Context-Aware Social Media Recommendation Based on Potential Group

Cheng Zeng[1], Dawen Jia[1,+], Jian Wang[1], Liang Hong[1], Wenhui Nie[1], Zhihao Li[1], Jilei Tian[2]
[1]State Key Lab of Software Engineering, Wuhan University, China, 430072
[2]Nokia Research Center, Beijing, China, 100176

ABSTRACT

Data recommendation as a kind of active mode is more meaningful and important than traditional passive search mode in social media environment. The importance of contextual information has also been recognized by researchers and practitioners in many disciplines, including recommendation system, e-commerce, information retrieval, mobile computing and so on. In this paper, we propose a novel approach for context-aware social media recommendation via mining different granularities of potential groups, called *Common Preference Group (CPG)*. Intuitively, *CPG* mining is to cluster those users who are interested in any topic set with certain context and have similar affection degree for each topic in the set. It means each user could belong to multiple *CPG* corresponding to different topic sets. The approach absorbs the characteristic of Collaborative Filtering (*CF*) technique but overcomes its defects, such as cold-start, data sparseness. Moreover, we build the *Tag-Feature Semantic-pairs (TFS)* to represent the semantic topics implied in media object to improve the accuracy of *CPG* mining. To evaluate the efficiency and the accuracy of our approach, we use two datasets: D_e is a simulated dataset and D_p is a real-life corpus collected from Flickr. The experimental results show the superiority of our approach for social media recommendation.

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

General Terms: Algorithms, Experimentation.

Keywords: Social media recommendation, Common Preference Group, Group recommendation, Context-aware

1. INTRODUCTION

Social media is the use of web-based and mobile technologies to turn communication into interactive dialogue, namely media for social interaction. User-orientation is a salient characteristic of social media sites which allow the creation and exchange of user-generated content. At present, Google Ad planner claims that YouTube has 1,500 million unique users and Flickr has 88 million unique users. Besides, Flickr has more than 5 billon images currently and the numbers of users and media objects are rapidly increasing every day. Therefore, social media recommendation as a kind of active mode is more meaningful and important than traditional passive search mode.

Corresponding author: brilliant@whu.edu.cn

ContextDD'12, August 12, 2012, Beijing, China.

As a matter of fact, the most popular social media sites (e.g. Youtube, Flickr) provide the group mechanism, where the user can manually create groups for media sharing and recommendation. [1] pointed out that more than half of Flickr's users participated in at least one group, which indicated that a large number of users engaged in group activities. Users create and join groups for social purposes. The formation of groups has gained great popularity and attracted an enormous number of users [2]. Basically, each group represents one/many common topics and users who are interested in this topic can join the group as a member and upload the relevant media data into the group. Study shows adding photo into group was one of the main reasons for photo diffusion [9, 10]. It is no doubt that the group is a useful mechanism for media sharing and recommendation.

In addition, with the rapid progress of mobile device, the rich context information captured by the mobile device can be used to understand user preference and record the environment of created media object, which can be related with user's demographic/social information and bring a great business value, such as targeted advertising, data-driven user studies for marketing and personalized recommendation. So the context information is considered in this paper to enrich the connotation of group mechanism. Although the context information has been utilized for media recommendation, the relation between group and context information is not considered, namely how to mine and utilize the contextual group for social media recommendation.

In this paper, we propose a novel approach to automatically mine potential groups based on users' context-aware preferences for topics implied in media objects. To distinguish the group created manually in social media sites[1], we call the potential group mined automatically by our approach as *Common Preference Group* (*CPG*). In fact, the essence of a user marking "favorite" to a media object is that the user is interested in some topics implied in the media object, and the favorite is even relevant to the user's context, such as time and scene. So we can regard user's preference as the affection degrees for some contextual topics instead of the media objects. Intuitively, a *CPG* is a set of users who share the common preference, in other words, they have the similar affection degree on each topic in certain context.

For enhancing the clarity of our approach, we give a simple example as follows. Table 1 represents the relations between favorite image topics and context for user u_1, u_2, u_3, u_4 where we only consider one dimension of context: time, though our approach can support n-dimensions context.

The numbers in table 1 denote the users' affection degree

for corresponding topics in certain context and could be respectively interpreted as dislike/unaware, general, like and crazy. Traditional *CF* technique ignores the context information so that u_1 and u_4 are regarded as users with most similar preferences and flower image of u_1 uploading will be recommended to u_4. But in fact, u_1 and u_4 pay attention to the same topic images in different contexts. And u_4 only browses images at night, recommending images to him in the daytime is unsuited. Furthermore, although the entire preference of u_1, u_2 and u_3 are dissimilar in *CF*'s idea, they have preferences intersection in topic set {tiger, car} with even the same context. So a new *CPG* will be generated as <{2•tiger,3•car}, $\{u_1,u_2,u_3\}$> and all resources and preferences will be precisely shared and recommended each other in the *CPG*. If we directly utilize current grouping mechanism, the recommended results will be unpredictable because a user is possible to join a group which just includes the topics attracting him.

Table 1. The relation between topics and contexts

u_1:

	tiger	sea	flower	car	bridge
day	2	0	1	0	1
night	0	0	0	3	0

u_2:

	tiger	sea	flower	car	bridge
day	2	0	3	0	1
night	0	1	0	3	0

u_3:

	tiger	sea	flower	car	bridge
day	2	3	0	0	0
night	0	0	0	3	0

u_4:

	tiger	sea	flower	car	bridge
day	0	0	0	0	0
night	2	0	0	3	1

The features of the *CPG* bring much superiority for social media recommendation. First, users could discover and distinguish those people who have the similar preferences with them in some aspects and these preferences are even related to certain contexts. Second, users' potential preferences could be predicted or inferred from other member's behaviors of the same *CPG*. More specifically, *CPGs* could flexibly represent any combination of topics and contexts that overcome the flaws of *CF* technique, such as cold-start, data sparseness, e.g. a new user without any rating history could still obtain media recommendation if he has the profile information or directly join a *CPG*. Lastly, different *CPG* can be recommended to users based on user's context-aware preference and the social media objects can also be distributed to different *CPG* without human involving. In order to evaluate the accuracy and efficiency of our approach, we use the real dataset extracted from the most popular image sharing site Flickr and randomly generated dataset, respectively. The detail will be elaborated in the experimental section.

2. RELATED WORKS

We compare our work with related domains from three different viewpoints.

