Invited Paper

Interactive Tutoring of Cooking Activities with Personalized Multimedia Recipe Search

Qing Li
Dept of Computer Science
City University of Hong Kong
Kowloon, Hong Kong
itqli@cityu.edu.hk

Xudong Mao
Dept of Computer Science
City University of Hong Kong
Kowloon, Hong Kong
xudonmao@cityu.edu.hk

Haoran Xie
Dept of Computer Science
City University of Hong Kong
Kowloon, Hong Kong
hrxie2@student.cityu.edu.hk

ABSTRACT
With the fast development of the Internet, there are more and more learning resources including various multimedia resources on the Web. Meanwhile, e-learning becomes more popular and important due to its convenience and autonomy. However, how to help users conduct high quality resource retrieval remains a challenge and should be considered first for an e-learning system. The traditional indexing approach which classifies multimedia resources into predefined categories can hardly meet user demands. Collaborative tagging (also known as folksonomy) systems provide users with a simple but powerful mechanism to obtain required multimedia resources. Moreover, feedbacks made by users can be utilized to refine the retrieval results during the interactive process between users and systems. In this paper, we present an interactive tutoring system with personalized multimedia resource search, based on a collaborative tagging mechanism. We also describe several application scenarios in which content-based image retrieval (CBIR) is combined with collaborative tagging to facilitate interactive tutoring in our cooking recipe database system.

Categories and Subject Descriptors
H.3.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Information Search and Retrieval, H.5.1 [**INFORMATION INTERFACES AND PRESENTATION**]: Multimedia Information Systems.

General Terms
Algorithms, Management, Human Factors.

Keywords
Collaborative Tagging, Personalized Search, Interactive Search, Content-based Image Retrieval.

1. INTRODUCTION
With the explosive growth of various resources on the Internet, multimedia search becomes more popular and important. This trend also happens in the e-learning domain. The text resources alone can hardly meet user demands. On the other hand, there are vast amount of multimedia learning resources on the Web. Yet how to help users conduct high quality resource retrieval remains a challenge and should be considered as the first task for an e-learning system. Recently, there are more and more web sites like Del.icio.us [6] and Last.fm [10] using the collaborative tagging (also known as folksonomy [8]) mechanism, where users can tag bookmarks in what they are interested. These tags not only can provide semantics-based descriptions to resources, but also may reflect users' interests. In particular, the tags given by users to a resource can be utilized to describe that resource. Similarly, the resources tagged by a user and tags given by the user can provide rich information to infer the user's interest and/or preference.

During the interactive process between users and systems, user feedbacks can provide some quite useful information to the system, and we can infer a user's needs based on the selection(s) made by him/her. For example, when a user chooses a resource from some preliminary result, we can infer that this resource chosen by him/her is more in line with his/her needs. Thus the system can re-rank the searching result by arranging the resources which are more similar with the chosen one in a higher priority. By repeating this process, the searching result becomes more and more accurate.

In this paper, we present an interactive tutoring system with personalized multimedia resource search. We narrow down our focus on to a particular e-learning system named cooking recipe tutoring system. As the demand of people to healthy food is increasing, more and more people are interested in cooking. In general, recipes can be associated with many multimedia resources such as texts, images and videos. Our recipe learning system uses the collaborative tagging mechanism. Based on the tags, user profiles and resource profiles can be captured and utilized to support personalized search. We also propose a method to refine the searching result by utilizing user feedbacks. As users may want to search a recipe by an example image (e.g. photo), we present a method which combines the content-based image retrieval with collaborative tagging.

The structure of the remaining paper is as follows: Section 2 reviews some related works. Our interactive tutoring system with personalized multimedia resource search is presented in section 3. In section 4, we present several scenarios and explain how interactive and personalized recipe searches are conducted in these scenarios. Finally, we conclude the paper and introduce future work in section 5.

MTDL'11, December 1, 2011, Scottsdale, Arizona, USA.

