Available online at www.sciencedirect.com
ScienceDirect
The Journal of Systems and Software 81 (2008) 1377–1388
The Journal of Systems and Software
www.elsevier.com/locate/jss

Document recommendation for knowledge sharing in personal folder environments

Duen-Ren Liu *, Chin-Hui Lai, Chiu-Wen Huang

Institute of Information Management, National Chiao Tung University, Hsinchu 300, Taiwan

Received 17 October 2006; received in revised form 12 October 2007; accepted 27 October 2007
Available online 17 November 2007

Abstract

Sharing sustainable and valuable knowledge among knowledge workers is a fundamental aspect of knowledge management. In organizations, knowledge workers usually have personal folders in which they organize and store needed codified knowledge (textual documents) in categories. In such personal folder environments, providing knowledge workers with needed knowledge from other workers' folders is important because it increases the workers' productivity and the possibility of reusing and sharing knowledge. Conventional recommendation methods can be used to recommend relevant documents to workers; however, those methods recommend knowledge items without considering whether the items are assigned to the appropriate category in the target user's personal folders. In this paper, we propose novel document recommendation methods, including content-based filtering and categorization, collaborative filtering and categorization, and hybrid methods, which integrate text categorization techniques, to recommend documents to target worker's personalized categories. Our experiment results show that the hybrid methods outperform the pure content-based and the collaborative filtering and categorization methods. The proposed methods not only proactively notify knowledge workers about relevant documents held by their peers, but also facilitate push-mode knowledge sharing.

Keywords: Document recommendation; Knowledge management; Personal folder; Knowledge sharing; Text classification

1. Introduction

The rapid emergence of Knowledge Management in recent years has played a key role in helping organizations gain and maintain a competitive advantage. Sharing sustainable and valuable knowledge among knowledge workers is a fundamental aspect of knowledge management. Organizational knowledge and expertise are usually codified into textual documents, including forms, letters, papers, manuals and reports, to facilitate knowledge capture, searching, and sharing (Nonaka, 1994).

Knowledge workers tend to keep their codified knowledge in personal folders. Textual documents stored in each worker's personal folder are usually organized into categories. In such personal folder environments, providing knowledge workers with needed knowledge from other workers' folders is important to facilitate knowledge sharing. Although conventional knowledge management systems (KMS) provide a search function to help workers find needed knowledge, very few KMS address the issue of proactively providing workers with needed knowledge in personal folder environments. Recommender systems can be adopted to provide an effective means of addressing this shortcoming of KMS.

Conventional application domains of recommender systems cover areas such as "Music", "Movie" and "Product" recommendations. Various recommendation methods have been proposed for such systems (Breese et al., 1998; Burke, 2002; Li and Kim, 2003; Liu and Shih, 2005). For example, Content-based Filtering (CBF) utilizes users' profiles to determine recommendations for target users. In applications that recommend documents, CBF provides

* Corresponding author. Fax: +886 3 5723792.
E-mail address: dliu@iim.nctu.edu.tw (D.-R. Liu).

0164-1212/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2007.10.027

recommendations by matching user profiles (e.g., interests) with content features (e.g., feature vectors of documents). Each user profile is derived by analyzing the content features of documents accessed by the user. Collaborative Filtering (CF), which assumes that items from similar (like-minded) users are often relevant, utilizes preference ratings given by the users to determine recommendations made to a target user. Hybrid recommender systems integrate content-based and collaborative filtering to enhance the quality of recommendations.

The LIBRA system (Mooney and Roy, 2000) is an example of a content-based filtering system that recommends books based on information extracted from Web pages. Meanwhile, Siteseer (Rucker and Polanco, 1997) uses collaborative filtering to provide Web page recommendations based on the folders of bookmarks. However, neither method considers recommending Web pages to appropriate categories. Knowledge Pump (Glance et al., 1998) classifies documents into a commonly agreed classification scheme based on the content of documents. However, the classification is a commonly agreed classification scheme, rather than a personalized one. RAAP (Delgado et al., 1998) is an example of a hybrid system developed to recommend a user's newly classified bookmark (document) to other users with similar interests. A common category schema, rather than a personalized one, is predefined for all users to support classification.