- **Recommendation Techniques**

There exist different kinds of recommendation techniques for various user tasks. Those techniques have a number of possible classifications, such as content-based [8, 19], *CF* [5, 12, 14], demographic [7], hybrid [4, 13] etc. *CF* is one of the most successful recommendation techniques. However it suffers from the weakness such as cold-start, data sparseness. Cui et al.[6] propose a Feature Interaction Graph (*FIG*) approach to fuse text, visual content, user and their correlations in social media objects to facilitate the recommendation application. The approach directly utilizes the user relation in group but ignores the disadvantages of current group mechanism. When applying these recommendation techniques in social media environment, we find a common fault that they recommend each object to the individual user. It will be time-consuming.

- **Group Recommendation**

The most mentioned problem in current social media recommendation domain based on group is that groups are self-organized so that a topic may correspond to lots of groups and a user may join many groups. The problem causes disordered sharing and recommendation, e.g. a user may receive many repetitive recommendations from different groups but miss other sharing. So lots of research focus on recommending groups to each user [3,17] or recommending groups for a given media object according to media content [2,4,5]. There into, [2] uses both visual content and textual annotations for group classifying and object recommendation, the idea is similar to the topic extraction in this paper. Though those approaches could recommend the best groups for users and media objects, most users with common interests may still separate in different groups and can not adequately share media objects each other.

- **Recommendation Based on Context Information**

Researchers have considered in bringing the context information into the field of multimedia recommendation [3, 11, 16]. The main reason is that users' partial interests might be strongly related to the context information. Therefore, the aim of those approaches is to recommend appropriate media objects with respect to the context, and then the personalized recommendation under certain context can be fulfilled.

Obviously, the group mechanism which could realize batch recommendation has higher efficiency since the number of groups could be much less than the number of users. However, both accuracy and coverage of the group mechanism are low since it emphasizes more on the common preference for most users, while neglects some personalized requirements caused by the difference among users. Recommendation approaches based on context can provide personalized functions to users, especially in the mobile internet era, but they lack appropriate reasoning and take no account of the common features among users. Most *CF* approaches infer the user's preference with the common information of similar users. The native defects of *CF* approach affect its efficiency and accuracy.

In this paper, we synthesize the superiorities of three kinds of approaches above by mining potential groups as shown in Fig.1. *CPGs* are automatically generated for those users who share common preferences under certain contexts together and even have similar demographic information in

some aspects. Both new users and media objects could be automatically added into the corresponding *CPGs* as well.

Fig. 1. Relation between our approach and traditional approaches

3. PRELIMINARIES

3.1 User Preference Modeling

As mentioned above, the essence of user preference is the topics implied in media objects. So we first extract the topics from media objects (discussed in Appendix A). Generally, more than one topic could be extracted from a media object. *TopicSpace* denotes all representative topics extracted from the whole media objects set. Besides, the context information is meaningful but it has different implications for users and media objects. As for the media object, we pay attention to the shot time and location which could contribute to context-aware recommendation. For example, a photo related to a certain location can be recommended to a user when he enters the adjacent regions. For the users, context can be classified as static information (such as age, gender and profession) and dynamic information (such as time and location). The most typical difference between two kinds of context information is that the value of static context is single-valued and stable, while that of dynamic context is multiple-valued and variable. Users' context information can provide deeper personalized recommendation. For example, a game propaganda photo can be recommended to a boy in school during his lunch break, because he and other persons with similar profiles with him used to browse this topic of photos in current time or location.

The essence of a media object *o* is a combination of context information set and topic set implied in *o*. The basis of mining *CPG* is to effectively structure the relations, namely User Preference Model (*UPM*), among user, context, topic and affection degree. Different contexts can be selected to build different dimensions of *UPM* according to the practical application request.

Definition 1: ***User Preference Model (UPM)***

User Preference Model (UPM) records the affection degree that the user favors the topics under certain contexts. *UPM* can be represented as a 5-tuple $<U, C, T, F, \Lambda>$, where $U, C = \{c_i | i = 1,2, \dots, N\}$, $T=\{t_j | t_j \in TopicSpace\}$, $F= \{\lambda_j\}$ denote user set, context set, topic set and corresponding affection degrees, respectively, $\Lambda \subseteq U \times C \times T \times F$ indicates the user preference relation among user, topic and affection degree under a certain context set. Please note that *C* can be multi-dimensional and they are orthogonal each other.

3.2 *CPG* Modeling

Definition 2: ***Common Preference Group (CPG)***

A *Common Preference Group (CPG)* is a subset of *UPM*, which satisfies the condition that all users in a *CPG* sharing the same $u_{preference} = \{< \{c_i\}, t_j, \lambda_j >\}$. A *CPG* can be represented as a pair $<U_i, cp(U_i)>$, where U_i is a user set and $cp(U_i) = \cap_{u \in U_i} u_{preference}$ denotes the common preferences of all users in U_i.

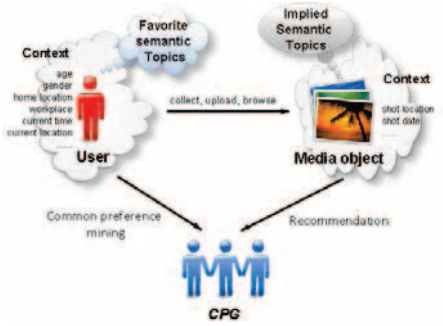


Fig.2. The relationships among user, media object and group

Although we do not refer to media objects in the two definitions above, *UPM* and *CPG* have close relation with media objects. User's interests are reflected by the topics implied in his favorite media objects. It means that the topic is the bridge between user and media object. So each *CPG* contains those media objects whose topics are favored by all users in this *CPG*. The relationships among users, media objects and groups can be represented as shown in Fig.2.