2. RELATED WORKS

2.1 Collaborative Tagging

Collaborative tagging, also known as folksonomy, has become popular on the Web. Golder and Huberman [8] analyzed the collaborative tagging systems and discovered several regularities in user activity, tag frequencies and so on. Bischoff et al. [2] presented a study of tagging behavior for different kinds of resources and systems. In [7], Diederich and Iofciu proposed a method to create user profiles in collaborative tagging systems. In [9], Kim et al. tried to model the user profile to recommend content relevant based on user interests. Bao et al. [1] utilized social annotations to benefit web search in similarity ranking and static ranking. In [4], Chirita et al. proposed a method called P-TAG which automatically generates personalized tags for Web pages. Vallet et al. [17] proposed a method to represent a user profile and exploit the users' social tags to re-rank searching results. In our earlier study [3], a new user profile and resource profile model along with a similarity measurement are presented. In this study, we present a new similarity measurement between two resources to meet the user needs more comprehensive.

2.2 Interactive Retrieval

Relevance feedback has been used to refine retrieval results in information retrieval and multimedia communities. Its basic idea is to utilize user interactions to infer user needs. Document-based relevance feedback is one of explicit feedback mechanisms, which is commonly used. In this mechanism, users are asked to rate the relevance of presented documents. Rocchio [13] proposed a method to extend the original query based on both the relevant documents and the non-relevant documents. Another feedback mechanism is based on implicit feedbacks (e.g., clicks, reformulations and dwell time), which is more convenient for users and much faster than editorial judgments. Claypool et al. [5] studied how implicit behaviors are related to the user interests. In this study, we presented an implicit feedback mechanism based on user clicks.

2.3 Content-based Image Retrieval

The goal of content-based image retrieval (CBIR) is to support image retrieval based on content properties such as color, shape and texture. Image descriptors are used to extract feature vectors and to measure the similarity between two images in CBIR. In [16], Torres and Falcao concluded that there are three common descriptors, i.e. color descriptor, shape descriptor and texture descriptor. Color Histogram [15] is the most commonly used color descriptor. The similarity between two color histograms can be calculated by their intersections. Stricker and Orengo [14] also proposed a color descriptor which uses the mean, variance and skewness to form the feature vector. In our recipe tutoring system, we use the color descriptor to identify which cooking method a recipe uses.

3. INTERACTIVE TUTORING ON COOKING: ARCHITECTURE AND MECHANISMS

In this section, we present our approach of supporting interactive tutoring on cooking activities, by means of introducing the main mechanism and components. Figure 1 shows the main architecture of our system.

3.1 User Profile and Resource Profile

To model user profile and resource profile, we use the vector space which has been commonly used in information retrieval. An intuitive thinking is to use the frequency of a tag to represent the value of this tag. But in the resource profile, using the frequency to describe the resource may lead to the so called cold-start problem which means that new resources have few tags and the frequency of their tags is also limited. Considering this problem, we use the normalized approach to model the resource profile. Another method we use to improve the cold-start performance is to extract key words from the descriptions (e.g., ingredients and directions) of a recipe as secondary tags of the recipe.

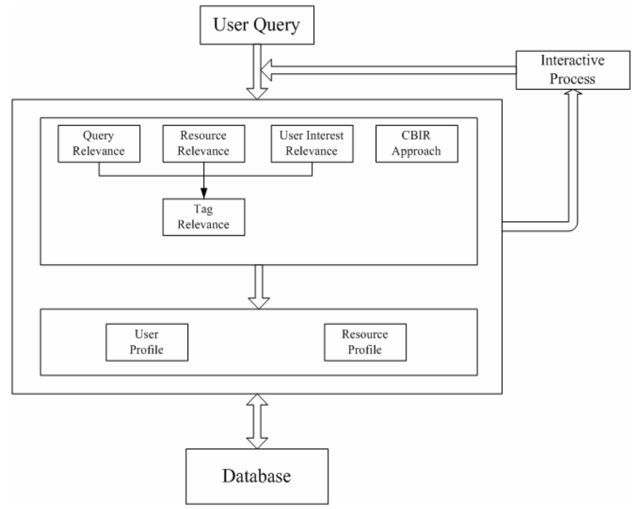

Figure 1. Architecture of Our System

We now define the notion of user profile as follows:

Definition 1. The profile of user *i*, denoted by $\vec{U}_i$, is a vector of tag: value pairs:

$$\vec{U}_i = (t_{i,1} : v_{i,1}, t_{i,2} : v_{i,2}, \ldots, t_{i,n} : v_{i,n}),$$

where $t_{i,j}$ denotes a tag used by user i, $v_{i,j}$ is the number of times user i uses the tag j to annotate resources, which indicates the preference degree of user i on tag $t_{i,j}$, n is the total number of tags that user i has used.

