both click counts and tag ranks/weights are applied to re-rank the baseline similar image list.

4.5 Definitive Image Generation

For a given "tag" or query, what are its most representative images? After tag ranking, the tag weight measures the relevance of a tag/query to an image. For all images that contains a certain tag, we can sort the image by the tag weight to rank images and identify the most representative images.

More sophisticated approaches can take the balance of diversity, visual quality, representativeness, freshness and completeness into account in ranking the images and generating representative images.

4.6 Experimental Results

We randomly sampled 1000 images from Clickture-Full dataset to evaluate the approaches presented in this section. In case the query image should not be included in Clickture (a.k.a. "out-of-database" images) in an experiment (for example, annotation by NN search, definitive tag for images that are not in Clickture, etc.), to save labeling cost, we use the same 1000 images but remove their tags and at the same time remove them from Clickture.

Table 7 lists the average numbers of tags or accuracy of image search with the following definitions:

- *Number of Original Tags*: number of tags with the query image
- *Number of Original Useful Tags*: number of tags that are strongly connected to the query, labeled manually
- *Number of Enriched Tags*: number of tags got from the candidate neighbors. Always 5 in this experiments
- *Number of Enriched Useful Tags*: number of tags got from the candidate neighbors that are strongly connected to the query
- *Number of Good Definitive Tag*: for each image, we regarded the tag with the largest weight as the definitive tag, and count how much percentage of the primary tag can well represent the sematic content of the image
- *Rate of Baseline Relevant Images*: number of the image with same sematic content in the first 20 images after ranking by only image features
- *Rate of Relevant Image After Tag Filtering*: number of the image with same sematic content in the first 20 images after tag filtering
- *Rate of Relevant Image After Tag Ranking*: number of the image with same sematic content in the first 20 images after refinement

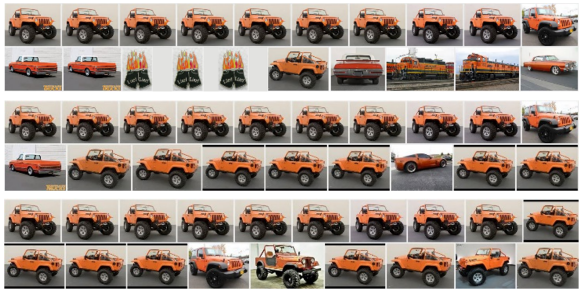
From the table, on average, tag ranking adds 1.18 strong relevant tags for in-database images. The number increased to 2.04 for out-of-database images. The reason is that the in-database images already contain 1.89 strongly relevant tags (on average) so there is less room to add more high-quality tags.

Table 7: Tag precessing evaluation.

#/Rates	In Database	Out of Database
Number of Original Tags	4.8	n/a
Number of Original Useful Tags	1.89	n/a
Number of Enriched Tags	5	5
Number of Enriched Useful Tags	1.18	2.04
Number of Good Definitive Tag	0.91	0.89
Rate of Baseline Relevant Images	0.3945	0.3945
Rate of Relevant Image After Tag Filtering	0.5145	n/a
Rate of Relevant Image After Tag Ranking	0.6110	0.5905

It is also observed that, for in-database images, tag filtering improves the relevance 12 points compared with the baseline (in terms of percentage of relevant images in top 20 returned results), and tag ranking improves more than 21 points. For out-of-database image, tag ranking improves 19.6 points (tag filtering will not work in this case). Figure 7 and 8 shows a few exemplar results of in-database and out-of-database image queries, respectively. The third result in Figure 7 is an example with worse result after tag ranking. The reason is that a few images about music notes have a

(a) Ranked High Confidence Tags: *Cocker, Spaniel, Puppies*



(b) Ranked High Confidence Tags: *Jeep, Jeeps, Orange*

(c) 'Ranked High Confidence Tags: *Shoes, Music, Notes*

Figure 7: Three In-Database Example. Each example has three sets of results shown and (top-left image is the query image): top row: baseline search results; middle row: re-ranking by tag filtering; bottom row: re-ranking by tag ranking. The ranked high confidence tags are shown under each result set and the first one is the detected "definitive tag".

very similar shape with a shoe, thus the tags of music notes are wrongly propagated to the shoe image. Figure 9 shows more results of definitive tag detection.

To evaluate definitive image detection, we randomly sampled 1000 queries and compare the top one definitive images for the following two settings: (1) based on original click counts only; (2) based on ranks after tag ranking. Table 8 shows the evaluation results and Figure 10 shows a few examples.

4.7 Discussion

It is possible to use different features and distance metrics to build graphs. Different tags (clicked queries) may be more effectively and precisely propagated on different features and/or distance metrics.

