Cross Domain Search by Exploiting Wikipedia

Chen Liu #1, Sai Wu #2, Shouxu Jiang *3, Anthony K. H. Tung #4

# *School of Computing, National University of Singapore*
13 Computing Drive, 117417, Singapore
{1 liuchen, 2 wusai, 4 atung}@comp.nus.edu.sg

* *Institute of Computer Science and Technology,*
Harbin Institute of Technology, Harbin, China
3 jsx@hit.edu.cn

Abstract—**The abundance of Web 2.0 resources in various media formats calls for better resource integration to enrich user experience. This naturally leads to a new cross domain resource search requirement, in which a query is a resource in one modal and the results are closely related resources in other modalities. With cross domain search, we can better exploit existing resources.**

Intuitively, tags associated with Web 2.0 resources are a straightforward medium to link resources with different modality together. However, tagging is by nature an ad hoc activity. They often contain noises and are affected by the subjective inclination of the tagger. Consequently, linking resources simply by tags will not be reliable. In this paper, we propose an approach for linking tagged resources to concepts extracted from Wikipedia, which has become a fairly reliable reference over the last few years. Compared to the tags, the concepts are therefore of higher quality. We develop effective methods for cross-modal search based on the concepts associated with resources. Extensive experiments were conducted, and the results show that our solution achieves good performance.

I. Introduction

We have witnessed the phenomenal success of Web 2.0 over the last few years, which has enabled users to create and exchange self-organized resources on the web. The wide acceptance of Web 2.0 has resulted in a humongous amount of resources in folksonomy systems such as Flickr[1], Delicious[2] and Youtube[3]. For example, as of October 2009, Flickr hosted more than 4 billion images with manual tags from its users. However, current works are more focused on investigating each individual system while it will be much more useful if these systems could be cross supported. For example, while a user browses in Flickr, high probably he may for a plain document, if we can automatically assign Flickr images to the document, it would be more attractive for the readers. On the other hand, if we can find blogs/comments for an image, the user can have a better idea of the stories behind the image. This kind of service can enrich users experience significantly. Besides, with the wide spread use of mobile devices, there is an increasing demand of cross media search where these systems together provide a highly valuable database to support.

Unfortunately, the integration of Web 2.0 systems is hard to implement as most current systems do not support cross reference to related resources of each other. The difficulty of establishing such cross system links is how to define a proper mapping function for the resources of different domains. For instance, given an image from Flickr, we need a function to measure its similarities to the documents in Delicious or the videos in Youtube.

The problem of building connections between images and text has been well-studied [1][2][3][4] in the past. However, these methods always incur high computation overhead and involve complex learning algorithms. Most of the existing schemes are not flexible and cannot be extended to support an arbitrary domain. Namely, they are limited to the image and document formats. If a new Web 2.0 system emerges, the scheme needs to be fully redesigned to support it, which is not scalable for the real system.

To address the above problems, in this paper, we propose a uniform **Cross Domain Search** (CDS) framework which enables users to seamlessly explore and exploit huge amount of resources distributed across different Web 2.0 systems. In this framework, the user can input queries for any specific domain (e.g., keyword queries, image queries or audio queries) and the system will return the corresponding results from all related domains. The intuition is to transparently transform the submitted queries into a proper format for each target domain, where the candidate results are returned.

Intuitively, the metadata(tags) which are widely utilized in current Web 2.0 systems provides a potential solution of addressing the problem. In [3], they model the images and text in a keyword vector. However, due to the inherent problems in tagging[5], they are not capable of precisely capturing the semantics of resources. This will degrade the performance. Therefore, in this paper, we use Wikipedia as our knowledge base to recover such semantics. Wikipedia contains large number of concepts which are well described by their articles. As of July 24, 2008, Wikipedia has more than 260 million pages which covers most of the known topics. Compared to the personal tags used in Web 2.0 systems, these concepts have explicitly unique meanings, the rich text information and high quality links, which can work as a bridge for the cross domain resources.

To exploit the concepts in Wikipedia, we propose a two-step approach in our CDS framework. In the first step, resources from different domains are mapped to the same space, the

[1]http://www.flickr.com/
[2]http://delicious.com/
[3]http://www.youtube.com/

concept space of Wikipedia. This mapping is done by examining the tags assigned to each Web 2.0 resource. Intuitively, if two resources are associated with similar tags, they possibly describe the similar topics, regardless of their modalities. However, as tags are normally input by the humans, which may contain noisy terms or even typos, they cannot be directly used in the search. For instance, the tag can have the homonym (a single tag with different meanings) and synonym (multiple tags for a single concept) problem. Resource tagged as "orange" may refer to a kind of fruit or it just denotes the color of an object. A picture of Apple's operating system may be tagged as "Leopard", which may be confused with the animal. It is also unreasonable to assume that all the tags are relevant to the resource content. People perhaps apply an excessive number of unrelated tags to the resources such that the supposed unrelated resources will be connected by the spam tags. Fortunately, we can apply the Wikipedia concepts to remove the vague and low-quality tags.

Given the Wikipedia concept space $\mathcal{C}$, we wish to present both the documents (DOC) and images (IMG) in the same space so that direct links can be built between them.

$$DOC \Rightarrow \mathcal{C} \Leftarrow IMG$$

We achieve the above objective by representing the resources in the *uniform concept vector*.

Definition 1: Uniform Concept Vector: For each resource, we build a vector $V = \{v_1, v_2, ...v_n\}$ *as the description.* v_i *represents a Wikipedia concept and its value indicates the confidence of the relation between the concept and the resource.*

Given the vectors, we are able to evaluate the similarity between various Web 2.0 resources in the same space. The uniform concept vector of a resource is established according to its tags. However, not all of them are equally useful when building the vector, e.g., the spam tags, therefore, we first select a set of key tags and then generate a set of concept elements based on their relevancy.

The second component of our framework deals with the cross domain resource retrieval scenario where both the query and results could be in various modalities. In particular, we assume there is no tags associated with the query. This assumption could give our users more flexibility when issuing a query. Given a query, as above, we build the uniform concept vector of it as well, derived from their **top-K Homogeneous Resources** which are already stored in the database.

Definition 2: Top-K Homogeneous Resources(THR): Given a query Q, its THR should satisfy the following three requirements: $\forall r \in Q.THR$
1) Both r *and* Q *are in the same modality.*
2) r *is already represented by a uniform concept vector.*
3) r *is one of the top-k similar resources of* Q *according to a similarity function.*

Based on the uniform concept vectors of both the query and resources, we perform a concept-sensitive retrieval method to return the relevant results.

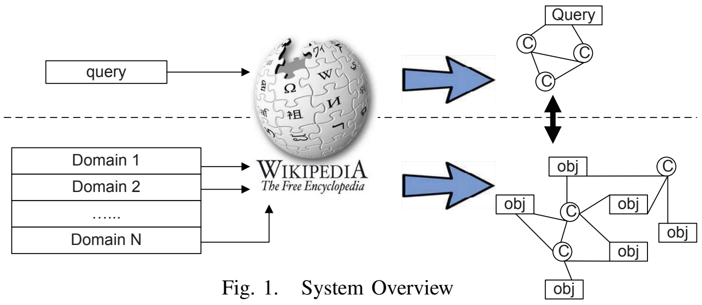


Fig. 1. System Overview

In summary, the main contributions of our paper includes:

- We present a general framework to integrate resources of different Web 2.0 systems and support cross domain search.
- We provide several methodologies which can be applied to many other applications, such as the spam tag detection, topic clustering, query expansion and automatic image annotation.
- We conduct a comprehensive performance study to validate our techniques.

The rest of this paper is organized as follows. In the next section, we state our problem and the proposed system architecture. In Section III, we explain how to select the useful tags and build the uniform concept vector. In Section IV, we present our cross domain resource retrieval method. We report our experimental study in Section V and review related works in Section VI. Finally we conclude the whole work in Section VII.