Definition 2. The profile of resource c, denoted by $\vec{R}_c$, is a vector of tag: value pairs:

$$\vec{R}_c = (t_{c,1} : w_{c,1}, t_{c,2} : w_{c,2}, \ldots, t_{c,n} : w_{c,n}),$$

where $t_{i,j}$ denotes a tag posted to describe resource c, n is the total number of tags posted to describe resource c, $w_{c,j}$ is the weight on how resource c possesses the tag $t_{c,j}$, which can be obtained as follows:

$$w_{c,j} = \frac{M_{c,j}}{M_c},$$

where $M_{c,j}$ is the number of times that tag $t_{c,j}$ has been used to annotate resource c, and M_c is the total number of users who have ever annotated resource c. Note that $w_{i,j}$ is normalized by M_c. By using the normalized method, the new resources that have less tags can have a fair competition with old resources, which is more reasonable.

Another method to cope with the problem of cold-start is to extract key words from the descriptions of resource c, as its secondary tags, denoted by $t^e_{c,j}$, and the value $t^e_{c,j}$ can be obtained as follows:

$$w^e_{c,j} = \alpha \times Min(w_{c,1}, w_{c,2}, \ldots, w_{c,n}),$$

where $w_{c,j}$ is the tag that resource c already has, and α is a parameter in the range of [0, 1] which is to adjust the weight of secondary tags.

3.2 Relevance Measurement

In our recipe learning system, three types of relevance measurement have been identified for us to handle, i.e. query relevance, resource relevance and user interest relevance. As all these measurements are actually based on the tag relevance measurement, we shall present the tag relevance measurement first, and then explain how to use the tag relevance to measure the other types of relevance.

3.2.1 Tag Relevance Measurement

The tag relevance means how a tag is relevant to a resource. Let $\vec{R}_c = (t_{c,1} : w_{c,1}, t_{c,2} : w_{c,2}, \ldots, t_{c,n} : w_{c,n})$ denote the vector of resource c. The tag relevance problem can be regarded as to decide what the probability is for a tag, denoted by t, to be able to describe a resource c. According to the Bayes' rule, the probability is:

$$P(t \mid c) = \frac{P(c \mid t) * P(t)}{P(c)}.$$

Further, $P(t)$ and $P(c)$ can be set to a constant (e.g. 1). Assuming that the presence or absence of a tag in resource c is independent of the presence or absence of any other tag, so we have:

$$P(t \mid c) \sim P(c \mid t) = \prod_{t_i \in c} P(t_i \mid t).$$

If we use this formulation directly, however, a problem occurs: if the tag t_i and tag t have no relevance, $P(t_i \mid t)$ will be zero, which leads to that the multiplication also becomes zero. Obviously, this is unreasonable. So we describe the relevance equally by the logarithm of this term. Since log is a monotonic function, we have the following

$$P(t \mid c) \sim \sum_{t_i \in c} \log P(t_i \mid t).$$

But the problem remains unsolved. Considering $P(t_i \mid t)$ to be less than 1, this makes $\log P(t_i \mid t)$ to be negative. Further, $\log P(t_i \mid t)$ will be negative infinity when t_i and t have no relevance. So we transform the formula to following:

$$P(t \mid c) \sim \sum_{t_i \in c} \log P(t_i \mid t) \sim \sum_{t_i \in c} \frac{1}{|\log P(t_i \mid t)|}.$$

Now the problem mentioned above can be solved. If the tag t_i and tag t have no relevance, $\frac{1}{|\log P(t_i \mid t)|}$ tends to be zero just like before, but the formula is now an adding-up sum rather than a multiplication, which is what we expect.