Conventional document recommender systems generally assume a common category schema without considering personalized categories. Since both the source and the target user have the same category schema, such recommender systems are simplified to recommending documents to the target user without considering which category the document belongs to. Although the Siteseer system (Rucker and Polanco, 1997) considers the personalized folders of bookmarks, it simply takes one specific folder (category) of the target user at a time as the target for recommendation, and does not address the issue of recommending items to the target user's appropriate categories. In this paper, we investigate the issue of recommending textual documents to appropriate categories in personal folder environments. Each knowledge worker has a personal folder for storing documents in user-defined categories. In personal folder environments, knowledge workers can define their own categories, so the recommender system also needs to consider the appropriate category for a recommended document. Generally, text categorization techniques (Langari and Tompa, 2001; Larkey and Croft, 1996) can be used to allocate documents to appropriate categories. We propose novel recommendation methods that incorporate text categorization techniques to recommend documents to the appropriate categories of a target worker's personal folders. Several novel methods have been proposed for this purpose, including content-based filtering and categorization, collaborative filtering and categorization, and hybrid methods. The proposed methods can proactively provide knowledge workers with needed textual documents from other workers folders. Experiments are conducted to evaluate the performance of various methods using data collected from a research institute laboratory. The experiment results show that the hybrid methods outperform the other methods.

The remainder of the paper is organized as follows. Section 2 reviews the background of this study, including knowledge management, information retrieval, text categorization, and recommender systems. Our proposed method is described in Section 3. Section 4 evaluates the performance of our methods. Finally, Section 5 presents our conclusions and indicates the direction of our future work.

2. Background and related work

In this section, we describe the basic concepts of our research, including knowledge management, information filtering and retrieval, text categorization, and recommender systems.

2.1. Knowledge management

Knowledge management is a systematic process of gathering, organizing, sharing, and analyzing knowledge in terms of resources, documents, and people skills within and across an organization (Davenport and Prusak, 1998; Nonaka, 1994). Textual data, such as articles, reports, manuals, and know-how documents are treated as valuable and explicit knowledge; thus, effective document management is especially important (Nonaka, 1994). Generally, existing knowledge management systems adopt codified approaches (Zack, 1999) or social network dialog (Agostini et al., 2003) to facilitate knowledge-sharing and support.

2.2. Information retrieval and information filtering

Information retrieval (IR) deals with the representation, organization, storage, and access to information items (Baeza-Yates and Ribeiro-Neto, 1999). Essentially, IR focuses on searching for and indexing a large number of documents and then presenting users with data that meets their information needs. One popular IR method uses a vector model, which assigns non-binary weights to index the most discriminating terms in documents based on the tf–idf approach (Salton and Buckley, 1988; Baeza-Yates and Ribeiro-Neto, 1999), where terms with a higher frequency in one document and a lower frequency in other documents are better discriminators for representing the terms of the document. In the tf–idf approach, tf denotes the occurrence frequency of a particular term in a document, while idf denotes the inverse document frequency of a particular term measured by $\log_2 (N/n + 1)$, where N is the number of documents in the collection, and n is the number of documents in which term i occurs at least once. The weight of a term, i, in a document, j, is expressed as follows:

$$w_{i,j} = \mathrm{tf}_{i,j} \times \mathrm{idf}_i = \mathrm{tf}_{i,j} \times \left(\log_2 \frac{N}{n} + 1\right), \quad (1)$$

where $\mathrm{tf}_{i,j}$ is the frequency of term i in document j, and idf_i is the inverse document frequency of term i.

Information filtering helps maintain users' personal files by separating relevant and irrelevant documents based on their individual profiles. In this way, only useful information is sent to the user (Baeza-Yates and Ribeiro-Neto, 1999; Chen and Kuo, 2000; Shapira et al., 1999).

2.3. Text categorization

Text categorization or text classification assigns category or class labels to new documents automatically (Langari and Tompa, 2001; Lewis and Ringuette, 1994; Larkey and Croft, 1996). Two kinds of text categorization, namely k-NN and category vector methods are widely used (Langari and Tompa, 2001). The k-nearest neighbor method (k-NN) tries to find the top-k documents that are most similar to the target (unlabeled) document, and then assigns the target document to the category that has the majority of k-nearest neighbors. Each document can be represented as a term vector in the multi-dimensional vector space, where the weight of a term in a document is usually generated by the tf–idf approach, introduced in Section 2.2. For each unlabeled term vector, we use the cosine similarity measure to find the k nearest training term vectors. The cosine similarity measure is normally used to measure the degree of similarity between two items, x and y, by computing the cosine value of the angle between their respective feature vectors, Q and R, as shown in Eq. (2). The degree of similarity is higher if the cosine value is close to 1.

$$\mathrm{sim}(Q,R) = \mathrm{cosine}(Q,R) = \frac{Q \cdot R}{|Q||R|}. \quad (2)$$