II. Problem Statement

Various types of Web 2.0 systems are established to serve different groups of users. For example, Flickr is the first choice for the photographers to share their products, while Youtube is the playground for the video lovers. To enhance the users' experience, most of these systems provide an in-site search service, which exploits the state-of-art information retrieval techniques to improve the search results. However, to our knowledge, none of current systems provides an integrated search that allows the user to search all Web 2.0 systems by various modality queries via a single search portal, as it is challenging to provide a universal metric to link and rank different types of resources.

Formally, each Web 2.0 system can be considered as a domain, $\mathcal{D}$. Given two resources r_i and r_j in the same domain, we have a function $f(r_i, r_j)$ to evaluate how similar the two resources are. But if r_i and r_j come from different domains, no such similarity function exists. The idea of this paper is to exploit the semantics of Wikipedia to connect different domains so as to support cross domain search. Wikipedia is the largest online encyclopedia and to this date is still growing with newsworthy events and topics often added within a few days. It contains more than 2 million entries, referred to as

Wikipedia concepts throughout this paper, and most of them are representative name entities and keywords of different domains. The size of Wikipedia guarantees the coverage of various resources in different domains. In addition, the rich semantics and collaboratively defined concepts could help us relieve the disambiguation and noisy problems in the description of Web 2.0 resources.

Figure 1 shows the overview of the system. For each Web 2.0 system, we develop a crawler based on the provided API. All crawled resources are maintained in our data repository. We organize the resources by their domains and a similarity function is defined for each domain. In the offline processing, resources from different domains are mapped to the Wikipedia concepts by their tags and represented by the uniform resource vectors. In this way, resources from different domains are linked via the concepts. Given a query, which may be issued to an arbitrary domain, the query processor translates it into the uniform concept vector via the same approach. Then, query processing is transformed into matching resources of similar concepts, which can search all domains seamlessly.

A. Wikipedia Concept

Before presenting the details of our cross domain search, we give a brief introduction of Wikipedia concepts. In this paper, each Wikipedia article is considered as the description of a concept. The title of the article is used as the name of the concept. For example, Figure 2 shows the article about concept "VLDB". In this way, the Wikipedia dataset can be considered as a collection of concepts, $\mathcal{C}$. We use $D(c)$ to denote the article of concept $c \in \mathcal{C}$. To catch the semantics between concepts, we define three relationships for them.

The first relationship is defined based on the article structures of Wikipedia.

Definition 3: **Link between Tag and Concept**
In a Wikipedia article $D(c_i)$, if tag t is used to refer to another concept c_j, t is linked to c_j and we use $t \rightsquigarrow c_j$ to denote the relationship.

In Figure 2, tag $tuples$ and $filesystem$ are used to refer to the concept "Tuple" and "File System", respectively. Therefore, we have $tuples \rightsquigarrow Tuple$ and $filesystem \rightsquigarrow File\ System$. Generally, if $t \rightsquigarrow c_i$, when clicking t, Wikipedia will jump to the article $D(c_i)$. This behavior is similar to the hyperlinks between webpages. As Wikipedia articles are created by the internet authors, who may use different tags to describe the same concept, multiple tags are probably linked to one concept. Then, we can have $t_x \rightsquigarrow c_i$ and $t_y \rightsquigarrow c_i$. To measure how closely a tag is related to a concept, we define $w(t_x \rightsquigarrow c_i)$ as how many times tag t_x is linked to c_i in Wikipedia.

The second relationship is used to track the correlations of concepts. The intuition is that if two concepts appear in the same article, they may be highly correlated with a high probability.

TABLE I
WIKI-DB INTERFACE

Function Name	Description
$w(t_i \rightsquigarrow c_j)$	Return the link score between tag t_i and concept c_j
$P(c_j \vert c_i)$	Return the correlation of c_j to c_i
$d_c(c_i, c_j)$	Return the semantic distance between c_i and c_j

Definition 4: **Correlation of Concepts**
Concept c_i is said to be correlated with concept c_j, if

- $\exists t \in d(c_i) \rightarrow t \rightsquigarrow c_j$
- $\exists t \in d(c_j) \rightarrow t \rightsquigarrow c_i$
- *there is a document $D(c_0)$, satisfying $\exists t_1 \in D(c_0) \exists t_2 \in D(c_0)(t_1 \rightsquigarrow c_i \wedge t_2 \rightsquigarrow c_j)$*

Based on Figure 2, concept "VLDB", "Tuple", "File System" and "Terabyte" are correlated to each other. In particular, given two concepts, c_0 and c_1, we use

$$P(c_1|c_0) = \frac{\sum_{c_i \in \mathcal{C}} \theta(O(c_i), c_0) f(O(c_i), c_1)}{\sum_{c_i \in \mathcal{C}} \theta(O(c_i), c_0)}$$

to compute the correlation of c_1 to c_0. $\theta(O(c_i), c_j)$ returns 1 if there is a tag in $D(c_i)$ linking to c_j. Otherwise, $\theta(O(c_i), c_j)$ is set to 0.

The last relationship is derived from the hierarchy of Wikipedia. In Wikipedia, articles are organized as a tree[4] since each article belongs to multiple categories and there is hierarchy existing in the categories. We use the tree distance to represent the semantic distance of concepts. To simplify the presentation, we use $L(c_i)$ to denote the level of the concept. Specifically, the level of root category "Articles" is 0. The semantic distance of concept c_i and c_j is defined as:

Definition 5: **Semantic Distance**
Given two concepts c_i and c_j, let c_0 be the lowest common ancestor of c_i and c_j. The semantic distance of c_i and c_j is computed as:

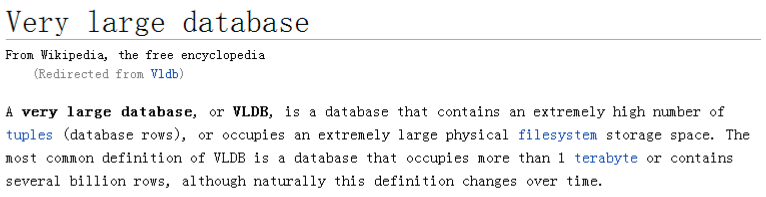
$$d_s(c_i, c_j) = L(c_i) + L(c_j) - 2L(c_0)$$

In this paper, the above relationships are combined and used to rank the similarities between concepts and tags. However, it is costly to evaluate the similarities of concepts on the fly. Therefore, in the preprocessing, we scan and compute all the tag links, concept correlations and semantic distances. The preprocessing results are maintained in our MySQL database. We wrap the database searches to a set of simple interfaces as shown in Table I.

III. CROSS-DOMAIN CONCEPT LINKS

We develop customized crawlers for each Web 2.0 website. The crawled resource (including images, videos and web pages) is abstracted as $(\mathcal{T}, \mathcal{V})$, where $\mathcal{T}$ denotes the tags assigned to the resource and $\mathcal{V}$ is the binary values of the

[4]Strictly speaking, as some articles are classified into multiple categories, the concept graph is not a tree. But in most cases, it can be processed as a tree.


Very large database

From Wikipedia, the free encyclopedia
(Redirected from Vldb)

A **very large database**, or **VLDB**, is a database that contains an extremely high number of tuples (database rows), or occupies an extremely large physical filesystem storage space. The most common definition of VLDB is a database that occupies more than 1 terabyte or contains several billion rows, although naturally this definition changes over time.

Fig. 2. Wikipedia Snippet

resource. Given a concept space $\mathcal{C}$ in Wikipedia, our idea is to create a mapping function between the resources and concepts. The result is the uniform concept vector $v_i = (w_1, w_2, ..., w_n)$ for each resource r_i. w_j denotes the similarity between r_i and concept $c_j \in \mathcal{C}$ (namely, the weight of c_j). The mapping is constructed via the tag sets, which can be formally described as:

Definition 6: **Mapping Function**
The mapping function f is defined as $f = \mathcal{T} \times \mathcal{C} :\Rightarrow w_1 \times w_2 \times ... \times w_n$, where $n = |\mathcal{C}|$ and $0 \leq w_i \leq 1$.