We can approximate $P(t_i \mid t)$ by utilizing the number of times that tag t_i and tag t appear in the profile vector of the same recipe. Further, $P(t_i \mid t)$ may fall into three classes, i.e. ,

1. t_i and t are identified,
2. t_i and t have appeared at least once in the profile vector of the same recipe,
3. t_i and t never appear in the profile vector of the same recipe.

For the first case, we can directly use the $w_{c,j}$ of the resource c , i.e. ,

$$P(t_i \mid t) \sim w_{c,j}, t_i = t = t_{c,j}.$$

For the second case, we approximate it by the following:

$$P(t_i \mid t) \sim \frac{\sum_{i=1}^{m} \frac{Min(M_{i,x}, M_{i,y})}{Max(M_{i,x}, M_{i,y})}}{m},$$

$$t_{i,x} = t_i, t_{i,y} = t \quad or \quad t_{i,x} = t, t_{i,y} = t_i,$$

where $M_{i,x}$ is the number of users who use tag $t_{i,x}$ to annotate resource i, $M_{i,y}$ is the number of users who use tag $t_{i,y}$ to annotate resource i, and m is the total number of recipes which contain both tag $t_{i,x}$ and $t_{i,y}$. Here we use the maximum one between $M_{i,x}$ and $M_{i,y}$ to divide the minimum one based on a simple observation: The greater the gap between $M_{i,x}$ and $M_{i,y}$ the less relevance between $t_{i,x}$ and $t_{i,y}$.

For the 3rd case, $\frac{1}{|\log P(t_i \mid t)|}$ tends to be zero. As mentioned before, it is now reasonable to set it zero, since the formula is an adding-up sum.

Now we are able to define the query relevance measurement, resource relevance measurement and user interest relevance measurement easily, since all the three measurements are based on the tag relevance measurement, as discussed below.

3.2.2 Query Relevance Measurement

We model a query given by a user i, denoted by $\vec{q}_i$, as a vector of terms, i.e.:

$$\vec{q}_i = (t^q_{i,1}, t^q_{i,2}, \ldots, t^q_{i,m}),$$

where $t^q_{i,j}$ denotes a term in the query, and m is the total number of terms in the query. For example, a user may type in a query like 'spicy beef' to search for dishes, then $\vec{q}_i$ is of the form (spicy, beef).

As query $\vec{q}_i$ is typically composed of several tags, an intuitive idea of the query relevance measurement for us is to add up all the values of relevance between each tag of $\vec{q}_i$ and the tags of resource's profile vector.

Specifically we can measure the query relevance as follows:

$$r(q, c) = \sum_{t^q_j \in q} \sum_{t_i \in c} \frac{1}{|\log P(t_i \mid t^q_j, L)|},$$

where q denotes the query given by a user, c denotes a resource, t_i denotes a tag in resource c, and t_j^q denotes a term in the query q.

The larger value of $r(q,c)$ is, the higher relevance between query q and resource c.

3.2.3 Resource Relevance Measurement

In the case of resource relevance measurement, one of the resources can be regarded as the query resource, and the other as the target resource. Overall it is similar to the query relevance measurement. However, there is one difference. In the case of query relevance measurement, the weight of tags in the query is all same, i.e., we set 1 to each tag. But for resource relevance measurement, the weight of tags in the query resource should be handled differently. In particular, we use value of tag in the profile vector of the query resource as the weight of that tag.

We can now measure the resource relevance between two resources c_1 and c_2, as follows:

$$r(c_1,c_2)=\sum_{t_{1,i}\in c_1}\sum_{t_{2,j}\in c_2}\frac{w_{1,i}}{\left|\log P(t_{2,j}\mid t_{1,i},L)\right|},$$

where c_1 and c_2 denote two resources, $t_{1,i}$ denotes a tag of resource c_1, $t_{2,j}$ denotes a tag of resource c_2 and $w_{1,i}$ is the value of tag $t_{1,i}$ in the profile vector of resource c_1.

The larger value of $r(c_1,c_2)$ is, the higher relevance between the two resources c_1 and c_2.