After tag filtering and refinement over graphs, the noise in the clicked log reduces. We may be able to build better statistical models upon them.

Building sub-graphs on certain domains is another idea to more effectively using the correlations among images. Images within a specific domain, for example, people, landmarks, animals, etc., generally have denser and more trustable connections.

Going one step further, connecting image graph with web page graph (which is constructed through in-links and out-links on the web pages), and query/label ontologies is another promising direction to infer more accurate semantics.

(a) Definitive Tag is "House Family". Row 1: baseline search results. Row 2: search results after tag ranking. The leftmost one is the query image (the same for the other examples).

(b) Definitive Tag: "AR Rifles Gun"

(c) Definitive Tag: "Watch Spiral Rolex"

Figure 8: Three out-of-database examples. For each example, first row: baseline results; second row: re-ranking by tag ranking. The first image of each result is the query image.

Table 8: Definivtive image evaluation.

	Average Precision
Using Click Counts	0.8020
Using Ranks after Tag Ranking	0.9300

This may blur the boundary between image semantic understanding and image search, as they are leveraging each other to achieve better understanding/search performance.

5. EXAMPLARY RESEARCH (III): QUERY INTENT ANALYSIS

Preliminary intent analysis has been reported in [26]. Besides revealing the connections between image content and semantics, click data also reflect users' query intents. Through image categorization and click data analysis, we may be able to discover users' search intent of a query.

Search intent of a query (or "query intent" in brief) can be described by categories in different levels. For example, in a relatively higher level, we may say the search intent of

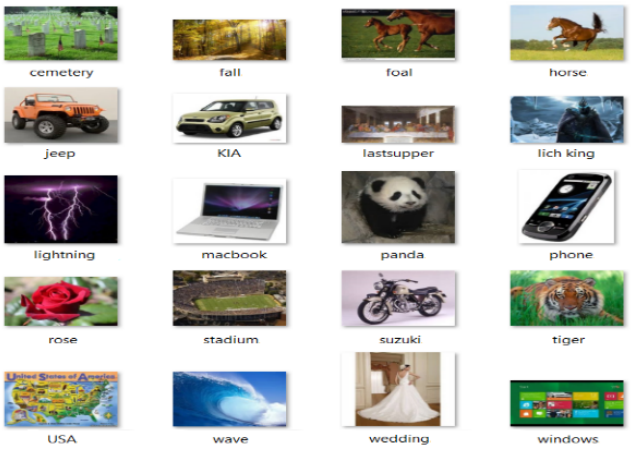


Figure 9: Example of definitive tag detection.

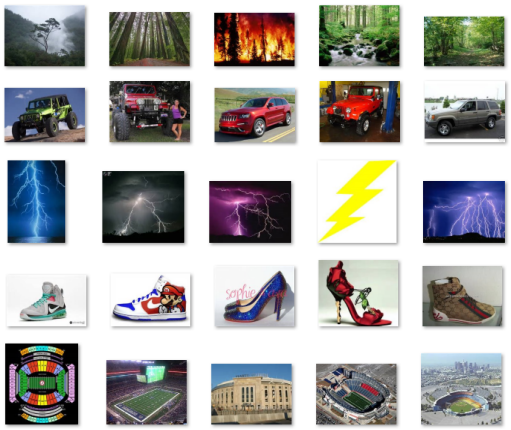


Figure 10: Examples of definitive images (top five in order). From top to bottom: forest, jeep, lighting, shoes, and stadium.

"Canon 60D" is "searching images of a product", while in a relatively lower level, the intent can be described as "searching images of a camera", or "searching images captured by cameras of a certain model". If the intent of a query is known, the search results can be ranked better accordingly.

Query intents can also be "implicit", which we can also call it query "types". For example, for an image query of "World War II", most people would like to see real World War II pictures, instead of pictures in a movie about World War II. As all real World War II pictures are mostly in black and white, then "black and white" can be an implicit or hidden "query intent". This kind of "type/intent" categorization can also help ranking. For example, we will not rank color images high for query "World War II" when we know the query intent of this query is "black and white".

One thing we should point out is that "query intents" are not necessarily exclusive to each other. For example, query "World War II Criminals" can be classified as "black and white" and at the same time "people".

After define a set of "query intent" categories, search intent of a query can be learned from labeled data. Alternatively, if we can classify the images into corresponding image categories, users' search intent can be mined from click data. For example, if we can detect whether an image contains people's face, we can classify whether a query's intent is looking for people or not.