4. CPG MINING

	t_1	t_2	t_3	t_4
u_1	d1	d2	d1	0
u_2	d1	0	0	d3
u_3	d1	d2	d1	d3
u_4	d1	d3	d3	d3
u_5	d2	d3	d3	d1
u_6	d2	0	0	0

(a)The User Preference Relation(*UPR*)

	t_1	t_2	t_3	t_4
u_1	d1	d2	d1	0
u_2	d1	0	0	d3
u_3	d1	d2	d1	d3
u_4	d1	d3	d3	d3
u_5	d2	d3	d3	d1
u_6	d2	0	0	0

(b)Common Preference Group(*CPG*)Mining

	t_1	t_2	t_3	t_4
g_1	d1	d2	d1	0
g_2	d1	0	0	d3
g_3	0	d3	d3	0

(c)The Group Preference Relation(*GPR*)

	u_1	u_2	u_3	u_4	u_5	u_6
g_1	1	0	1	0	0	0
g_2	0	1	1	1	0	0
g_3	0	0	0	1	1	0

(d) The User-Group Relation(*UGR*)

Fig.3. An example of *CPG* mining

Definition 3: ***CPG Mining***

Given a user preference model $UPM=< U, C, T, F, \Lambda >$, find a state $<G, GPR, UGR>$ that is consistent with *UPM*, such that for any state $<G', GPR', UGR'>$ that is consistent with *UPM*, $\#G \leq \#G'$, where *G*, *G'* is a set of common preference groups, $GPR, GPR' \subseteq G \times C \times T \times F$ is the *group-context-topic-degree* relation, and $UGR, UGR' \subseteq U \times G$ is the *user-group* assignment relation. A state is consistent with *UPM*, if and only if every user in *U* has the same set of *context-topic-degree* relation as in *UPM*.

We define two parameters *minUsers* and *minTopics* to denote the minimal numbers of user and topic to form a *CPG*, respectively. Then we give a simple example to demonstrate *CPG* mining approach. We define a user set $U = \{u_1, \dots, u_6\}$, a topic set $T = \{t_1, \dots, t_4\}$ and both *minUsers* and *minTopics* are equal to 2. As depicted in Fig.3(a), the affection degrees

$\lambda_i \in \{d_1, d_2, d_3\}$ of topic for each users are recorded in the user preference relation Λ, where λ_i is an integer and the bigger value indicates that the user favors more on the topic, The next step is to mine *CPGs* from Λ. As shown in different shape regions in Fig.3(b), three *CPGs* are mined. g_1 = <{t_1d_1,t_2d_2,t_3d_1}, {u_1,u_3}>, g_2 = <{t_1d_1,t_4d_3}, {u_2,u_3,u_4}> and g_3 = <{t_2d_3,t_3d_3}, {u_4,u_5}>. *GPR* and *UGR* can be easily inferred with *CPG*, which indicate the relationship between *CPGs* and topics, and the relationship between *CPGs* and users, respectively. E.g., as shown in Fig.3(c)(d), g_1 has GPR_1= <g_1, {t_1d_1,t_2d_2,t_3d_1}>, UGR_1 = <g_1, {u_1,u_3}>. It is obvious that *GPR* could guide new user or media object to assign to suitable *CPGs* and *UGR* could help user to make friends or discover subconscious interests indirectly.

4.1 The *CPG* Mining Algorithm

We elaborate the process of *CPG* mining algorithm step by step with the example given in Fig.3.

Step 1: Clustering users based on topic preference

With *UPR* in Fig.3 (a), we can easily cluster the users who have the common preference based on their affection degrees for each topic (shown in Table 2) and those elements in the cells with the same column do not have user intersection each other.

Table 2. User clustering table

	t_1	t_2	t_3	t_4
d_1	u_1,u_2,u_3,u_4		u_1,u_3	u_3(delete)
d_2	u_5,u_6	u_1,u_3		
d_3		u_4,u_5	u_4,u_5	u_2,u_3,u_4

Table 3. Intersection of $p(d_1t_1)$

	t_1	t_2	t_3	t_4
d_1	u_1,u_2,u_3,u_4		u_1,u_3	
d_2		u_1,u_3		
d_3		(Delete) u_4,u_5	(Delete) u_4,u_5	u_2,u_3,u_4

First, each element is regarded as a candidate *CPG*. We delete the elements in which the number of user is less than *minUsers* and add the rest to array *initialList*, *e.g.*, the element $p(d_1t_4)$ will be deleted if *minUsers* = 2.

Step 2: Mining CPG with multiple common topics

The *CPG* with single topic is not meaningful so that we merge the elements in Table 2 to form *CPG* with multiple common topics. For each element $p(d_it_j)$ in *initialList*, we implement intersection operations between it and those elements with different columns t_j. We take the first element $p(d_1t_1)$ as an example. Table 3 shows the results of intersection operations between $p(d_1t_1)$ and other elements. We likewise delete the results in which the number of user is less than *minUsers*, and put the effective results into the list *intersList*. For each element in *interList*, we merge it with elements in the candidate *CPG* List (*cCList*) and put itself into *cCList* as well.

Take $p(d_1t_1)$ as an example, $intersList(d_1t_1)$={<{d_2t_2}, {u_1,u_3}>, <{ d_1t_3}, {u_1,u_3}>, <{ d_3t_4}, {u_2,u_3,u_4}>}.

(1) For the element <{d_2t_2}, {u_1,u_3}> in $intersList(d_1t_1)$, we try to merge it with those elements in *cCList* by **MergeWithcCList** algorithm, but *cCList* is empty at this moment. So we add <{d_1t_1,d_2t_2}, {u_1,u_3}> into *cCList*.

(2) For the element <{d_1t_3}, {u_1,u_3}> in $intersList(d_1t_1)$, we merge it with current element in *cCList*, namely {<{d_1t_1,d_2t_2},{u_1,u_3}>}. Both the new element {<{d_1t_1,d_2t_2,d_1t_3},{u_1,u_3}>}, which satisfies *CPG* restraint condition, and <{d_1t_1,d_1t_3},{u_1,u_3}> will be added into *cCList*. We revoke ***Step 3*** to remove those elements which are fully overlapped by other elements in the same *cCList*. In this example, both <{d_1t_1,d_2t_2}, {u_1,u_3}> and <{d_1t_1,d_1t_3}, {u_1,u_3}> are removed, since they are fully covered by <{d_1t_1,d_2t_2,d_1t_3}, {u_1,u_3}>.

(3) For the element <{d_3t_4}, {u_2,u_3,u_4}> in $intersList(d_1t_1)$, we likewise merge it with element <{d_1t_1,d_2t_2,d_1t_3}, {u_1,u_3}> in *cCList*. However, the number of user in the new element <{d_1t_1,d_2t_2,d_1t_3,d_3t_4}, {u_3}> is less than *minUsers*, so that it will not be added into *cCList*. We only add <{d_1t_1,d_3t_4}, {u_2,u_3,u_4}> into *cCList*.