3.2.4 User Interest Relevance Measurement

As we have modeled the user profile as a vector of tag: value pairs, we can measure the user interest relevance in the same way as the resource relevance. In the case of user interest relevance measurement, the weight of tags in the user profile also should be handled differently. Like the resource relevance measurement, we can multiply the formula by the weight of the tags in the user profile.

In particular, we can measure user interest relevance between a user u, and a resource c as follows:

$$r(u,c)=\sum_{t_j\in u}\sum_{t_i\in c}\frac{v_j}{\left|\log P(t_i\mid t_j,L)\right|},$$

where u denotes a user, c denotes a resource, t_j denotes a tag of user u, t_i denotes a tag of resource c and v_j is the value of tag t_j in the profile vector of user u.

The larger value of $r(u,c)$ is, the higher relevance between user u and resource c.

3.3 Personalized Ranking

Personalized ranking in our system means that the result not only matches the query relevance but also meets user interests. Thus we combine the query relevance and the user interest relevance to conduct personalized ranking, i.e.:

$$r(q,u,c)=r(q,c)+\alpha\times r(u,c),$$

where u denotes a user, q denotes a query given by user u, c denotes a resource and α is a parameter in the range of [0, 1], which is used to adjust the weight of user interest relevance.

The larger value of $r(q,u,c)$ is, the higher position the resource c is in the searching result list of user u's query q.

3.4 Interactive Process

When the preliminary result is presented to a user, the user will choose a recipe which can better meet the user's need than other candidate recipes. We regard the recipes which are similar to the chosen one to have a high probability to meet the user's need. So we can re-rank the result by arranging the similar recipes in priority as follows:

$$r'(q,u,c)=\beta\times r(q,u,c)+(1-\beta)\times r(c,c_1),$$

where $r(q,u,c)$ is the preliminary result, u denotes a user, q denotes a query given by user u, c denotes a resource, c_1 denotes the resource chosen by the user, and β is a parameter in the range of [0, 1], which is used to adjust the weight of interactive process.

If the user continuously chooses a recipe in the new result list, we can repeat the process as follows:

$$r_{n+1}(q,u,c)=\beta\times r_n(q,u,c)+(1-\beta)\times r(c,c_n).$$

where c_n denotes the resource chosen by the user in the n-th round of retrieval (n>0), and β is a parameter in the range of [0, 1], which is used to adjust the weight of interactive process.

3.5 Content-based Approach

As users may want to search a recipe by an image example, we present a mechanism using the content-based approach to get a preliminary result, and then we can use the interactive process to refine the searching result.

According to [18], there are about 27 kinds of cooking methods in Chinese recipes. Our goal of the content-based approach is to identify what kind of cooking method the query recipe uses. We employ the color property as the image descriptor because the most significant difference among different kinds of cooking methods is color. Here we assume the user can identify the main ingredient through the picture. We use the method of Color Histogram [15] as color descriptor and compute Histogram Intersection to measure the similarity between two images. In [15], the color histogram is obtained by discretizing the image colors and counting the number of times each discrete color occurs in the image array.

The similarity measurement, based on the color histogram, called Histogram Intersection [15] is defined as follows:

$$HI(I,M)=\sum_{i=1}^{n}Min(I_i,M_i),$$

where I and M is a pair of histograms and n is the number of bins I and M contain.

We apply color histogram to identify the genre of cooking method of the query recipe, as follows:

$$\arg\max_{1\le i\le 27}(AVG_i(\sum_j HI(q,c_{i,j}))),$$

where HI is short for Histogram Intersection, q denotes the query recipe, and $c_{i,j}$ denotes a recipe resource which uses the i-th kind of cooking method.

After this process, we get a recipe set which is constrained by the cooking method and the main ingredient. Furthermore, we can rank these recipes according to the user interest relevance. Based on the preliminary result, users may have an interactive process with the system to refine the searching result.

4. INTERACTIVE TUTORING BASED ON SEARCH

In this section, we present several scenarios which a user may meet in practice, and explain how to use our method to conduct interactive tutoring for these scenarios.