In this section, we first discuss how we process the tag set $\mathcal{T}$ for each resource and then we introduce our tag-based mapping function.

A. Tag Selection

Several studies have been done on mapping the tags to Wikipedia concepts. The first category is quite a straightforward method. It leverages the power of existing search engines, e.g., Google[5] to identify related concepts[6]. The second category utilizes Wikipedia to generate features of text fragments, e.g., explicit semantic analysis (ESA)[7][8] and then look for the corresponding concepts. However, we argue that as people can tag the resources with arbitrary phrases, it is necessary to filter and remove the spam tags. As an example, in our collected Flickr dataset, each image is tagged by about 10 unique tags. Many of them are ambiguous or unrelated to the central topic of the images. If all those tags are applied to search Wikipedia concepts, we will end up with too many unrelated concepts.

The tag selection affects the efficiency and accuracy of the mapping process. Based on the observation that correlated tags are normally used for one resource, we propose a cluster-based approach. Given a tag set $\mathcal{T}$, our idea is to group the tags into a few subsets: S_1, S_2,...,S_k. The subsets satisfy

$$\bigcup S_i = \mathcal{T} \& S_i \cap S_j = \emptyset$$

KL-divergence is used to measure the quality of clustering. In information retrieval, they have a theory that if the keywords are topic similar, the query comprised of those keywords will get a better result. In the statistical point view, the returned pages will represent a different keyword distribution with the whole collection[9]. We adopt the similar idea here. The intuition is that a good algorithm should result in a set of clusters whose related Wikipedia articles have a significantly different tag distribution with the whole article set. Let $\mathcal{W}$ and $\mathcal{C}$ denote the whole tag set and concept set of Wikipedia respectively. For two subsets, S_i and S_j, we have

$$KL(S_i, S_j) = \sum_{t_x \in \mathcal{W}} P(t_x|S_i \cup S_j) \log \frac{P(t_x|S_i \cup S_j)}{P(t_x|\mathcal{W})} \quad (1)$$

$P(t_x|\mathcal{W})$ is computed as the probability that tag t_x appears in all articles of Wikipedia

$$P(t_x|\mathcal{W}) = \frac{tf(t_x)}{\sum_{t_y \in \mathcal{W}} tf(t_y)}$$

while $P(t_x|S_i \cup S_j)$ is estimated using the following way.

We replace $S_i \cup S_j$ with their involved articles in the Wikipedia. In particular, for a tag $t_x \in S_i \cup S_j$, let $D(t_x)$ be the articles[6] that related to t_x, that is
$D(t_x) = \bigcup D(c_i)$ *where* $w(t_x \rightsquigarrow c_i) > 0$.
The whole involved articles are

$$D(S_i \cup S_j) = \bigcup_{t_x \in S_i \cup S_j} D(t_x)$$

Then, $P(t_x|S_i \cup S_j)$ is computed as:

$$P(t_x|S_i \cup S_j) = \sum_{t_y \in S_i \cup S_j} P(t_x|t_y) P(t_y|S_i \cup S_j)$$

$$= \sum_{t_y \in S_i \cup S_j} P(t_y|S_i \cup S_j) \sum_{c_z \in D(S_i \cup S_j)} P(t_x|c_z) P(c_z|t_y)$$

where c_z is a concept in the article set $D(S_i \cup S_j)$. To compute $P(t_x|c_z)$, we define $tf(t_x, c_z)$ as the term frequency of t_x with regarding to the article of c_z.

$$P(t_x|c_z) = \frac{tf(t_x, c_z)}{\sum_{t' \in \mathcal{W}} tf(t', c_z)} \quad (2)$$

Recall that in last section, we compute the link score between a tag and a concept (See Table I). It can be applied to estimate $P(c_z|t_y)$ as well. $P(t_y|S_i \cup S_j)$ can be estimated as the relative frequency of t_y in $S_i \bigcup S_j$.

$$P(c_z|t_y) = \frac{w(t_i, c_j)}{\sum_{c' \in \mathcal{C}} w(t_i, c')}$$

Iterating all tags in $\mathcal{W}$ is costly. Instead, in the computation, we only use tags in $\mathcal{T}$ to estimate Equation 1. Based on the analysis and experiments in [10], such simplification does

[5]http://www.google.com

[6]$D(t_x)$ can be also considered as a set of concepts for the corresponding articles.

Algorithm 1: Wikipedia based Tag Clustering

```
Input: Tag set $\mathcal{T}$ of the input resource
Output: Subsets of $\mathcal{T}$
1  AllSubset $result = \emptyset$ ;
2  foreach $t_i \in T$ do
3      Subset $S_i$ = new Subset($t_i$) ;
4      $result$.add($S_i$);
5  foreach $i$= 1 to $result.size$ do
6      foreach $j$= 1 to $result.size$ do
7          if $i \neq j$ then
8              $M_{i,j} = Gain(S_i, S_j)$ //KL matrix;
9  while true do
10     Double $max$=0, int $a$=0, int $b$=0 ;
11     foreach $i$= 1 to $result.size$ do
12         foreach $j$= 1 to $result.size$ do
13             if $M_{i,j} > max$ then
14                 $max = M_{i,j}$;
15                 $a = i, b = j$ ;
16     if $M_{i,j} > th$ then
17         Subset $S_{new} = S_a \cup S_b$;
18         $result$.remove($S_a$), $result$.remove($S_b$) ;
19         $result$.add($S_{new}$);
20         update $M$ by removing the rows and columns
           involving $S_a$ and $S_b$;
21         add in a new row and column in $M$ for $S_{new}$ to
           record the gain of $S_{new}$ to all other subsets ;
22     else
23         break;
24 return $result$ ;
```

not degrade the accuracy significantly while save a lot of computation. In addition, we normalize the KL value by the total tag number in the subset.

Given a crawled resource with a tag set $\mathcal{T}$, Algorithm 1 illustrates how we group the tags into disjoint subsets. Initially, $|\mathcal{T}|$ subsets are generated with each subset only containing one tag (line 2-4) and a KL-Matrix is computed to record the gain value (introduced later) between any two subsets (line 5-8). Then, we iteratively combine the subsets with current maximal gain until reaching the threshold (line 9-23). When a new subset is created, we replace the old entries in the matrix with new ones (line 16-21). In this way, the gain is incrementally updated to support the next iteration of clustering.

In the experiment, we discover that if one of the child subsets gets a high KL score, then the score of the new merged subset tends to be high as well. In order to capture the benefit of forming a new subset more accurately, we use the gain of KL score as the merging criteria.

$$\begin{aligned} Gain(S_i, S_j) &= KL(S_i, S_j) - KL(S_{ii}, S_{ij}) \\ &+ KL(S_i, S_j) - KL(S_{ji}, S_{jj}) \end{aligned}$$

$S_{ii}, S_{ij}, S_{ji}, S_{jj}$ are the respectively child subsets of S_i and S_j. The value of the threshold th should be set adaptively based on the tag distribution. Our experience shows that using a small number of samples is enough for estimating a good threshold.

The generated subsets are then sorted by their weights, which are computed as follows:

$$W(S_i) = |S_i| * \sum_{t_x \in \mathcal{T}} P(t_x|S_i) \log \frac{P(t_x|S_i)}{P(t_x|\mathcal{T})} \quad (3)$$

The weight definition combines the size of the subset and the tag distribution. It can be evaluated in the same way as Equation 1. In our current implementation, we assume that each resource only has one core topic and therefore, we just keep the subset with the highest weight. In other words, we prune the unimportant tags from the tag set $\mathcal{T}$.

B. Concept Mapping

Our selected tags are used to discover correlated concepts of Wikipedia and establish the mappings between the resource and concepts. We observe that most tags are linked to more than one concepts, which can be classified into different categories. For instance, the tag *Marina Bay* is linked to a few concepts such as "Marina Bay MRT", "Marina Bay Sands" (the casino), "Marina Bay Financial Centre" and "Marina Bay Singapore". Supposing the image is taken for the Merlion at Marina Bay, only the last concept is the correct match, while the other concepts, if applied in search, will definitely cause ambiguities.