(4) Finally, it returns *cCList* = {<{d_1t_1,d_2t_2,d_1t_3}, {u_1,u_3}>, <{d_1t_1,d_3t_4 }, {u_2,u_3,u_4}>}

Step 3: Remove fully overlapped elements in cCList

An element will be deleted if it is fully covered by another element in the same *cCList*.

Step 4: For each element in initialList, repeat Step 2

We repeat *step 2* to get a *cCList* for each element in *initialList,* shown in Table 4. Since the element in *cCList* mined from $p(d_2t_2)$ is covered by the element <{d_1t_1,d_2t_2,d_1t_3},{u_1,u_3}>, it will be deleted. Actually, if the users in element p are the subset of the users in p' which appears before p, *cCList* mined from p is redundant. We could mine <{d_2t_2,d_1t_3},{u_1,u_3}> by *step 2*. However, the user set of $p(d_2t_2)$ is the subset of $p(d_1t_1)$, we even need not to mine *cCList* of $p(d_2t_2)$. Finally, we add all elements in *cCList* to *CPGList*. We can easily generate *GPR* and *UGR* by *CPGList*.

Table 4. All cCLists

P	*cCList*
$p(d_1t_1)$	{<{d_1t_1,d_2t_2,d_1t_3}, {u_1,u_3}>, <{d_1t_1,d_3t_4 }, {u_2,u_3,u_4}>}
$p(d_2t_1)$	Φ
$p(d_2t_2)$	{<{d_2t_2, d_1t_3}, {u_1,u_3}>}(*Delete*)
$p(d_3t_2)$	{<{ d_3t_2,d_3t_3}, {u_4,u_5}>}
$p(d_1t_3)$	Φ
$p(d_3t_3)$	Φ

Step 5: CPG merging

We've already got all *CPGs* that we could mine from *UPR* in *Step 4*. But at this moment, all *CPGs* are actually original common preference groups, called *Ori-CPG*, which is sensitive to the difference among user preferences. It means all user preferences are completely consistent in the same *CPG*. In this paper, we absorb the advantages of *CF* technology and take it into *CPG* mining algorithm, called *CF-CPG*. It can tolerate the slight difference among user preferences, realize the preference inferring in certain extent and thus gain a better recommendation effect. For a more concise description, we do not distinguish *CF-CPG* from *Ori-CPG* in an elaborate and uniform used concept of *CPG*.

Table 5 gives a simple example of the situation where the divergence between CPG_1 and CPG_2 is only about topic t_1 and the difference of referred users set is very small. So we merge the two groups to obtain a bigger range of sharing and recommendation. It is similar to the idea of *CF*, but the operated target is a group instead of a single user. As a result, the recommendation efficiency will be higher due to the number of *CPG* is far less than that of users and *CPG* actually prestores the preference relation among users.

Table 5. The situation of *CPG* merging

CPG_1

	t_1	t_2	t_3	t_4
u_1	d_2	d_1	d_3	d_1
u_2	d_2	d_1	d_3	d_1
u_3	d_2	d_1	d_3	d_1
u_4	d_2	d_1	d_3	d_1
u_5	0	d_1	d_3	d_1
u_6	0	d_1	d_3	d_1

CPG_2

The principle of *CPG* merging follows the formula (1). When the similarity value $H(g,g')$ is more than a certain threshold, g and g' will be merged and the merging method is to find the minimal rectangle containing them. It means some users in the new *CPG* will be automatically recommended those preferences which they did not have before. We suppose the number of users in g', namely $U(g')$, is more than that in g, namely $U(g)$, and then the comparison about the number of topic is inverse, otherwise g' will contain g, or they have not intersection at all.

$$H(g,g') = \prod\left(\frac{Sum(\Phi\cap\Phi')}{Sum(\Phi\cup\Phi')}\right)\cdot\frac{1}{\sum_{i\in\{T\cup T'-T\cap T'\}}Sub(F_i,F_i')}$$

$$Sub(F_i,F_i') = |\sum_{j\in U^\wedge}F_{ij}'/Sum(U^\wedge) - F_i|$$

$$U^\wedge = U(g') - U(g) \qquad (1)$$

where Φ,Φ' denote the user/topic sets of group g and g', respectively. *Sum*(*) is the cardinality of the set. $(T\cup T'-T\cap T')$ represents the difference of topics for g and g', in other words, it is a topics set which have different affection degrees for the same topic. The intuitional means of formula (1) is to calculate the similarity between two *CPG* which is proportional to their common user, topic and affection degree. Although it seems that the computation complexity of formula (1) is high, we actually only compute a part of it every time. For example, we can directly stop merging if the intermediate result has been lower than the threshold, because each part, such as $\frac{1}{\sum_{i\in\{T\cup T'-T\cap T'\}}Sub(F_i,F_i')}$, $\frac{Sum(\Phi\cap\Phi')}{Sum(\Phi\cup\Phi')}$ or $\Phi\in T/U$, is a decimals and their product can be only smaller.

4.2 Context-aware *CPG* Mining

When we take the context information into consideration, the user preference relation Λ could be different as long as the context information varies. However, a user has the unique affection degree for each topic in a certain context. We discuss the multidimensional context-aware *CPG* mining method in this subsection.

Suppose there exists n-dimensional context information $C=C_1\times C_2\times\ldots\times C_n$. For each $C_i\in C$, we can establish a Λ and mine a *CPG* set corresponding to this context. Therefore, as shown in Table 6, we improve our *CPG* mining method to generate new *CPG* which has multiple context conditions.

Table 6. *CPG* mined from different context

C_1	C_2	C_3	...	C_n
C_1-CPG_1	C_2-CPG_1	C_3-CPG_1	...	C_n-CPG_1
C_1-CPG_2	C_2-CPG_2	C_3-CPG_2	...	C_n-CPG_2
...	...	...	...	...
C_1-CPG_W	C_2-CPG_X	C_3-CPG_Y	...	C_n-CPG_Z

As we know, Table 2 clusters users based on their affection degrees for each topic and we could mine a set of *CPGs* based on it. Formula (2) shows the variable mapping between Table 2 and Table 6. We could apply the similar process of the basic *CPG* mining approach to mine the context-aware *CPG* based on Table 6.

$$\begin{cases} C_i \rightarrow t_i \\ C_i\text{-}CPG_j \rightarrow \text{User clusting element } p_j \text{ in column } i \end{cases} \qquad (2)$$

There are 3 variables in Table 2: topics t, affection degrees λ and user clustering elements p. Since the variable λ is only used to cluster the users, and users in Table 6 have already been clustered, variable λ can be ignored.