The *category vector* method, on the other hand, derives the term vector of each category by using the tf–idf or *centroid* approach based on labeled documents. The tf–idf approach uses a similar process to that described in Section 2.2 to derive the term vector of each category (Langari and Tompa, 2001). The centroid approach derives the term vector of a category c_r by averaging the term vectors of the documents in that category, as shown in Eq. (3). Let D_{cr} denote the document set of a categoryc_r; let $w_{i,cr}$ denote the weight of a term i in c_r; and let $dw_{i,j}$ denote the weight of a term i in a document j. Then, $w_{i,cr}$ is derived as follows:

$$w_{i,cr} = \frac{1}{|D_{cr}|} \sum_{d_j \in D_{cr}} dw_{i,j}. \quad (3)$$

The similarity of a category, c_r, to an unlabeled document d_x is then calculated as $\mathrm{sim}(\vec{d}_x, \vec{c}_r)$ using the cosine measure, where $\vec{d}_x$ is a document vector and $\vec{c}_r$ is the category vector. According to the similarities between categories and unlabeled documents, we then classify the unlabeled object by assigning it the label of the most similar category, or the labels of the categories whose similarity is above a certain threshold.

2.4. Recommender systems

A recommender system helps users select items of interest from a huge stream of data. As mentioned earlier, three approaches can be used to develop recommender systems: Content-Based Filtering (CBF), Collaborative Filtering (CF), and Hybrid Filtering (Konstan et al., 1997).

Content-based recommender systems (Kamba et al., 1995; Woodruff et al., 2000) assume that if users liked certain items in the past, they will like similar items in the future. CBF systems obtain an item's characteristics (product features) and compare them with the user's profile to predict his/her preferences. Various techniques can be employed to compare and match item features with user profiles, the simplest of which is keyword matching (Claypool et al., 1999). Examples of CBF for text recommendation include the newsgroup filtering system NewsWeeder (Lang, 1995) and LIBRA (Mooney and Roy, 2000). The latter uses book information extracted from the web pages to learn a profile with weighted terms using a Bayesian text classifier. The profile is then used to predict the scores of the selected books and those with the top scores are recommended to users.

Collaborative filtering is based on the concept that if like-minded users like an item then the target user will probably like it as well (Breese et al., 1998). In other words, collaborative filtering systems consider the preferences of people who have the same or very similar interests to those of the target user. Well-known collaborative filtering systems include GroupLens (Konstan et al., 1997), Ringo (Shardanand and Maes, 1995), Siteseer (Rucker and Polanco, 1997), and Knowledge Pump (Glance et al., 1998). Many systems apply a neighborhood-based algorithm to choose a group of users based on their similarity to the target user. A weighted aggregate of the user's ratings is then used to generate predictions for the target user. The steps of the algorithm are as follows:

Step 1: Calculate the similarity between users by computing the Pearson correlation or the cosine measure of the user vectors.

Step 2: To find the neighborhood of the target user, use either the threshold approach or the k-NN (nearest neighbor) approach to select k users that are the k most similar (ranked by similarity) to the active user. In this research we use k-NN approach.

Step 3: Make a prediction based on the aggregated weights of the selected k nearest neighbors' ratings, as shown in Eq. (4):

$$P_{u,j} = \bar{r}_u + \frac{\sum_{i=1}^{k} w(u,i)(r_{i,j} - \bar{r}_i)}{\sum_{i=1}^{k} |w(u,i)|}, \quad (4)$$

where $P_{u,j}$ denotes the prediction made about item j for the target user u; $\bar{r}_u$ and $\bar{r}_i$ are the average

ratings of user u and user i, respectively; $w(u,i)$ is the similarity between target user u and user i; $r_{i,j}$ is the rating of user i for item j; and k is the number of users in the neighborhood.

Collaborative filtering assumes that documents from like-minded users are often relevant, and therefore computes the preference ratings given by various users to make a list of recommendations. Siteseer (Rucker and Polanco, 1997) provides web-page recommendations based on folders containing bookmarks (Web-page URLs), giving preference to pages held in multiple folders in the neighborhood. Recommendations are made for each of the target user's folders (categories of interests) as follows. A target user's specific category of interest (folder) is used as the basis to form a virtual community of the target user. Users in the community are virtual neighbors of the target user and are selected based on the user-folder similarity, which is measured by the degree of overlap (such as common URLs) between the neighbor's folder and the target user's specific folder. Although Siteseer considers personalized folders of URLs, it does not recommend items (URL bookmarks) to appropriate categories. Instead, it simply takes one specific folder (category) of the target user at a time to make recommendations. In general, folders may have multiple levels with hierarchical relationships that form a hierarchy of categories. Neither our approach nor Siteseer utilizes the hierarchical relationships between folders in the design of recommendation methods. Knowledge Pump (Glance et al., 1998) classifies documents into commonly agreed categories based on the content of the documents. Then, the CF technique is used to recommend documents based on the personal profiles of advisors – people whose opinions the user trusts. The classification scheme used in the recommender system is commonly agreed, rather than personalized.