To address this problem, we exploit the context of tags. For a resource, all its tags in $\mathcal{T}$ after pruning are considered as context tags to each other. If an image is tagged with *Merlion, Singapore, Nikon* and *Marina Bay*, we can infer that *Marina Bay* refers to the concept "Marina Bay Singapore". For a tag $t \in \mathcal{T}$, its context tag set is $\mathcal{T} - \{t\}$. Let $C(t)$ and $\mathcal{C}$ denote all concepts linked by t and the whole concept space, respectively. We have $C(t) = \{c | w(t \rightsquigarrow c) > 0 \wedge c \in \mathcal{C}\}$. The similarity between concepts in $C(t)$ and t can be estimated as:

$$s(c, t) = P(t|c)P(\mathcal{T} - \{t\}|c) = P(t|c)\Pi_{t_i \in \mathcal{T} - \{t\}} P(t_i|c) \quad (4)$$

$P(t|c)$ can be computed as $\frac{w(t \rightsquigarrow c)}{\sum_{t_x \in \mathcal{W}} w(t_x \rightsquigarrow c)}$. Otherwise, t_i does not have explicit connection with c ($w(t_i \rightsquigarrow c)$). In this case, we use the concept correlations to discover the hidden semantic links between tags and concepts. Formally, we expand $P(t_i|c)$ as

$$P(t_i|c) = \sum_{c_j \in C(\mathcal{T})} \frac{P(t_i|c_j)P(c_j|c)}{d_c(c_j, c)} \quad (5)$$

where $P(t_i|c_j)$ is computed as Equation 2, $P(c_j|c)$ is the correlation between c_j and c (see Table I) and $d_c(c_j, c)$ is the semantic distance between c_j and c. Combining Equation 4 and 5, we can compute the weights of concepts for each tag. Finally, the candidate concept set of a resource consists of concepts with the highest score of each tag $C(\mathcal{T}) = \bigcup_{t \in \mathcal{T}} \{c | \max s(c, t) \wedge c \in C(t)\}$

Finally, the uniform concept vector of the resource can be generated by iterating all concepts in $C(\mathcal{T})$.

1) If c_j is not in $C(\mathcal{T})$, w_j is set to 0.
2) Otherwise, $w_j = \sum_{t_i \in \mathcal{T}} s(c_j, t_i) * W(S)$. $W(S)$ is the weight of the generated tag subset of $\mathcal{T}$ which can be computed as in Equation 3.

In the offline processing, we compute the concept vectors for all crawled resources and store them in our database. Thus, the various modal resources are transformed to the uniform representations which are facility?? for future operations.

IV. Cross Domain Search

The second component of our system is the resource retrieval part which performs cross domain search. Our cross domain search accepts various types of queries. The user can submit typical keyword queries; or he can upload an image or document as the query. The difference between the cross domain search and conventional search service is that the user can issue a query to any specific domain and the cross domain search can return results from different domains. For example, the mobile user can take a photo of a building and submit the image as a query to find the documents, videos and other images associated with the building.

In our system, the query is transformed into the uniform concept vector and we retrieve the crawled resources with similar concepts as the results. The key challenge here is how to map queries into concepts. Different from crawled resources which are tagged by users, the query normally does not come along with any tag. We cannot exploit the tag-concept links as in the last section. Therefore, we plan to build the concept vector of a query by leveraging its THR. Next, we first introduce the way of extracting THR.

A. Intra-Domain Search

For query q and resource r_i in the same domain $Domain_j$, the query processor computes their similarities via the similarity function $Sim(q, r_i)$. Only the resources with high similarities are returned as the results. In this section, we use text documents and images as our examples to show the idea of intra-domain search. As a matter of fact, other domains, such as videos, can be easily integrated, if the similarity function has been defined.

1) Document: In the domain of text documents, each document is represented as a word vector, $v(r_i) = (t_1, t_2, ..., t_n)$. Each dimension of the vector refers to a specific term and its value denotes how important the term is. In this paper, $tf-idf$ is used as our metric to measure the term weights. Therefore, $t_j = tf_{ij} \times idf_j$. The query in this domain is also transformed into a word vector $v(q)$ and similarity is computed as the cosine distance.

$$Sim(q, d_i) = \frac{v(d_i) \times v(q)}{|v(d_i)||v(q)|}$$

2) Image: In the domain of image, each image is represented as a feature vector, $v(r_i) = (f_1, f_2, ..., f_n)$. Each dimension of the vector represents an image feature, such as color, texture and shape. These features are proven to be effective in many applications[11]. In our system, the visual features of the image are viewed as a distribution in the visual space. Thus, the similarity between images and query images is captured by the KL divergence.

$$Sim(q, d_i) = \sum_{j=1}^{n} v(q).f_j \times \log \frac{v(q).f_j}{v(d_i).f_j}$$

Parameter K of THR may affect the recall and precision of the results and this will be studied in our experiments.

B. Uniform Concept Vector Building of Query

One straight-forward way of building the vector is to combine the concept vectors of its THR. Let THR be the candidate resources of query q. We generate the concept vector for q by aggregating the vectors of the candidate resources:

$$v_q = \sum_{r_i \in THR} v_i$$

For a specific concept c_j, its weight for q is

$$w(q)_j = \sum_{d_i \in THR} v_i[w_j]$$

where $v_i[w_j]$ is the jth weight of d_i's concept vector.

However, we should note that not all the concepts existing in the concept vectors of its THR are the proper description of the query. For example, Figure 3 shows a query image and its top-4 homogeneous resources. All four candidates are very similar to the query image based on their image features. However, in fact, two of them (a and c) are not related to the query image and will bring misleading concepts. Using them as candidate resources will definitely generate noisy concepts. In addition, the above naive ranking approach fails to consider the correlations between concepts and assumes that all candidate resources are equally important. This may degrade the precision of results, because the false positives (such as a and c in Figure 3) are introduced into the search process. To improve the quality of the vector, we adopt a more sophisticated ranking approach. The intuition is that different from the noisy candidate resources, the good results always have correlated concepts (the concepts of the query).

Our ranking approach considers three factors:

1) *Resource Importance* : Concepts from different candidate resources should not be considered equally. For a concept c_j from resource r_i, we normalize c_j's weight by $Sim(q, r_i)$, the similarity between the query and the resource.
2) *Concept Importance* : Each concept in Wikipedia links to some other concepts. If concept c_i exists in the article of concept c_j, we say c_j is linked with c_i. Usually the general concepts will connect to more concepts than the

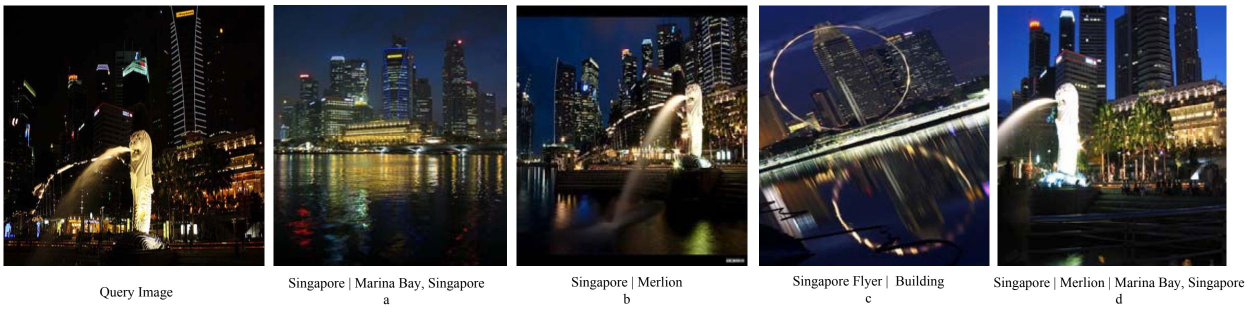

Query Image
Singapore | Marina Bay, Singapore
a
Singapore | Merlion
b
Singapore Flyer | Building
c
Singapore | Merlion | Marina Bay, Singapore
d

Fig. 3. A Query Image and Its Top-4 Intra-domain Similar Images

specific concepts. We propose to compute a concept's generality as following:

$$g(c) = \frac{1}{log\frac{|\mathcal{C}|}{|links(c)|} + \lambda}.$$

where $|\mathcal{C}|$ denotes the total number of concepts in Wikipedia, $|links(c)|$ indicates the number of concepts linking with c and λ is a parameter to smooth the function. A larger $g(c)$ implies a more general concept, which is less distinguished in search.