As shown in Fig.4, there are three different contexts C_1, C_2, C_3. We can mine C_1-CPG_1 = <{t_1d_2,t_2d_3}, {u_1,u_2,u_3}>, C_2-CPG_1 = <{t_1d_3,t_2d_2,t_3d_1}, {u_2,u_3}> and C_3-CPG_1 = <{t_1d_2,t_2d_1}, {u_2,u_3}>, C_3-CPG_2 = <{t_2d_1,t_3d_2}, {u_1,u_2}>.

So two new context-aware *CPG* can generate: $contextCPG_1$ = <{$C_1(t_1d_2,t_2d_3)$, $C_2(t_1d_3,t_2d_2,t_3d_1)$, $C_3(t_1d_2,t_2d_1)$}, {u_2,u_3}>; $contextCPG_2$ = <{$C_1(t_1d_2,t_2d_3)$, $C_3(t_2d_1,t_3d_2)$}, {u_1,u_2}>.

The essence of the context-aware *CPG* is a set of users who share the common preferences in same context. For example, $contextCPG_2$ refers to two users u_1, u_2 who have the common affection degrees on topic t_1 and t_2 in the context C_1, and have the common affection degrees on topic t_2 and t_3 in the context C_3, respectively.

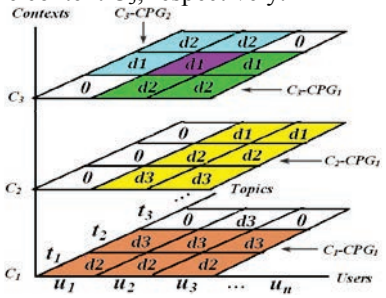


Fig.4. An example of context-aware *CPG* mining

5. CPG UPDATING

When new media objects or users join the system, they will be added into the corresponding *CPG*. How to select suitable *CPG* is a key problem.

5.1 User Updating

As mentioned in section 3, a user contains a preference set with context-aware topics while a *CPG* contains the

common preference fragments of many users and some media objects with the same contextual topics. When we choose suitable *CPG* to assign new user, the essence is the similarity calculation between their preference sets. However, there are lots of new users every day and large-scale potential *CPG* for current social media sites. The similarity calculation will be time-consuming. So we first filter those *CPGs* which have little similarity.

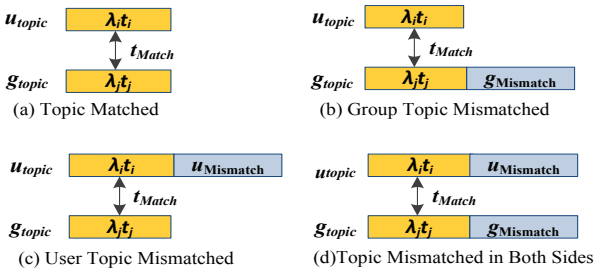


Fig.5. Different matching types between u_{topic} and g_{topic}

We define u_{topic} as a set of topics t_i favored by a user u and g_{topic} as a set of topics t_j implied in a *CPG* g. We calculate the similarity between u_{topic} and g_{topic}, and then filter those *CPGs* whose corresponding similarity values are less than a certain threshold. As depicted in Fig.5, there exist four different matching cases between u_{topic} and g_{topic}.

The similarity of u_{topic} and g_{topic} is determined by *Positive Score* (P_{score}) and *Negative Score* (N_{score}). The formula of similarity score is given below where δ is a constant to avoid the denominator for zero.

$$Sim(u_{topic}, g_{topic}) = \frac{P_{score}}{\delta + N_{score}} \quad (3)$$

In Fig.5, the matched topics contribute to P_{score} and the mismatched topics contribute to N_{score}. The topic set t_{Match} includes all the topics in both u_{topic} and g_{topic}, namely $t_{Match} = u_{topic} \cap g_{topic}$, which contributes to P_{score}. In formula (4), α is the P_{score} value, which is equal to the sum of affection degrees of all matched topics in two sets.

$$\alpha = \sum_{t_v \in t_{Match}} \lambda_v \quad (4)$$

$\gamma_k (k = 1,2,3)$ indicate the factors in inverse proportion to $Sim(u_{topic}, g_{topic})$. In Fig.5(a), $u_{topic} = g_{topic}$, $N_{score} = \gamma_1$ which is calculated with formula (5) and indicates that the difference of weight λ between two matched topics sets contributes negative score to the similarity. In Fig.5(b), $u_{topic} \subset g_{topic}$, $N_{score} = \gamma_1 + \gamma_2$. γ_2 is equal to the sum of affection degrees of all mismatched topics in g_{topic}. In Fig. 5(c), $u_{topic} \supset g_{topic}$, $N_{score} = \gamma_1 + \gamma_3$. γ_3 is caused by the mismatch of user topics. Although formula (6) and (7) are similar, we still distinguish them since the difference of them will be used in media object updating. In Fig.5(d), $u_{topic} \cap g_{topic} \neq \emptyset$, $u_{topic} \not\subset g_{topic}$ and $u_{topic} \not\supset g_{topic}$, $N_{score} = \gamma_1 + \gamma_2 + \gamma_3$.

$$\gamma_1 = \sum_{t_i, t_j \in t_{Match} \text{ and } i \neq j} |\lambda_i - \lambda_j| \quad (5)$$

$$\gamma_2 = \sum_{t_i \in g_{Mismatch}} \lambda_i \quad (6)$$

$$\gamma_3 = \sum_{t_i \in u_{Mismatch}} \lambda_i \quad (7)$$

If $Sim(u_{topic}, g_{topic})$ is less than a certain filter threshold, the corresponding *CPG* will be directly filtered. Then we calculate the preference sets similarity between user and remaining *CPG*.

We suppose $\{c_i^m\}$ denotes the m^{th} context sets for user u's topic t_i, $\{c_j^n\}$ denotes the n^{th} context information sets for *CPG* g's topic t_j. The similarity calculation formula between user u and *CPG* g is as follows:

$$Sim(u, g) = \frac{\sum_{e \in Q} c_e}{Sum\ Q} \cdot Sim(u_{topic}, g_{topic}) \quad (8)$$

where $c_e = \begin{cases} 1 & if\ Sum(\{c_i^m\} \cap \{c_j^n\}) > \varphi\ when\ t_i = t_j \\ 0 & others \end{cases}$

In formula (8), Q denotes the set of all preferences in user u and *CPG* g. φ is a threshold to filter those preferences with same topic but little context similarity between u and g. c_e is represented as context similarity value between user and *CPG*. As a result, we will add the user u into those *CPG* where there is a high $Sim(u, g)$.