4.1 Feature-based Interactive Search

It is not easy to type in the exact name of a recipe when a user needs to search for a recipe. In more cases, a user conducts a recipe search with some constraints such as the main ingredient, cooking method and flavor. For example, a user wishes to eat the beef with braised cooking method. Therefore he may type in 'beef' and 'braised' as a query to search a recipe. Meanwhile, we find that the user likes spicy food from his profile. As a result, the system returns the recipes which have high relevance with 'beef', 'braise' and 'spicy'. After that, the user may have an interactive process with the system as discussed earlier.

4.2 Content-based Interactive Search

Sometimes users may want to search a recipe by an image example. For example, a user may get a recipe picture from a website, and that picture looks very attractive to him. As a result, the user wishes to learn cooking that recipe, and thus he needs to search that recipe first. However, the user can not find the name of the recipe, but can identify the main ingredient through the picture. For such a scenario, the user can conduct a search by uploading the picture and typing in the main ingredient of the recipe to our system, and the system will return a preliminary result by using the CBIR approach first (cf. section 3), with a constraint on the cooking method and/or the main ingredient. Furthermore, the preliminary result is ranked according to the user interest relevance. After that, the user may proceed to an interactive process with the system to get a more accurate result. This scenario is shown in Figure 2.

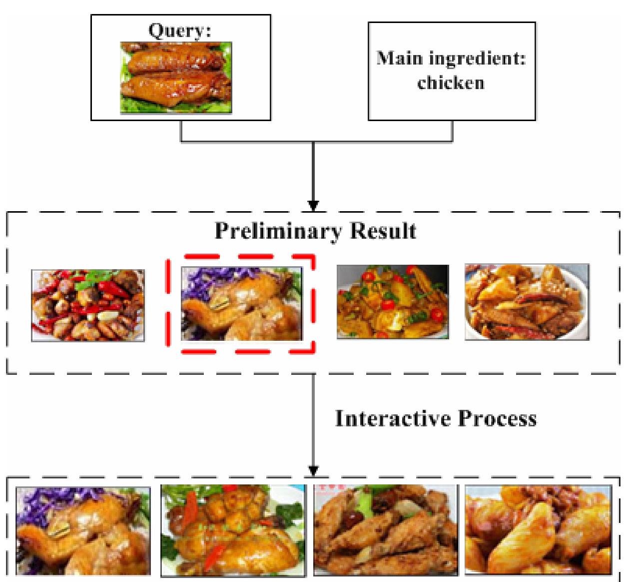

Figure 2. Example of Content-based Interactive Search

4.3 Interactive Search for Recipe Adaptation

Under certain circumstances, users may have the need to make an adaptation to a recipe. For example, a user (Tom) finds an attractive recipe 'curry chicken' from the database. However, he does not have the chicken but only beef available to him, so he may need to search a recipe which uses the similar cooking method with 'curry chicken' but replaces the chicken with beef. As another example, user Carol is interested in a dish named 'braised potato with sauce', but she likes spicy food, thus she may search a recipe which is similar to "braised potato with sauce" but also has a spicy flavor. These two examples show the necessity of accommodating ingredient adaptation and flavor adaptation to a recipe. Here, we summarize nine aspects that users may need to make an adaptation to a recipe, including ingredient(s), flavor, cooking method, cooking difficulty level, time cost, Chinese cuisine genres[1], nutrition, dietotherapy and suitable crowd. In practice, a user may wish to make more than one adaptation in the nine aspects to meet his needs better, and he can type in the adaptation keywords as well as the recipe he wants to adapt, by conducting an adaptive search.

To conduct an interactive search for recipe adaptation, we present a method as follows:

$$r(c, c_x, T_{adaptation}) = \alpha \times r(T_{adaptation}, c) + (1-\alpha) \times r(c, c_x),$$

where c denotes a resource, c_x denotes a resource that the user intends to make a adaptation, $T_{adaptation}$ denotes the set of adaptation tags, and α is a parameter in the range of [0, 1] which is used to adjust the weight of adaptation tags.