3) *Concept Correlation* : We exploit the correlations between concepts to find the hidden semantics. Given a concept c which not appear in resource $r_i$from THR , we can measure its weight via its correlations with other concepts in r_i. Specifically, for resource $r_i \in THR$,

$$s(c, r_i) = \sum_{c_j \in r_i.concept} P(c|c_j)v_i[w_j]$$

where $P(c|c_i)$ is the correlation in Table I and $v_i[w_j]$ is the jth weight of resource r_i's concept vector which corresponding to c_j.

Combining the above factors, we adjust the scores of concepts in the candidate resources. For query q, the new score of c is computed as:

$$w(c, q) = \frac{\sum_{d_i \in THR} s(c, d_i)Sim(q, d_i)}{g(c)} \qquad (6)$$

In this way, we can generate the uniform concept vector for q and use it to rank the resources of different domains.

C. *Resource Search*

To facilitate the search, all concept weights are normalized into the range of $[0, 1]$. The uniform concept vector can be considered as an n-dimensional point, where n is the number of concepts. Although there may be millions of concepts, for each resource, only a small portion of concepts are involved. Given a query q and its concept vector $v_q = (w_1, w_2, ..., w_n)$, we only need to retrieve the resources, which share the same concepts with q. For this purpose, an inverted index $(cid, DList)$ is built. cid is the concept ID and the $DList$ is a set of resources, satisfying that

$$\forall r_i \in DList \rightarrow v_i[cid] > 0$$

Namely, the concept vectors of resources in $DList$ have a non-zero weight for concept cid. Let v_q and v_i be the concept vector of query q and resource r_i, respectively. Their similarity is estimated by combining the cosine distance of the concept vectors and the intra-domain similarity:

$$Score(q, r_i) = \frac{v_q \times v_i}{|v_q||v_i|} + Sim(q, r_i)$$

If q and r_i are in different domains, $Sim(q, r_i)$ is set to 0. Otherwise, it can be computed by the intra-domain similarity function.

V. Experimental Study

We have implemented our cross domain search system[12] based on the proposed framework. In this section, we evaluate its performance with real dataset from Web 2.0 sites. In particular, we collect images from Flickr and documents from Delicious as the experimental data. After removing the duplicated contents, the repository includes $136k$ images and $114k$ documents. Those images/documents are considered as the cross domain resources in our system, which are classified into different categories, such as food, landmarks, sports, airplanes and etc. The English version of Wikipedia, which is released on 2008.07.24, is employed as our knowledge base. It contains around $2.7M$ concepts, covering almost every domain. We summarize the dataset statistics in Table II.

In the following subsections, we perform a comprehensive study on our adopted strategies. All of our experiments are conducted on a linux server with Quad-Core AMD Opteron(tm) Processor 8356, 128GB memory and running RHEL 4.7AS.

A. *Uniform Concept Vector Building*

In our system, each resource is linked to a set of Wikipedia concepts using a two-step process. In the first step, we cluster the tags and select the most important tags to represent the resource. In the second step, based on the concept correlations, we generate a concept with the highest weight for each left tags. In this experiment, a test resource set *TR* is generated by randomly choosing 25 images (referred as *IMG*) and 25 documents (referred as *DOC*) from the underlying data repository. The average number of tags for resources in *IMG* and *DOC* are 7.7 and 9.4 respectively.

TABLE II
CROSS DOMAIN RESOURCE DATA STATISTICS

	Resource Number	Unique Tag Number	Total Tag Number
Flickr	135,962	120,097	1,290,511
Delicious	114,085	133,943	4,006,636

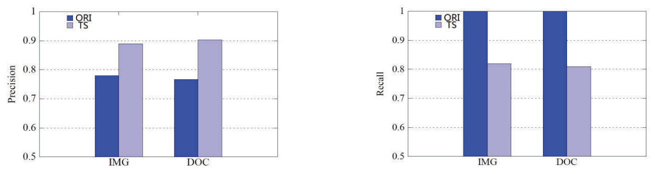


Fig. 4. Tag Selection Performance on TR

1) Tag Selection: To measure the quality of tags selected by our clustering algorithm, we define two evaluation metrics: *precision* and *recall*. The $precision$ is computed as $\frac{r}{l}$, where l is the total number of tags associated with the resource and r is the number of relevant tags. $precesion$ is used to measure the effectiveness of our algorithm in terms of removing unimportant tags. On the other hand, the $recall$ is computed as $\frac{h}{r}$, where r is defined as above and h is the number of relevant tags selected by our algorithm. A high $recall$ indicates that our algorithm can retain most relevant tags.

We illustrate how these two scores are computed using the examples in Table III. The first row is about an image, which describes a F1 racing car in Singapore Grand Prix. The left column shows the original tags annotated by the user, while the right column is the tags returned by our algorithm. As "Singapore, Grand Prix, F1, FIA, Racing, Motorsport" are the relevant tags and "1" is not a candidate tag, the precision of original tagging (ORI) is $\frac{6}{7}$. Correspondingly, the precision of the tags after our tag selection algorithm (TS) becomes $\frac{4}{4}$ since all 4 selected tags are relevant. We assume ORI contains all the relevant tags. Thus the recall of the ORI is always 1. On the contrary, we can compute the recall of TS as $\frac{4}{6}$.

The comparison result of ORI and TS is shown in Figure 4. The high precision and recall scores indicate that TS is capable of removing most irrelevant tags and keep the relevant tags as many as possible. The reason that TS achieves such a high precision is due to the usage of clustering. If we are able to identify the correct cluster, as all tags in the cluster are semantically closed to each other, with a high probability, the tags in the cluster are all high-quality tags for the resource.

For the recall perspective, although TS falsely removes some relevant tags, the average recall, 0.8, is still acceptable. In most cases, the false negatives are caused by the tags describing the context of the resource, e.g., "Singapore" in the above example. It is too general to be categorized into a cluster with other tags. Another reason is that we only keep the tags in the highest weighted cluster, while some resources are annotated with tags that come from different semantic categories. In the future work, we will study how to set the cluster number adaptively.

TABLE III
EXAMPLES OF TAG SELECTION

Original Tags	Processed by TS
Singapore, Grand Prix, 1, F1, FIA Motorsport, Racing	F1, Racing, Motorsport, FIA
Singapore, Merlion, Marina Nikon, Photograph	Singapore, Merlion Marina
3d, artist, colour, colourwallah exhibition, gallery, london, sculpture	sculpture, exhibition artist

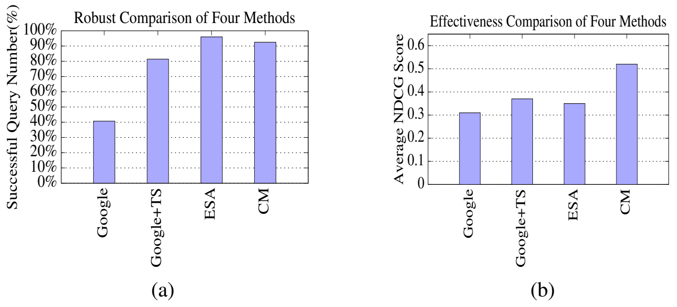


Fig. 5. Comparisons of Four Methods

2) Concept Mapping: In this experiment, we show the effectiveness of using uniform concept vectors to represent the resources. Specifically, we compare our concept mapping method (*CM*) with three other concept generation methods, including *Google*, *Google+TS* and *ESA*. In *Google* method, we query Google by the tags of a resource in *TR* and restrict the search scope in Wikipedia.org. The returned article titles are used as the resource's Wikipedia concepts. This method is used to verify whether the search engine method still works in our cross domain scenario. In *Google+TS* method, we first process the queries by our *TS* algorithm and then search in *Google*. By doing this, we examine whether the performance will be improved if we apply our tag selection on the tags. *ESA* is the state-of-art technique introduced before.