5.2 Media Object Updating

The topics of users and media objects in a *CPG* are similar though their contexts are possibly different. Media object updating is similar to user updating, but the difference is that each media object only contains single context information and the weights of implied topics are ignored. In the same way, we first calculate the topics similarity between media object and *CPG*.

We define o_{topic} as a topic set implied in the object o. We can calculate the topic similarity $Sim(o_{topic}, g_{topic})$ between media object o and *CPG* g with similar method as formula (3). Because the weights of topics in media object are ignored, the calculating methods for the cases in Fig.5(c)(d) will be changed which are reflected in the formula below:

$$\gamma_3 = \sum_{t_i \in o_{Mismatch}} avg\lambda \quad (9)$$

where $avg\lambda$ is the average value of λ in matched topics which will smooth the diversity. Then the context similarity between media object and *CPG* can be computed by the successive matching method because the media object context is single. Likewise, if the final similarity value is higher than a certain threshold, the media object will be added into the *CPG*.

It is obvious that we can use the same method of media object updating to realize media object recommendation. It is worth noting that media objects are only recommended to those relative *CPG* instead of users in traditional methods due to the specialty of *CPG*. This batch recommendation mechanism will obtain higher efficiency.

6. EXPERIMENTS AND ANALYSIS

The *CPG* mining, media object recommendation and *TFS* extraction algorithms are implemented in Java. We design and conduct a series of experiments to evaluate the efficiency and accuracy of the proposed approach. All experiments are conducted on a PC with 2.8 GHz CPU and 3 GB memory, running the Windows 7 operating system.

6.1 Data Preparation

To evaluate the performance of our approach, we adopt different data sets D_e and D_p for the efficiency and accuracy experiments, respectively. Although the approach proposed in this paper is general for any social media sites, the methods of media object processing and topic extracting for different modality of data are different. So we collect the real-life dataset as D_p from Flickr which is a popular social media site for image sharing and recommendation.

Discovering users' preference is the foundation of our *CPG* mining and social media recommendation approaches. Flickr provides a function which allows users express their interests by marking "favorite" to images and some images contains the context information such as shot time and location. Each user's "favorite" images set imply his/her interests. Therefore, we can utilize this function in Flickr for *CPG* mining and recommendation evaluation, i.e., the image in the "favorite" set is the correct recommendation.

Based on the above considerations, we first initialize a user set. The user set is determined by those users who marks one or many of Top-*k* most interesting images of each day from 2010.1 to 2010.6. We then eliminate the users who have favorite images less than 100 and larger than 1,000. Finally, we download all favorite images and relative tags, context information in the user set, and totally collect 1,238 users, 401,455 images and 4,000,399 tags. On average, each user marks 324 images as favorite and each image has 10 tags. We use 1/2 of user favorite images, namely those from 2010.1 to 2010.3, to extract topics and model the user's preference, and the rest images are used for recommendation evaluation.

Although the user scale is enough for the accuracy evaluation of *CPG* mining and media object recommendation approaches, it is insufficient for efficiency evaluation, so that we also randomly generate a more massive amount of users and relative attributes to build dataset D_e.

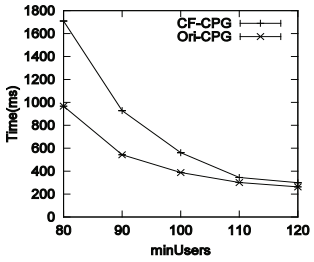
Besides, each user favors lots of topics with different affection degrees and *TFS* sets of different users are various. The computation complexity will be very high if we enumerate all possible *TFS* during *CPG* mining. So we select representative *TFS* by classical *LDA* (Latent Dirichlet Allocation) method [18] to construct *TopicSpace*. How to ascertain the number of dimensions of *TopicSpace* will be discussed in section 6.3.

6.2 Efficiency Evaluation

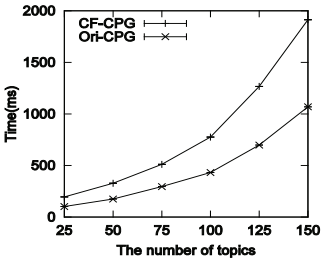
Due to no similar work about mining potential group, this section only shows the efficiency experiment results of our own approach. The experiment conducts based on two different *CPG* mining strategies: *Ori-CPG* and *CF-CPG*. The algorithms given in section 4 introduced the *Ori-CPG* strategy which mines the original *CPG* while the *CF-CPG* strategy performs additional merging process with *CF*'s idea. Both memory inverted index on mined *CPG* and hash technique are used to further improve the efficiency.

There are 6 parameters *U, C, T, F, minUsers, minTopics*, which have been explained in section 3, to affect the efficiency of the *CPG* mining algorithm. The quantification operation of affection degree for each topic is implemented to avoid excessive discrete of the data distribution in *UPM*. Otherwise, most of users would not have common preferences and the mined *CPG* would be very few as well.

We conduct a series of experiments to evaluate the efficiency of our *CPG* mining algorithm by the two strategies above. All experiments in this subsection are repeated 5 times for each case and the averaged values are employed.



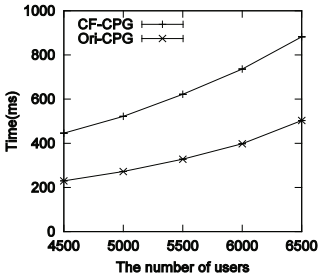
|T|=50, |U|=5000, |F|=3, |minTopics|=2;
Fig.6. Different *minUsers*



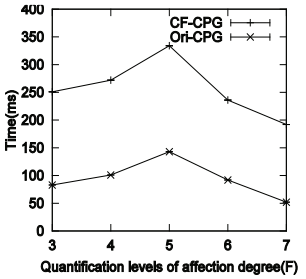
|U|=5000, |minUsers|=80, |F|=2, |minTopics|=2
Fig.7. Different number of *TFS*

Fig.6 shows the result of the time efficiency varies with parameter *minUsers*. We set the parameter $T = 50$, $F = 3$ and $U = 5{,}000$. The experimental result shows that: *Ori-CPG* strategy has a higher mining efficiency which is about twice as much as *CF-CPG* strategy because the latter has additional computation task.