5.1 Query Intent Estimation

In this experiment, we estimate a set of implicit or explicit query intents that the corresponding image type classification approaches are relatively mature, which include 11 non-exclusive categories: *black & white, portrait photo, clipart, high visual quality, line drawing, indoor, outdoor, cityscape, landscape, human full body,* and *head & shoulder*. Exemplary approaches to extracting those features can be found in [17, 24, 20, 25]. These classifiers were built on separate annotated datasets. The output of each classifier is either 1 (positive) or -1 (negative), or 0 if cannot be categorized into either of them.

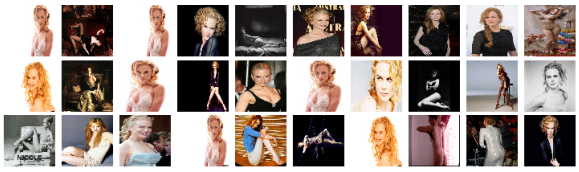
To estimate query intent, we first apply image classifiers on all the images in the Clickture datasets to get images types. For a certain query q, suppose its clicked images are $\{\langle x_i, C_i, f_i\rangle, 1 \le i \le n\}$, where x_i is the image, C_i is the corresponding click count under this query, and f_i is the output of an image classifier of type t. Then the correspondence between query q and image type t can be estimated as

$$R(q,t) = \sum_{i=1}^{n} \frac{g(C_i) f_i}{g(C_i)} \tag{5}$$

where $g(C_i)$ is a weighting function to adjust the importance of click counts. In our experiments, we empirically set $g(C_i)$ to $\sqrt{C_i}$. Then the correspondence between q and t is formulated by

$$R(q,t) \begin{cases} \ge \delta & Positive \\ \in (-\delta, \delta) & Nocorrespondence \\ \le -\delta & Negative \end{cases} \tag{6}$$

Figure 11 shows a few examples of clicked images for a few exemplar queries, together with the estimated query types.



(a) Query: "Nicole Kidman". Positive Intent/Image Type: Portrait, High Visual Quality, Human Full Body, Head and Shoulder. Negative Intent/Image Type: Clipart, Line Drawing.

(b) Query: "Seattle". Positive Intent/Image Type: Outdoor, Cityscape, High Visual Quality. Negative Intent/Image Type: Clipart, Line Drawing, Human Full Body, Black & White.

(c) Query: "Word War II". Positive Intent/Image Type: Outdoor, Back '& White. Negative Intent/Image Type: High Visual Quality, Clipart, Human Full Body.

Figure 11: Examples of detected query intent types.

5.2 Search Re-Ranking by Query Intent

To verify whether query intent categorization help relevance ranking, we apply a re-ranking step on the results obtained in Section 3.3. Suppose the relevance prediction of image I and query q is r, the image type vector is $\{t_i\}$ and the query type vector is $\{q_i\}$. Then the updated relevance prediction of I and q is

$$r' = ReRank(r, \sum t_i q_i) \tag{7}$$

where $ReRank(\cdot)$ is a re-ranking function combining the original relevance prediction with the correspondence between image types and query types. Different re-ranking functions can be designed. In our experiment, we set the re-ranking function as

$$ReRank(r,s) = r + \frac{\alpha s}{\|\{t_i\}\|} \tag{8}$$

where α is a balancing weight. After applying query intent based re-ranking, the DCG on the 100 selected queries used in Section 3.3 is increased to 0.494 on Clickture-Lite and 0.505 on Clickture-Full (under Setting 2), where we can see obvious improvements (1.1 and 1.2 points, respectively) compared with the original DCG results.

5.3 Discussion

This section demonstrates only a preliminary query intent analysis and its application on search re-ranking. There is still rooms to exploit along this direction.

Similar to generating ontologies from clicked queries that we discussed in Section 3, ontologies for query intent is also important for query understanding. In the above exemplary study, we generate a few query intents/types manually. To support large-scale query intent analysis, automatic approach is desired. In addition, building connections between query ontologies and intent ontologies is another topic to be explored.

The query intents of a specific query can be complex. Different people may have different intents when a query is issued. Even the same person may have different intents at different time with the same query. For example, "Oscar" could be a person's name, or the academy awards, or a film, etc. When a person queries this word during the Oscars ceremony time in 2013, then the intent mostly could be the pictures of the ceremony and pictures of awarded actors and actresses in Oscar 2013. Studying and modeling the shifting and revolution of query intents through click data analysis can be another interesting research topic.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we show how Clickage can advance multimedia research by doing extensive experiments on a number of existing and new research problems based on a click-based dataset. We argue that the growth of click data has the potentials of bridging semantic and intent gap in search and search related applications, and makes it highly valuable to advance the state-of-the-art of multimedia research.