Robustness: Figure 5(a) shows the percentage of queries that have valid results in each method. We find that *Google* only returns valid results for about 40.7% queries, while the rest of queries get no result at all. The reason is that Google cannot effectively handle the queries which contain ambiguous keywords or too long. However, after enhancing with our tag selection techniques, the *Google+TS* scheme becomes more robust since the spam tags are pruned. The best results are observed from the TS and ESA approach, which can process more than 90% queries.

Effectiveness: To measure the effectiveness of different schemes, we adopt the Normalized Discounted Cumulative Gain(NDCG) metric [13]. In this metric, the relevance of returned results are labeled as one of the five levels (1-5). The average NDCG score for the query set is computed as follows:

$$NDCG(Q,k) = \frac{\sum_{j=1}^{Q} Z_{kj} \sum_{i=1}^{k} \frac{(2^{r(j,i)}-1)}{log(1+i)}}{|Q|}$$

where Z_{kj} is a normalization constant so that the optimal NDCG is 1 and $r(j,i)$ is the relevance level of the ith concept (tag) in query j. This metric takes both relevancy and ranking order into consideration and is widely used to measure the result quality. To compute the NDCG score, ten students from the CS department are volunteered to give scores for each query.

Apparently, TS approach outperforms all the other methods as shown in Figure 5(b). This is because the concepts in TS are generated for each individual tag. Therefore, fewer noisy concepts will be introduced. However, one limitation of TS is that when a phrase is falsely segmented into separated parts, e.g., from “Paris Hilton” to “Paris” and “Hilton”, we are unable to find the correct concept (“Paris Hilton”, the celebrity) which can represent them as a whole. This also explains why our method performs worse in *DOC* than in *IMG*. As in Delicious, the false segmentation happens more frequently. In addition, the comparison of $Google$ and $Google+TS$ in both robustness and effectiveness experiments reveals the usefulness of our *TS* method in another perspective. It demonstrates that we can indeed remove the noisy unnecessary tags in the resources.

B. Cross Domain Search Evaluation

In this section, we compare our concept-based retrieval (CBS) with the tag-based retrieval (TBS) method. TBS adopts the similar idea as CBS but based on the tags. The intuition is to examine how much the results can be improved by representing resources with Wikipedia concepts instead of the manually added tags. The resources in TR are used as queries. In the test, we discard the tags for the query resource, and hence the query is just the image/document itself. As mentioned in Section IV, we perform the cross domain search in two steps: build the concept vector for the query and then perform the resource retrieval in the repository.Before evaluating the two steps, we first investigate the effect of K in THR.

1)THR Selection: To generate the uniform concept vector of a query, we exploit its Top-K homogeneous resources. The effect of different K values is shown in Figure 6. The y axis represents the NDCG score of the generated concepts for different K. In the diagrams, both of the two datasets achieve their highest score when $K = 50$. The reason is that if K is set to a small value (e.g., 20), many related concepts cannot be discovered due to the small number of intra-domain resources. On the other hand, when K is set to a large value (e.g.,100), noisy concepts are introduced with a high probability and they may increase each others’ weights via the correlations, which is even worse when processing the query. In summary, a proper number of intra-domain similar resources are preferred. Another interesting observation is that the scores of DOC are higher than those of IMG. It is because the intra-similarity function between documents is much more accurate than that of the images. Therefore, more relevant concepts can be retrieved in the document setting. The third finding is that the gap between scores of IMG is smaller than that of DOC. After checking the data, we find that the gap of intra-similar score between the image query and its candidates are much greater than the scores between text query and similar documents. For example, on average, the top ranked candidate image has 8 times higher score than that the image ranked 50th, and 150 times higher than the 100th one. However, the corresponding data for DOC is only 2.2 and 2.9. This comparison illustrates that the lower ranked image candidates do not have as much influence on the final results as that of DOC so that including more candidates will not hurt the effective performance too much but make the whole process become slower.

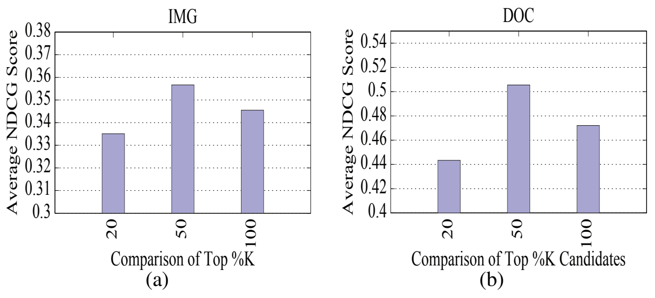


Fig. 6. Top K Comparison

2) Uniform Concept Vector Building of Queries: We compare the relevancy of concepts (tags) generated by *CBS*, *TBS* and the fixed annotation-based cross-media relevance model(FACMRM)[14] which is an automatic image annotation method. By using a training set of annotated images, FACMRM learns the joint distribution of images and words. Then fixed length annotations can be generated by using the top k words to annotate the images. Here, we adapt the FACMRM idea as a method of generating concepts for a query. In the evaluation, as before, we choose TR as the query set and NDCG as the measurement metric as before.

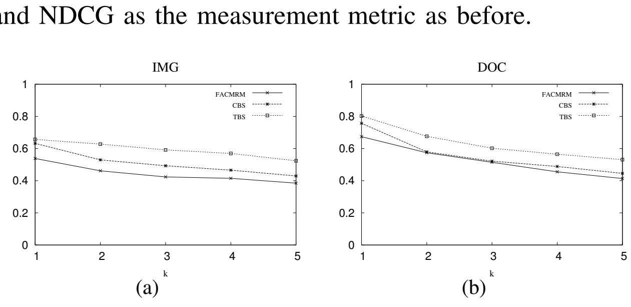


Fig. 7. Average NDCG Comparison of Concept (Tag) Retrieval

Figure 7 illustrates the NDCG score comparison between the three methods. Among the three methods, our CBS perform the best. In the three methods, the highest score is observed when only the first tag is used, which indicates that the most relevant concept (tag) is always ranked the highest. Apparently, CBS and TBS performs better for various values of k. The reason is analyzed as in CBS and TBS, we add the correlation factor into the concept(tag) ranking and this leads to better results. We manually select the ranked concepts(tags) and find that our concept based method tends to reduce the bias which may exist in the underlying repository. For example, "Merlion" is tagged much less than "Singapore" in the repository. Thus "Singapore" has a larger possibility to appear in the *THR* of a query. However, ranking "Singapore" higher does not help the search much as it is so general. Our method can overcome this problem and rank "Merlion" higher.

Generally, the average scores for images are lower than that of documents. The score gap is caused by the semantic gap between low level features and high level semantics of images. The visually similar images are perhaps not semantically similar and may introduce noisy concepts. Compared to images, it is more credible to capture the semantic similarities between documents, leading to the higher score of documents.

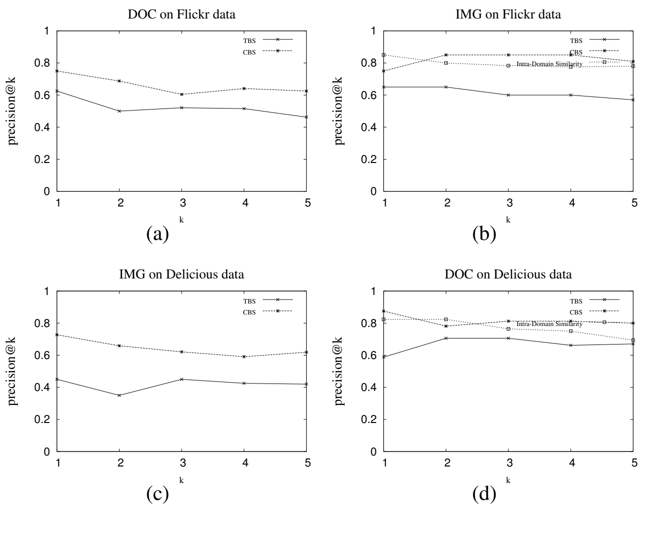


Fig. 8. Precision@k Comparison

3) Cross Domain Search: We do a set of experiments to measure the result quality of cross domain search. Generally the results are evaluated in four categories: searching images by image, searching images by document, searching documents by document and searching documents by image. For each query, we select the top-5 concepts (tags) to build the concept vectors and then retrieve the relevant images and documents respectively.