The larger *TopicSpace* will lead to the richer topic implied in *CPG* set and more accurate user preference expression. However, it will also consume more computing time for *CPG* mining and media object recommendation. With the number of *TFS* increasing, the efficiency variation of the *CPG* mining algorithm is shown in Fig.7. It shows the computing time is in proportion to the number of *TFS*. Therefore, it is necessary to determine an appropriate range of *TopicSpace*, namely the number of representative *TFS* which will be discussed in section 6.3.



|T|=30,|minUsers|=80,|F|=4, |minTopics|=2;
Fig.8. Different user number



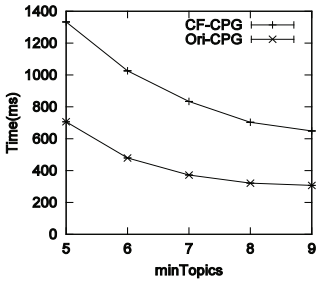
|T|=30,|U|=3000,|minUsers|=80, minTopics|=2
Fig.9. Different affection degree

Fig.8 shows the experiment result when T=30, *minUsers* = 80, $F = 4$ and user number varies from 4,500 to 6,500. The two curves which presents slight upward parabola indicates our approach has a good performance for the increasing of user number.

The experiment results in Fig.9 are different from others above. There is respectively a peak in the two curves. When the number of topics is 30, the peak value is 5. The reason of emerging peak value is that if quantification level is low, the personality of each user will be weakened and most users

tend to be same. As a result, both the number of mined *CPG* and calculating time are low. When the quantification level of affection degree is increased, the original difference among users is exposed so that *CPG* number and calculating time consuming rapidly rises. However, after the quantification level continues to increase, the difference among users is magnified. It is difficult to find those users with the same preferences in this status and both *CPG* number and calculating time decline sharply.

minTopics implies the abundance extent of topics for each *CPG*. If *minTopics* is very small, the *CPG* set will contain more *CPG* with pure topics. Fig.10 shows the *CPG* mining time variation with different *minTopics*. The time curve declines as a whole tendency but it slows down when *minTopics* is more than a threshold value.



|U|=5000, |minUsers|=500, |F|=3, |T|=100

Fig.10. Different *minTopic*

6.3 Accuracy Evaluation

We use F-measure $= \frac{2*precision*recall}{precision+recall}$ to evaluate the recommendation accuracy between our two *CPG* mining strategies and *CSP*[2], *FIG*[6]. Moreover, each *CPG* mining strategies considers the situations with context (*Ori-CPG'*, *CF-CPG'*) and without context (*Ori-CPG*, *CF-CPG*). It is worth noting that no more than 0.1% user profiles have more than 3 items in D_p. So we only utilize image context information in the experiment.

We take an image extracted from a user's favorite image set as the example. If the image can be recommended to the user, it will contribute to the precision of corresponding approach. At last, we calculate the accuracy of four approaches in terms of the proportion in D_p which are correctly recommended to the users who favorite them.

Table 7. Accuracy comparing of different approaches

	Ori-CPG	***CF-CPG***	***Ori-CPG'***	***CF-CPG'***	***CSP***	***FIG***
F-measure	*29.5%*	*37.3%*	*33.9%*	*40.2%*	*32.4%*	*36.1%*

Table 7 shows the comparing results, we can see *CF-CPG'* has the highest recommendation accuracy in the four approaches and *Ori-CPG'* has also good effect. Due to the specialty of our *CPG* mining approach, it overcomes the shortcoming of current group mechanism and discovers those potential groups with essentially common interest or preference among users, while the traditional groups used in *CSP* and *FIG* are built by subjective consciousness. In our experiment, we ignore the feedback learning in *CSP* in consideration of fairness. *Ori-CPG* mines potential groups based on the real situation completely, so that it could reflect the fact of common user preference. But it has not any inferring function and can not predict the variation of user preference. As a result, *Ori-CPG* obtains the lower accuracy than *CF-CPG* when we take the new data as verification. *FIG* obtains better accuracy than *Ori-CPG* because it adequately utilizes social relation network while *Ori-CPG* only use the common preference histories among users. However, *FIG* does not take inferring mechanism into account as well so that it falls behind *CF-CPG* for new data recommendation.

Due to the restricted dataset, we cannot evaluate the real inferring effect of *CF-CPG* which relies on the user feedback for the recommended objects. In fact, many media objects are possibly recommended to those potential users who do not reveal their total preferences and are even unaware of their own preferences. As a result, it would inevitably lower the accuracy evaluation of *CF-CPG*. Furthermore, the lack of user context information in current open social media datasets, the superiority of our *CPG* mining approach is limited.

7. CONCLUSION

In social media environment, group mechanism is useful for social media sharing and recommendation, but the flaws of exiting group mechanism becomes a severely obstacle. In this paper, we proposed a novel potential group mining approach to compensate for the flaws of current group mechanisms. With our approach, users who have the common interests or preferences under the same contexts will be automatically grouped together into a *CPG*. We also proposed an approach about *CPG* updating and the social media recommendation mechanism based on *CPG*. The experimental results indicate that our approaches can efficiently discover potential groups and gain preferable recommendation effect.

8. REFERENCES

[1] Radu Andrei Negoescu, Daniel Gatica-Perez: Analyzing Flickr groups. CIVR 2008: 417-426

[2] Jie Yu, Xin Jin, Jiawei Han, Jiebo Luo: Collection-based sparse label propagation and its application on social group suggestion from photos. ACM Transactions on Intelligent Systems and Technology 2(2): 12 (2011)

[3] Boutemedjet, S., Ziou, D., A graphical model for context- aware visual content recommendation. *IEEE Transactions on Multimedia*, 10(1):52–62, 2008.

[4] Justin Basilico, Thomas Hofmann: Unifying collaborative and content-based filtering. ICML 2004

[5] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, John Riedl: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. 22(1): 5-53 (2004)

[6] Bin Cui, Anthony K. H. Tung, et al: Multiple feature fusion for social media applications. SIGMOD Conference 2010: 435-446

[7] Mahmood, T., Ricci, F.: Towards learning user-adaptive state models in a conversational recommender system. In: A. Hinneburg (ed.) LWA 2007: 373–378.