Hybrid recommender systems combine content-based filtering and collaborative filtering to improve the accuracy of recommendations. Two such methods, the weighted model and the meta-level model, use different strategies to combine content-based filtering and collaborative filtering (Burke, 2002; Li and Kim, 2003). The weighted model uses linear combinations of the prediction results. For example, the method was applied to recommend news in an on-line newspaper (Claypool et al., 1999). The meta-level model employs a sequential combination of collaborative and content-based filtering, whereby the output generated by content-based filtering is used as the input for collaborative filtering (Burke, 2002). The user profile of the target user contains user preferences for each product's features (i.e., it describes the user's interests). The similarity measures of the user profiles and product profiles (features of the products/items) are then derived to predict the target user's preference ratings on unrated items. This process converts a sparse user-rating matrix into a dense user-rating matrix. Collaborative filtering then uses the dense matrix to provide recommendations. For instance, Melville et al. (2002) proposed a Content-Boosted Collaborative Filtering (CBCF) approach for movie recommendations, where pseudo user-ratings are derived by combining users' actual ratings and content-based predictions on unrated items. Then, the method applies collaborative filtering based on this dense matrix.

RAAP (Delgado et al., 1998) is an example of a hybrid system that can classify and recommend bookmarks retrieved from the Web. A bookmark (document) is classified and stored in a user's category based on the document's content and the user's profile. A common category schema, rather than a personalized one, is predefined for all users to support the classification. The system uses a hybrid approach to recommend a user's newly classified bookmark to other users with similar interests. The InLinx system (Bighini et al., 2003) also supports the classification and recommendation of bookmarks retrieved from the Web based on content analysis and virtual clusters. However, a detailed description of the approach was not provided by the authors. Middleton et al. (2004) presented an ontological user profiling approach to recommend academic papers. This scheme makes recommendations according to the correlations between the users' current profiles (topics of interest) and papers classified as belonging to those topics. Users with similar interests are identified by computing the Pearson correlation between the users' profiles. Recommended papers are those that match the user's profile and have been read by similar users.

3. Proposed recommendation methods

This section describes the proposed methods, which combine recommendation techniques with text categorization techniques to recommend documents to the appropriate categories of the target user's personal folders.

In an organization, documents, manuals and reports from people in the same project team or with similar work experience can be useful when executing a new task. One way to reuse knowledge in an enterprise is to share it by an interflow of knowledge documents. However, this can create a problem for knowledge workers because they have to spend time managing the documents they receive. As mentioned earlier, each knowledge worker may organize his/her folders to manage different types of information in different categories that form a personal folder environment, as shown in Fig. 1. Thus, to be effective a knowledge management system must be able to recommend documents stored in other knowledge workers' folders to the appropriate category of the target worker's personal folder automatically.

The proposed recommendation methods can be used to proactively notify knowledge workers about peer-reviewed documents and facilitate push-mode knowledge sharing. Two strategies can be used to share knowledge among workers: a pull strategy and a push strategy (Lei et al., 2000; Meso and Smith, 2000). The pull strategy means that workers have to find and retrieve the knowledge they need,

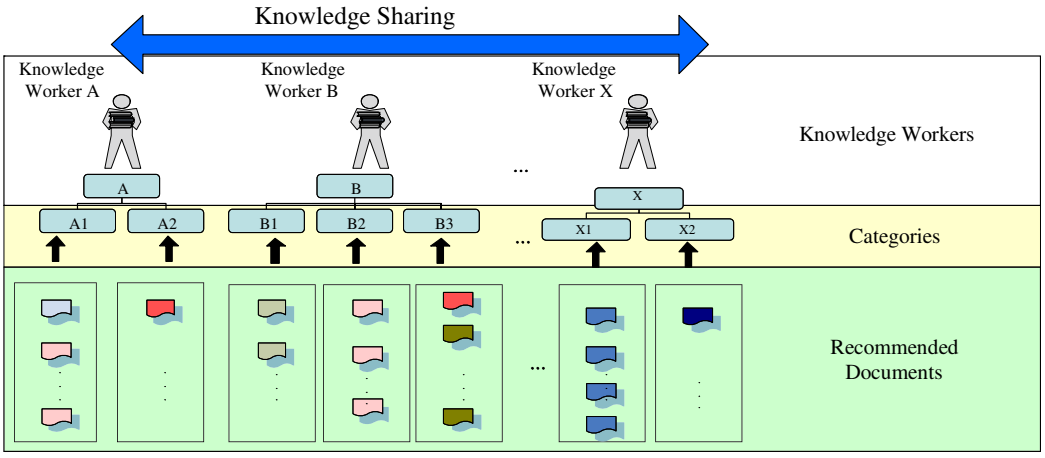

Fig. 1. Knowledge sharing in a personal folder environment.