Besides the ideas we presented in the "discussion" part of Section 3, 4 and 5, a variety of other research topics can be carried out based on the Clickage idea, for example,

Distance Metric Learning: Clicked queries provide large-scale additional and powerful hints to build a better distance metric between images.

Image Clustering: Clicked queries can be used as features to facilitate image clustering. The distance metric learned through clicked data can also be applied to aid image clustering.

Visualization and Browsing: Through semantic learned from Clickture, we may be able to build better methods to visualize and/or browse of search results and/or the entire image corpus.

Other Image Related Applications: Such as image advertising, image recommendation, image completion, super-resolution, compression, etc.

In addition, researchers may also use click data to study whether a query, label or tag is learnable, or how difficult to learn a specific label or a specific type of labels. New image visual features may also be invented to model different types of labels. Label relationship or ontology (for example, label graph or label hierarchy) may also be built based on the click data.

7. REFERENCES

[1] Image retrieval in clef, http://imageclef.org/2012/photo-web.

[2] G. Andrew and J. Gao. Scalable training of l1-regularized log-linear models. In *ICML*, 2007.

[3] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. Nus-wide: a real-world web image database from national university of singapore. In *CIVR*, 2009.

[4] M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. pages 1–8, 2002.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[6] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[7] A. Hanjalic, R. Lienhart, W. Y. Ma, and J. R. Smith. The Holy Grail of Multimedia Information Retrieval: So Close or Yet So Far Away? *Proceedings of the IEEE*, 96(4):541–547, Apr. 2008.

[8] X.-S. Hua. Looking into "msr-bing image retrieval challenge". In *Microsoft Research Technical Report MSR-TR-2013-76*, April 2013.

[9] X.-S. Hua and et al. Clickture: A large-scale real-world image dataset. In *Microsoft Research Technical Report MSR-TR-2013-75*, August 2013.

[10] V. Jain and M. Varma. Learning to re-rank: Query-dependent image re-ranking using click data. In *Proceedings of the International World Wide Web Conference*, March 2011.

[11] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[12] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *TOMCCAP*, 2(1):1–19, 2006.

[13] X. Li, C. G. M. Snoek, and M. Worring. Learning social tag relevance by neighbor voting. *IEEE Transactions on Multimedia*, 11(7):1310–1322, 2009.

[14] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang. Tag ranking. In *WWW*, pages 351–360, 2009.

[15] G. Mann, R. T. McDonald, M. Mohri, N. Silberman, and D. Walker. Efficient large-scale distributed training of conditional maximum entropy models. In *NIPS*, pages 1231–1239, 2009.

[16] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: primal estimated sub-gradient solver for svm. *Math. Program.*, 127(1):3–30, 2011.

[17] M. Szummer and R. W. Picard. Indoor-outdoor image classification. In *CAIVD*, pages 42–51, 1998.

[18] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(11):1958–1970, 2008.

[19] A. Torralba, B. C. Russell, and J. Yuen. Labelme: Online image annotation and applications. *Proceedings of the IEEE*, 98(8):1467–1484, 2010.

[20] P. A. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR (1)*, pages 511–518, 2001.

[21] L. von Ahn and L. Dabbish. Esp: Labeling images with a computer game. In *AAAI Spring Symposium: Knowledge Collection from Volunteer Contributors*, pages 91–98, 2005.

[22] J. Wang, J. Wang, X.-S. Hua, and S. Li. Scalable similar image search by joint indices. In *ACM Multimedia*, pages 1325–1326, 2012.

[23] J. Wang, J. Wang, G. Zeng, Z. Tu, R. Gan, and S. Li. Scalable k-nn graph construction for visual descriptors. In *CVPR*, pages 1106–1113, 2012.

[24] R. Yan, Y. Liu, R. Jin, and A. Hauptmann. On predicting rare classes with svm ensembles in scene classification. In *ICASSP*, pages 21–24, 2003.

[25] Z. Yang, M. Li, and H. Ai. An experimental study on automatic face gender classification. In *ICPR (3)*, pages 1099–1102, 2006.

[26] Z.-J. Zha, L. Yang, T. Mei, M. Wang, Z. Wang, T.-S. Chua, and X.-S. Hua. Visual query suggestion: Towards capturing user intent in internet image search. *TOMCCAP*, 6(3), 2010.

[27] F. Zhang, S. Shi, J. Liu, S. Sun, and C.-Y. Lin. Nonlinear evidence fusion and propagation for hyponymy relation mining. In *ACL*, pages 1159–1168, 2011.