[8] Jiajun Bu, et al. Music recommendation by unified hypergraph: combining social media information and music content. ACM Multimedia 2010: 391-400

[9] Kristina Lerman, Laurie Jones: Social Browsing on Flickr. International Conference on Weblogs and Social Media, 2007

[10] Roelof van Zwol: Flickr: Who is Looking? Web Intelligence 2007: 184-190

[11] Widisinghe, et al. picSEEK: Collaborative filtering for context-based image recommendation. International Conference on Information and Automation for Sustainability. (2010).

[12] Ioannis Konstas, Vassilios Stathopoulos, Joemon M. Jose: On social networks and collaborative recommendation. SIGIR 2009: 195-202

[13] Kazuyoshi Yoshii, et al.: Hybrid Collaborative and Content-based Music Recommendation Using Probabilistic Model with Latent User Preferences. ISMIR 2006: 296-301

[14] Badrul M. Sarwar, et al.: Item-based collaborative filtering recommendation algorithms. WWW 2001: 285-295

[15] Vincent Schickel-Zuber, Boi Faltings. OSS: A Semantic Similarity Function based on Hierarchical Ontologies. International Conferences on Artificial Intelligence, 2007

[16] Yu, Z., Zhou, X., Zhang, D., Chin, C.Y., et al., Supporting context-aware media recommendations for smart phones. *IEEE Pervasive Computing*, 5(3):68–75, 2006.

[17] Nan Zheng, Qiudan Li, Shengcai Liao, Leiming Zhang: Which photo groups should I choose? A comparative study of recommendation algorithms in Flickr. J. Information Science 36(6): 733-750 (2010)

[18] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet Allocation. Journal of Machine Learning Research, 3:993-1022, 2003.

[19] Mooney, R.J., & Roy, L.. Content-based book recommending using learning for text categorization. ACM Conference on Digital Libraries, pp.195-204(2000)

APPENDIX

A. SEMANTIC TOPIC EXTRACTION

Extracting topics from media object is a classic problem in multimedia domain [11,15]. Topics could be extracted from either media object or the rich text information surrounding the media object, such as tags, descriptions. The methods of extracting topics from different modalities of media (image, video and audio) are quite different. The accuracy of the topic extraction is the foundation of our approach. [2] shows that integrating both content feature and textual information will gain more accurate topic. Therefore, we build the *Tag-Feature Semantic-pairs (TFS)* as the topic of media object in this paper where $TFS =< \tau_i, v_j >$ and τ_i is a tag and v_j denotes the feature vector corresponding to a segmentation in a media object. The main procedure of building *TFS* includes two steps as follows.

A.1 Visual Word Recognition

To build *TFS*, we recognize middle level features "visual word" as the bridge between tag and feature content, which is generally generated by clustering image blocks. In this paper, we use the IAPR TC-12 dataset[1], denoted as D_v, as the knowledge base of visual word recognition. D_v includes 99,535 regions manually segmented from 20,000 images and the resultant regions have been annotated according to a predefined vocabulary including 275 labels. Each region corresponds to a 35 dimensions of feature vector.

For visual word recognition, we first segment each image into Z regions with Normalized Cuts Segmentation algorithm[2]. And then we extract 35 dimensions of feature vector from each region. The vector can be regarded as query condition to search in D_v where the vector set has been built hash index and all labels corresponding to returned vectors are potential visual words. For further improving the accuracy, we retain the Top-K similar labels for each image region as candidate visual words. And then we consider the coexisting probability between any two labels based on the rule that the more two labels appears in the same images, the higher the probability value is. Finally, the visual words in an image will be determined together when the sum of coexisting probability among labels is highest. The recognition formula for an image σ is represented below:

$$\Gamma_\sigma = \{w_\theta | \mathrm{Max}\left(\sum_{1<s,h<Z,1<i,j<K} P(R_s(w_i), R_h(w_j))\right)\} \quad (10)$$

Where $\theta \epsilon \{1,2,..,275\}$ denotes the finally selected label ID for image σ. $R_s(w_i)$ denotes the region s chooses label w_i as its visual word.

A.2 Mapping Visual Word to Image Tag

The semantic of visual word is abstract and sparse because there are only 275 labels in our knowledge base. On the contrary, the tags in social media sites can describe more abundant semantic. For example, user can mark an image "Tian An Men" shot in Beijing, but the visual word is just "Building". It is obvious that the recommendation will be not well. In our approach, we discover and utilize the relationship between feature content and tag to build *TFS* where visual word is viewed as the bridge between them.

At first, we will preprocess the tag set of all images. The *WordNet* stemmer is used to do stemming and delete those junk tags with frequency less than 5 times or in a stop word list. Then, a tag semantic similarity function based on *Yago*[3] is used to merge those tags with high similarity. The formula of tag semantic similarity function is as follows:

$$Sim(\tau_i, \tau_j') = \frac{log\left(1 + 2\hat{\beta}\left(\tau_j', \widehat{\tau_{ij}}\right)\right) - log(\hat{\alpha}(\tau_i, \widehat{\tau_{ij}}))}{max\ Sim}$$

$$\hat{\alpha}(\tau_i, \widehat{\tau_{ij}}) = APS(\widehat{\tau_{ij}})/APS(\tau_i)$$

$$\hat{\beta}(\tau_j', \widehat{\tau_{ij}}) = APS(\widehat{\tau_{ij}}) - APS(\tau_j')$$

$$APS(\tau) = \frac{1}{\rho_\tau + 2} \quad (11)$$

Where $\widehat{\tau_{ij}}$ is the lowest common ancestors of tag τ_i and τ_j' while $maxSim$ is their longest distance in *Yago*. *APS(τ)* and ρ_τ denote priori score and the descendant number of tag τ in *Yago*, respectively. Priori score is the probability that a tag is chosen in *Yago*. $\hat{\alpha}$ and $\hat{\beta}$ respectively correspond different processes of tag/concept generalization and specialization. Detailed mathematical introduction of formula (11) can be found in [15]. Finally, the calculation method of coexisting probability between visual words and tags in the same image is used, which is similar to formula (10). As a result, we will build the direct relationship between feature content and tag, namely *TFS*.

[1] http://www.imageclef.org/SIAPRdata

[2] http://www.seas.upenn.edu/~timothee/software/ncut/ncut.html

[3] http://www.mpi-inf.mpg.de/yago-naga/yago/