while the push strategy means that knowledge can be delivered to people proactively by KM systems or KM techniques. Knowledge diffusion can be evolved from "Pull" to "Push" by applying our proposed recommendation methods. In this way, explicit knowledge embedded in personal folders can be circulated peer-to-peer to facilitate knowledge sharing and diffusion.

We propose three document recommendation methods for personal folder environments, namely, Content-Based Filtering and Categorization (CBFC), Collaborative Filtering and Categorization (CFC), and Hybrid Filtering and Categorization (HFC). A knowledge worker may create folders with multiple levels to form a hierarchy of categories for classifying and managing his/her documents. In general, documents are stored in the leaf nodes (categories) of the hierarchy. To simplify our research problem, in this paper, a user's folders are regarded as one level of categories. In the proposed methods, hierarchical folders are translated into one level of categories by taking each node (folder) in the hierarchy as a category. Consequently, a user's folders with/without a hierarchy are regarded as one level of categories for recommending documents. Instead of using conventional approaches for making a list of documents for recommendation, we construct a list of document–category pairs for recommendation, where a document–category pair (d_j, c_a) indicates that a document d_j is recommended to be placed in the category c_a of the target user's folder. We discuss the process in detail in the following subsections.

3.1. Content-based filtering and categorization

Content-based filtering and categorization (CBFC) locates candidates (document–category pairs) for recommendation by examining the content of profiles and predicting if they are suitable for recommendation. The method comprises three phases: generating profiles, document filtering, and generating recommendations. Profile generation prepares three profiles: a Document Profile (DP), a Category Classifier (CC), and a User Profile (UP), which are used in the document filtering phase to measure the similarity between a document and a category of the target worker. In the last phase, a list of document–category pairs is generated for recommendation to the target worker(s). We now examine the three phases of CBF in depth.

3.1.1. Phase 1: profile preparation

As shown in Fig. 2, three kinds of profiles, user profiles, category classifiers, and document profiles, are used to record information about the documents, categories, and users, respectively. A document profile is generated from a specific document, while a category classifier is derived from documents in a specific category. The user profile is evolved from all the documents of interest to the user. In the following, we explain how to generate and denote these profiles.

3.1.1.1. Document profile (DP). A document can be represented as an n-dimensional feature vector of terms and their respective weights, derived from the term frequency and the inverse document frequency (Salton and Buckley, 1988). Let d_j be a document, and let document profile $\mathrm{DP}_j = \langle dt_{1,j}{:}dw_{1,j}, dt_{2,j}{:}dw_{2,j}, \ldots, dt_{n,j}{:}dw_{n,j} \rangle$ be the feature

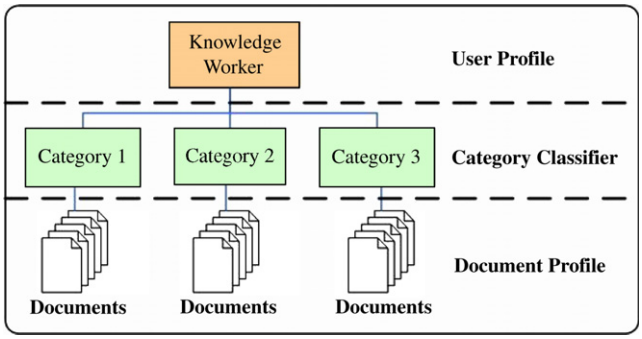

Fig. 2. Profiles in a personal folder environment.

vector of d_j, where $dw_{i,j}$ is the weight of $dt_{i,j}$ denoting a term i that occurs in d_j. Note that the weight of a term represents its degree of importance in the document. We adopt the tf–idf approach (Eq. (1)) to derive the document profile. Let the term frequency $\mathrm{dtf}_{i,j}$ be the occurrence frequency of term i in d_j, and let the document frequency df_i represent the number of documents containing term i. The importance of a term i to a document d_j is proportional to the term frequency and inversely proportional to the document frequency, expressed as:

$$dw_{i,j} = \frac{1}{\sqrt{\sum_i \left(\mathrm{dtf}_{i,j} \times \left(\log \frac{N}{df_i} + 1\right)\right)^2}} \mathrm{dtf}_{i,j} \times \left(\log \frac{N}{df_i} + 1\right), \tag{5}$$

where N is the total number of documents and the denominator is a normalization factor.