The larger value of $r(c, c_x, T_{adaptation})$ is, the higher position the resource c is in the adaptation searching list.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an interactive tutoring system with personalized multimedia resource search. The core method in our system is the tag relevance measurement, as all other methods are based on it. We have also presented an interactive process mechanism to refine the searching result. Several scenarios are presented to explain how to use our tutoring system in daily activities of learning how to cook.

There are some potential extensions to our current work. In particular, we would like to explore making a balance between the resource relevance and the user interest relevance in meeting the needs of users more effectively and efficiently. Besides, we will try to design a more comprehensive ranking algorithm for recipe adaptation. Furthermore, as the content-based image retrieval is important in the recipe domain and the color descriptor alone is not enough, we will try to design a more comprehensive content-based search mechanism by integrating other descriptors.

ACKNOWLEDGEMENT

The research described in the paper has been partially supported by a multi-server project [11][12] and a grant from the Research Grants Council of Hong Kong SAR, China [Project No. CityU 117608].

6. REFERENCES

[1] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei and Z. Su. Optimizing web search using social annotations. In *WWW' 07,* pages 501-510. ACM. 2007.

[1] Chinese cuisines have eight major genres, such as Sichuan cuisine, Cantonese cuisine, Shanghainese cuisine, and so on.

[2] K. Bischoff, C. S. Firan, W. Nejdl, and R. Paiu. Can all tags be used for search? In *CIKM'08,* pages 139-202. ACM, 2008.

[3] Y. Cai and Q. Li. Personalized Search by Tag-based User Profile and Resource Profile in Collaborative Tagging Systems. In *CIKM' 10*, pages 263-274.

[4] P. Chirita, S. Costache, S. Handschuh, and W. Nejdl. P-tag: Large scale automatic generation of personalized annotation tags for the web. In *Proc. Of the 16th Intl. World Web Web Conf,* pages 8-12, 2007.

[5] M. Claypool, D. Brown, P. Le, M. Waseda. Inferring User Interest. *IEEE Internet Computing*, vol. 5, no. 6, pp. 32-39, Nov./Dec. 2001, doi:10.1109/4236.968829.

[6] Del.icio.us. Available at: http://delicious.com

[7] J. Diederich and T. Iofciu. Finding communities of practice from user profiles based on folksonomies. In *Proceedings of TEL-CoPs'06*, 2006.

[8] S. A. Golder and B. A. Huberman. Usage patterns of collaborative tagging systems. *J. Inf. Sci.,* 32(2): 198-208, 2006.

[9] H. N. Kim, I. Ha, J. G. Jung and G. K. Jo. User preference modeling from positive contents for personalized recommendation. *Corruble, V., Takeda, M., Suzuki, E. (eds.) Ds 2007. LNCS (LANI),* 4755:116C126, 2007.

[10] Last.fm. Available at: http://www.last.fm

[11] B. Ng, R. Lau, A. Si, and F. Li. Multiserver support for large scale distributed virtual environments. *IEEE Trans. on Multimedia*, 7(6):1054-1064, 2005.

[12] B. Ng, F. Li, R. Lau, A. Si, and A. Siu. A performance study of multi-server systems for distributed virtual environments. *Information Sciences*, 154(1):85-93, 2003.

[13] J. J. Rocchio. Relevance feedback in information retrieval. 1971.

[14] M. A. Stricker and M. Orengo. Similarity of Color Images. In *Storage and Retrieval for Image and Video Databases (SPIE),* pages 381-392, 1995.

[15] M. Swain and D. Ballard. Color Indexing. *International Journal of Computer Vision,* 7(1): 11-31, 1911.

[16] R. S. Torres and A. X. Falcão. Content-based image retrieval: theory and applications. *Revista de Informática Teórica e Aplicada* 13 (2006), no. 2, 161–185.

[17] D. Vallet, I. Cantador, and J. M. Jose. Personalizing web search with folksonomy-based user and document profiles. In *Advances in Information Retrieval, 32nd European Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28-31, 2010. Proceedings,* pages 420-431, 2010.

[18] Y. Wang and M. G. Lv. Cooking of Chinese Recipes. *Dongbei University of Finance &Economics Press Co., Ltd*, 2003.