Figure 8 presents the comparison results of the $precision@k$ (top k returned results). We also present some examples in Figure 9 to better illustrate our results. From the figures and examples, it is clear to show that *CBS* does work and performs better than *TBS*. The reason is that by consistently representing the resources and queries in uniform concept vectors, we can capture the semantics between resources more precisely. Another reason is that concepts can receive a more reasonable weight than the tags. The weights for the concepts of queries play important roles in retrieving the resources. We find that even for the same concept set, when the weights vary a little, the final results will change a lot.

We also compare our methods with the intra-domain similarity function when the query and results are in the same domain. From Figure 8(b) and 8(d), we observe that adding one more step is a double-edged sword. We analyze the reason as follows. If the tags are not assigned correctly, errors will be introduced and propagated in each step. Finally the results will be degraded, such as TBS. However, if we can assign and weight the intermediate concept properly, it does help improve the performance.

Another trend observed from the above figures is that the precision of cross domain search (Figure 8(a), 8(c)) is a little bit lower than that of intra-domain search(8(b), 8(d)). Except the inherent difficulty in cross domain search, it is partially caused by the incompleteness of the underlying resource repository. For example, for some image queries, we have a few or none relevant document resources in our repository.

VI. Related Work

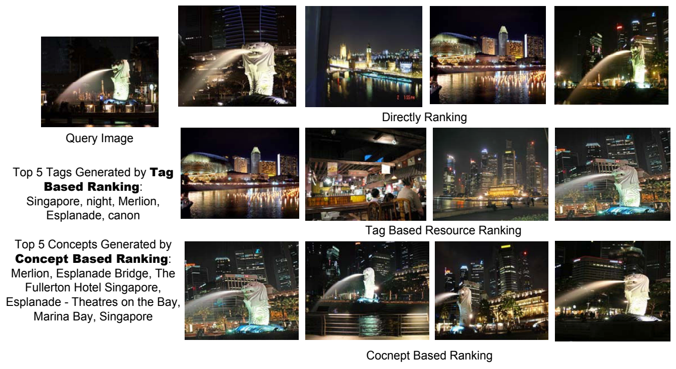
The objective of our system is to automatically map various domain Web 2.0 resources together and support cross domain search with a principled uniform resource model framework. In this section, we will review research and systems on linked data, especially linking data to Wikipedia and previous research effort on cross domain search.

A. Linked Data

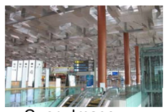
Linked Data is about using the Web to connect related data that wasn't previously linked, or using the Web to lower the barriers to linking data currently linked using other methods[15]. Most investigations conducted in this area pay more attentions on structured data linking[16][17].

Several applications[18][19][20]have recently become publicly available that allow connecting ordinary Web data to knowledge repository, e.g., DBpedia[7], in an automated or semi-automated fashion. In [20], a project prompted by BBC tries to provide background information on identified main actors in the BBC news by means of DBpedia. In [19], a semi-automatic method is proposed to translated tags to Wikipedia concepts. In [18], a service to link Flickr images to Wikipedia is described. However, it mainly focuses on geographic concepts. Similar to these work, we try to link Web data to the formal and accurate knowledge center, Wikipedia. Compared to these work, we pay more attentions on Web 2.0 resources and propose an automatic and general resource linking method. In addition, we provide a unified search function for the linked resources.

[7]http://www.dbpedia.org


Query Image
Top 5 Tags Generated by **Tag Based Ranking**: Singapore, night, Merlion, Esplanade, canon
Top 5 Concepts Generated by **Concept Based Ranking**: Merlion, Esplanade Bridge, The Fullerton Hotel Singapore, Esplanade - Theatres on the Bay, Marina Bay, Singapore
Directly Ranking
Tag Based Resource Ranking
Cocnept Based Ranking

(a)



Top 5 Tags Generated by Tag Based Ranking:
Singapore , Changi , Airport ,新加坡樟宜国际机场第3大厦 ,Terminal

Top 5 Concepts Generated by Concept Based
Indira Gandhi International Airport, Singapore Changi Airport, Changi, London Heathrow Airport , Airport (MBTA station)

Tag Based Ranking	Concept Based Ranking
http://www.zoo.com.sg/index.htm	http://www.airport-hotel.com.sg/
http://www.visitsingapore.com/publish/stbportal/zh/home.html	http://delhi.big-cities.org/airport.html
http://www.streetdirectory.com/	http://www.heathrowairport.com/
http://www.thingstodo-singapore.com/	http://wo61.healthyounger.com/singaporechangiairport.html
http://www.changiairport.com.sg/changi/en	http://www.heathrowairport.com/portal/site/heathrow/

(b)

Query URL: http://www.mountfaber.com.sg/

Top 5 Tags Generated by Tag Based Ranking: Restaurant , Singapore , Tourism, Travel Thailand

Top 5 Concepts Generated by Concept BasedRanking: Restaurant, Cafe, Tourism, Food, Eating

(c)

Query URL: http://www.gp2006.com/home
Top-5 Tags Generated by Tag Based Ranking:
Automobile , Auto racing , Video game, Formula One ,Sport

Top-5 Tags Generated by Tag Based Ranking:
Automobile, Auto racing, Porsche, Formula One, Motorcycle

Directly Ranking	Tag Based Ranking	Concept Based Ranking
http://clubf1.net/new/index.php	http://www.tmsf.jp/	http://www.f1.com/
http://www.tmsf.jp/	http://www.americanlemans.com/news/Article.aspx?ID=2691	http://blogs.iht.com/tribtalk/sports/f1/index.php
http://video.google.com/videoplay?docid=6756084931270105691	http://www.cb-racing.com/	http://www.autosport.com/gallery/
http://motorsport-legend.com/	http://au34.motorists-mico.com/automobileracing.html	http://www.formula1.com/news/
http://www.ferrariworld.com/	http://blogs.iht.com/tribtalk/sports/f1/index.php	http://www.formula1.com/gallery/

(d)

Fig. 9. Four examples of Cross Domain Search

B. *Cross Domain Search*

Generally, many work is related or partially related to cross domain search. As we mainly focus on cross domain search by text or images, we coarsely classify existing work into the following three categories according to the types of input query: searching by text, searching by images or searching by both.

Text Search: This kind of searching includes searching by keywords and documents. The well-known PageRank[21] achieved great success in keyword search. Except keyword search of documents, they further deploy similar philosophy in searching images by keywords [22]. There are also some attempts in searching images by a paragraph or a document. In [23], an approach of searching images using a paragraph is described. They first extract semantic keywords from the paragraph and then do the search in an annotated image database. In addition, querying by documents is studied in [24].

Image Search: Most researches on searching by images are essentially dependent on content based image retrieval techniques [25] (CBIR), leveraging image visual similarity computation. The visual similarity is computed from the low level features of images, such as color, shape and texture [26] [27]. CBIR techniques are the foundation of image retrieval stuff. In a project developed in [28], users can search for aviation photos by submitting a photo.

The wide spread use of mobile devices has prompted the demand of searching text by images. For searching keywords by images, commonly it is referred as automatic image annotation, which tries to generate a set of related words of an image[14][29]. Furthermore, in [30] and [31], they propose techniques of recognizing locations and searching the Web. Furthermore, users can optionally provide text as complementary part in searching by images. In [32], they develop a photo based question answering system to help people find useful information according to a query image.