3.1.1.2. Category classifier (CC). A category classifier is constructed by adopting the tf–idf approach (Eq. (1)) to extract the discriminating terms and their weights from the categories of a worker. Let $\mathrm{CC}_r = \langle \mathrm{cct}_{1,r}{:}\mathrm{ccw}_{1,r}, \mathrm{cct}_{2,r}{:}\mathrm{ccw}_{2,r}, \ldots, \mathrm{cct}_{n,r}{:}\mathrm{ccw}_{n,r} \rangle$ be the category classifier of category c_r, where $\mathrm{ccw}_{i,r}$ is the weight of $\mathrm{cct}_{i,r}$, i.e., a term i that occurs in c_r. In addition, let the term frequency $\mathrm{ctf}_{i,r}$ be the occurrence frequency of term i in c_r, and let the category frequency cf_i represent the number of categories of a target user u that contain term i. The weight $cw_{i,r}$ of term i in a category c_r is proportional to the term frequency and inversely proportional to the category frequency, expressed as in the following equation:

$$cw_{i,r} = \frac{1}{\sqrt{\sum_i \left(\mathrm{ctf}_{i,r} \times \left(\log \frac{L_u}{cf_i} + 1\right)\right)^2}} \mathrm{ctf}_{i,r} \times \left(\log \frac{L_u}{cf_i} + 1\right), \tag{6}$$

where L_u is the total the number of categories of user u. For hierarchical folders, each node (folder) in the hierarchy is regarded as a category in our methods. All documents stored in a node c_r, and the nodes of the sub-trees that have c_r, as the root node are used to generate the category classifier of c_r,.

3.1.1.3. User profile (UP). The user profile UP_x of a user u_x is represented as a feature vector with weighted terms derived by analyzing the document set of u_x. After the documents have been pre-processed and represented in the form of term vectors, UP_x is derived by averaging the feature vectors (i.e., using the centroid approach – Eq. (3)) of documents in u_x.

3.1.2. Phase 2: document filtering

This phase computes the similarity between a category and a document. Two similarity measures, the similarity between the category classifier and the document profile and the similarity between the user profile and the document profile, are used for content-based filtering and categorization. We adopt the cosine formula (Eq. (2)) to compute the similarity measures. There may be cases where the folder does not provide enough information due to poor category construction or insufficient documents. To resolve this problem, we consider the similarity between the document profile and the user profile. The predicted rating, $\widehat{p}_{a,j}$, of the recommended document d_j (DP_j) to the category c_a (CC_a) owned by target user u_x(UP_x) is expressed as follows:

$$\widehat{p}_{a,j} = (1 - \alpha_{\mathrm{CBFC}})\mathrm{sim}(\mathrm{CC}_a, \mathrm{DP}_j) + \alpha_{\mathrm{CBFC}}\mathrm{sim}(\mathrm{UP}_x, \mathrm{DP}_j), \tag{7}$$

where $\mathrm{sim}(\mathrm{CC}_a, \mathrm{DP}_j)$ is the similarity between the category classifier CC_a and the document profile DP_j; and $\mathrm{sim}(\mathrm{UP}_x, \mathrm{DP}_j)$ is the similarity between the user profile UP_x and the document profile DP_j. Note that user u_x is the owner of category c_a. The parameter α_{CBFC} is used to determine the relative influence of the category classifier compared to the user profile. The value of α_{CBFC} ranges from 0 to 1 and is decided by the analytical experiments.

3.1.3. Phase 3: recommendation list generation

In this phase, a list of recommended document–category pairs is generated for allocation to categories in the user's personal folder. The top-N approach can be used to recommend the document–category pairs based on their predicted ratings, i.e., the pairs with the top-N rankings are selected for recommendation. Alternatively, the threshold approach can be used to recommend document–category pairs with a predicted rating higher than a given threshold. Documents that the target user has already stored are not included in the recommendation list. We use the top-N approach to generate a recommendation list in our experiments.

3.2. Two collaborative filtering and categorization approaches

Collaborative filtering and categorization makes recommendations based on the opinions of other knowledge workers whose profiles are similar to that of the target user. Two approaches have been developed for this purpose: collaborative filtering and categorization (CFC), and collaborative filtering and categorization based on the joint coefficient (CFC-J). We consider CFC first.

3.2.1. Collaborative filtering and categorization (CFC)

CFC consists of four phases, as illustrated in Fig. 3. Phase 1 generates profiles of categories and users, and Phase 2 finds peers with similar interests. The approach considers neighboring (similar) categories to locate suitable document–category pairs. Phase 3 derives the predicted ratings for document–category pairs. In the final phase, the scheme generates a list of document–category pairs for recommendation.

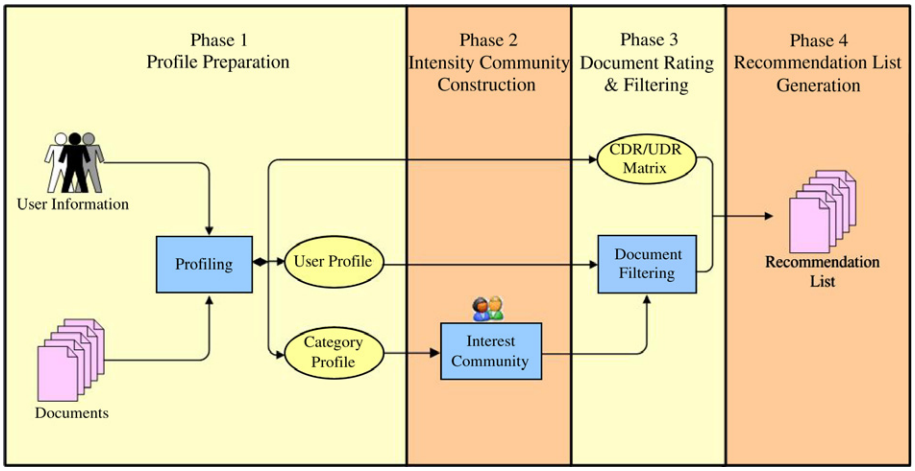

Fig. 3. The collaborative filtering process.

3.2.1.1. Phase 1: profile preparation. The purpose of this phase is to create profiles of categories and users. To generate the category classifier, CBFC uses the tf–idf approach, which considers the discriminating power of each term to distinguish between categories of a particular user. In other words, the classifier determines which category a document should be allocated to. However, it is not suitable for deriving the neighbors of categories, since the discriminating terms may distort the similarity of categories used by different workers. Therefore, a category profile is constructed to compute the similarity of categories and find their neighbors.

3.2.1.1.1. Category profile (CP). The category profile CP_a of category c_a is defined as the centroid vector obtained by averaging the feature vectors of documents in c_a. Similar to the generation of user profiles described in Section 3.1, category profiles are constructed by the centroid approach (Eq. (3)), which does not consider the effect of terms when determining the category of a user. For hierarchical folders, each node (folder) in the hierarchy is regarded as a category in our methods. All documents stored in a node c_a, and the nodes of the sub-trees that have c_a, as the root node are used to generate the category profile of c_a,.

3.2.1.2. Phase 2: identifying k-nearest neighbors. This phase finds the neighbors of the target category based on the similarity of category profiles. To recommend a document d_j to the target category c_a, the neighboring categories (neighbors) of c_a are selected from categories that contain d_j.

The cosine formula in Eq. (2) is used to decide the similarity of category profiles. There are two ways to choose neighbors: k-NN-based approaches and threshold-based approaches. The former ranks the similarity measures and chooses the k-nearest neighbors, while the latter chooses neighbors whose similarity measures are above a given threshold. We use the k-NN-based method in this work.

3.2.1.3. Phase 3: document rating and filtering. In addition to the above profiles, a *Category-Document Rating* (*CDR*) matrix and a *User-Document Rating* (*UDR*) matrix are used to record the ratings of categories and users for documents respectively. The ratings can be derived by a binary approach or a profiling approach. The binary approach derives ratings based on the criterion of whether the category/user folder contains a document. If a category c_a contains a document d_j, the rating value of c_a for d_j, $\mathrm{CDR}_{a,j}$, is 1; otherwise, it is 0. If the category c_a is used by the user u_x, i.e., u_x has document d_j, the rating value of u_x for d_j, $\mathrm{UDR}_{x,j}$, is 1; otherwise, it is 0. The profiling approach, on the other hand, uses the similarity between the category/user profile and the document profile to derive a rating. The rating value of c_a on d_j, $\mathrm{CDR}_{a,j}$, is equal to $\mathrm{sim}(\mathrm{CP}_a, \mathrm{DP}_j)$, i.e., the similarity of the category profile of c_a and the document profile of d_j. The rating value of u_x for d_j, $\mathrm{UDR}_{u,j}$, is set to $\mathrm{sim}(\mathrm{UP}_x, \mathrm{DP}_j)$, i.e., the similarity of the user profile of u_x and the document profile of d_j. The CDR/UDR generated by the binary approach is called a binary CDR/UDR, while the CDR/UDR generated by the profiling approach is called a non-binary CDR/UDR.