Cross Domain Search: There are also some works with quite similar goals to us[2][14]. In [2], they study the problem of how to jointly model the text and image co-occurring in the same documents. Then they learn the correlation between the two media with canonical correlation analysis. On the other hand, [14] learns the correlation between keywords and image blobs by a relevance model. We should note that both of these two methods work on a small dataset and require a training dataset which provides precise matchings between text and images. Comparably, our work tries to address a more intractable problem and targets at the larger volume Web 2.0 data with various data strutures. Therefore, in this case, we introduce Wikipedia as a semantic description of the Web 2.0 resources, aiming to avoid the ambiguous connections.

VII. Conclusions

In this paper, we propose a brand new cross domain framework for web-based applications, extending the existing annotation method by linking the resources to concepts in Wikipedia. Our framework fully utilizes the Wikipedia concepts with their well-organized contents and high-quality links. Our framework exhibits high extensibility and flexibility on the processing of cross domain search, which only depends on the correlation between the resources and the Wikipedia concepts. Our experiment results show that our proposal dramatically improves the search quality, as well as presents considerable potentials on the enhancement of web 2.0 applications.

Acknowledgment

The research and system development reported in this paper was supported by Singapore MDA grant R-252-000-376-27.

References

[1] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys*, vol. 40, no. 2, pp. 1–60, Apr. 2008.

[2] N. Rasiwasia, J. C. Pereira, E. Coviello, G. Doyle, G. R. G. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval." in *ACM Multimedia*, A. D. Bimbo, S.-F. Chang, and A. W. M. Smeulders, Eds. ACM, 2010, pp. 251–260.

[3] J. Magalhães, F. Ciravegna, and S. M. Rüger, "Exploring multimedia in a keyword space," in *ACM Multimedia*, A. El-Saddik, S. Vuong, C. Griwodz, A. D. Bimbo, K. S. Candan, and A. Jaimes, Eds. ACM, 2008, pp. 101–110.

[4] G. Qi, C. C. Aggarwal, and T. Huang, "Towards semantic knowledge propagation from text corpus to web images," in *WWW*, S. Srinivasan, K. Ramamritham, A. Kumar, M. P. Ravindra, E. Bertino, and R. Kumar, Eds. ACM, 2011, pp. 297–306. [Online]. Available: http://doi.acm.org/10.1145/1963405.1963449

[5] A. Mathes, "Folksonomies - cooperative classification and communication through shared metadata," 2004. [Online]. Available: http://www.adammathes.com/academic/computermediated-communication/folksonomies.html

[6] M. Sahami and T. D. Heilman, "A web-based kernel function for measuring the similarity of short text snippets," in *Proceedings of the 15th international conference on World Wide Web*, ser. WWW '06. New York, NY, USA: ACM, 2006, pp. 377–386. [Online]. Available: http://doi.acm.org/10.1145/1135777.1135834

[7] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis," in *In Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 1606–1611. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.76.9790

[8] J. Hu, G. Wang, F. Lochovsky, J. tao Sun, and Z. Chen, "Understanding user's query intent with wikipedia," in *WWW '09: Proceedings of the 18th international conference on World wide web*. New York, NY, USA: ACM, 2009, pp. 471–480. [Online]. Available: http://portal.acm.org/citation.cfm?id=1526773

[9] S. Cronen-townsend and W. B. Croft, "Quantifying query ambiguity," Jun. 17 2002.

[10] B. He and I. Ounis, "Query performance prediction," *Inf. Syst*, vol. 31, no. 7, pp. 585–594, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.is.2005.11.003

[11] T. Deselaers, D. Keysers, and H. Ney, "Features for image retrieval: A quantitative comparison," in *DAGM*, 2004, pp. 228–236.

[12] C. Liu, B. C. Ooi, A. K. Tung, and D. Zhang, "Crew: cross-modal resource searching by exploiting wikipedia," in *Proceedings of the international conference on Multimedia*, ser. MM '10. New York, NY, USA: ACM, 2010, pp. 1669–1672. [Online]. Available: http://doi.acm.org/10.1145/1873951.1874318

[13] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Transactions on Information Systems*, vol. 20, no. 4, pp. 422–446, Oct. 2002.

[14] J. Jeon, V. Lavrenko, and R. Manmatha, "Automatic image annotation and retrieval using cross-media relevance models," 2003.

[15] C. Bizer, R. Cyganiak, and T. Heath, "How to publish linked data on the web," Web page, 2007, revised 2008. Accessed 07/08/2009. [Online]. Available: http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/

[16] G. T. et al, "Sindice.com: Weaving the open linked data," in *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC2007)*, ser. LNCS, vol. 4825. Berlin, Heidelberg: Springer Verlag, November 2007, pp. 547–560.

[17] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "DBpedia - A crystallization point for the web of data," *J. Web Sem*, vol. 7, no. 3, pp. 154–165, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.websem.2009.07.002

[18] "http://www4.wiwiss.fu-berlin.de/flickrwrappr/."

[19] "Linked data tagging with lodr."

[20] J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov, "Silk — A link discovery framework for the web of data," in *2nd Workshop on Linked Data on the Web*, Madrid, Spain, Apr. 2009.

[21] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

[22] Y. Jing and S. Baluja, "Pagerank for product image search," in *WWW*, J. Huai, R. Chen, H.-W. Hon, Y. Liu, W.-Y. Ma, A. Tomkins, and X. Zhang, Eds. ACM, 2008, pp. 307–316. [Online]. Available: http://doi.acm.org/10.1145/1367497.1367540

[23] D. Joshi, J. Z. Wang, and J. Li, "The Story Picturing Engine—a system for automatic text illustration," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 2, no. 1, pp. 68–89, Feb. 2006.

[24] Y. Yang, N. Bansal, W. Dakka, P. G. Ipeirotis, N. Koudas, and D. Papadias, "Query by document," in *WSDM*, 2009, pp. 34–43.

[25] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys*, vol. 40, no. 2, pp. 1–60, Apr. 2008.

[26] Rubner, Tomasi, and Guibas, "The earth mover's distance as a metric for image retrieval," *IJCV: International Journal of Computer Vision*, vol. 40, 2000.

[27] G. Wu, E. Y. Chang, and N. Panda, "Formulating context-dependent similarity functions," in *ACM Multimedia*, H. Zhang, T.-S. Chua, R. Steinmetz, M. S. Kankanhalli, and L. Wilcox, Eds. ACM, 2005, pp. 725–734. [Online]. Available: http://doi.acm.org/10.1145/1101149.1101307

[28] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: Semantics-sensitive integrated matching for picture LIbraries," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 23, no. 9, pp. 947–963, 2001. [Online]. Available: http://www.computer.org/tpami/tp2001/i0947abs.htm

[29] P. Duygulu, K. Barnard, J. F. G. de Freitas, and D. A. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," in *ECCV*, 2002, p. IV: 97 ff. [Online]. Available: http://link.springer-ny.com/link/service/series/0558/bibs/2353/23530097.htm

[30] X. Fan, X. Xie, Z. Li, M. Li, and W.-Y. Ma, "Photo-to-search: using multimodal queries to search the web from mobile devices," in *Multimedia Information Retrieval*, H. Zhang, J. R. Smith, and Q. Tian, Eds. ACM, 2005, pp. 143–150. [Online]. Available: http://doi.acm.org/10.1145/1101826.1101851

[31] T. Yeh, K. Tollmar, and T. Darrell, "Searching the web with mobile images for location recognition," in *CVPR (2)*, 2004, pp. 76–81.

[32] T. Yeh, J. J. Lee, and T. Darrell, "Photo-based question answering," in *ACM Multimedia*, A. El-Saddik, S. Vuong, C. Griwodz, A. D. Bimbo, K. S. Candan, and A. Jaimes, Eds. ACM, 2008, pp. 389–398. [Online]. Available: http://doi.acm.org/10.1145/1459359.1459412