Eq. (8) computes the predicted rating for a document d_j recommended to a category c_a of the target user u_x:

$$\hat{p}_{a,j} = \frac{\sum_{c_b \in c_a\text{'s neighbor}} [(1-\alpha_{\mathrm{CFC}})\mathrm{sim}(\mathrm{CP}_a, \mathrm{CP}_b) \times \mathrm{CDR}_{b,j} + \alpha_{\mathrm{CFC}}\mathrm{sim}(\mathrm{UP}_x, \mathrm{UP}_y) \times \mathrm{UDR}_{y,j}]}{\text{Number of } c_a\text{'s neighbors}}, \tag{8}$$

where $\mathrm{sim}(\mathrm{UP}_x, \mathrm{UP}_y)$ is the similarity between UP_x and UP_y; $\mathrm{sim}(\mathrm{CP}_a, \mathrm{CP}_b)$ is the similarity between CP_a and CP_b; c_b belongs to c_a's neighbors; u_y is the owner of c_b; and α_{CFC} is a parameter used to adjust the relative importance of the category similarity and the user similarity.

3.2.1.4. Phase 4: recommendation list generation. This phase generates a list of document–category pairs to allocate documents to destination categories by using the top-N approach described in Phase 3 of Section 3.1.

3.2.2. Collaborative filtering and categorization based on the joint coefficient (CFC-J)

The difference between CFC and CFC-J is the way the similarity between profiles is computed. CF calculates the similarity by weighted term vectors, whereas CFC-J uses the joint coefficient, which represents the relationship between two categories/users based on the number of the documents they have in common. The more they have, the more similar they are. The joint coefficient (Jcof) in CFC-J is computed as follows:

$$\mathrm{Jcof}(c_a, c_b) = \frac{2 \times N_{a \cap b}}{N_a + N_b}, \tag{9}$$

where N_a/N_b is the number of documents in categories c_a/c_b, respectively; and $N_{a\cap b}$ represents the intersection of documents that c_a and c_b have in common. The binary CDR is used to derive N_a, N_b, and $N_{a\cap b}$.

CFC-J uses the joint coefficient instead of the profile similarity to derive the predicted rating, as expressed in Eq. (10). The joint coefficient between two users, u_x and u_y, is defined as $\mathrm{Jcof}(u_x, u_y)$:

$$\hat{p}_{a,j} = \frac{\sum_{c_b \in c_a\text{'s neighbor}} [(1-\alpha_{\text{CFC-J}})\mathrm{Jcof}(c_a, c_b) \times \mathrm{CDR}_{b,j} + \alpha_{\text{CFC-J}}\mathrm{Jcof}(u_x, u_y) \times \mathrm{UDR}_{y,j}]}{\text{Number of } c_a\text{'s neighbors}}. \tag{10}$$

3.3. Hybrid filtering and categorization

Hybrid filtering and categorization (HFC) combines content-based filtering and categorization (CBFC) and collaborative filtering and categorization (CFC) to improve the quality of recommendations. CBFC and CFC can be combined by linear or sequential combination.

3.3.1. Hybrid filtering and categorization based on linear combination (HFCL)

The hybrid filtering and categorization with linear combination method (HFCL) is a linear combination of the CBFC and CFC results. HFCL derives the predicted ratings of document–category pairs by merging the predicted ratings of CBFC and CFC described in Sections 3.1 and 3.2. The predicted rating for recommending a document d_j to a category c_a is shown in Eq. (11), where $\hat{p}_{a,j}^{\mathrm{CBFC}}$ is the predicted rating derived according to Eq. (7), and $\hat{p}_{a,j}^{\mathrm{CFC}}$ is the predicted rating derived according to Eq. (8). The parameter α_{HFCL} is used to represent the relative importance of CBFC and CFC. HFCL-J linearly combines the predicted ratings of document–category pairs from CBFC and CFC-J by Eq. (12); and $\hat{p}_{a,j}^{\text{CFC-J}}$ is the predicted rating derived according to Eq. (10):

$$\hat{p}_{a,j} = (1-\alpha_{\mathrm{HFCL}})\widehat{p}_{a,j}^{\mathrm{CBFC}} + \alpha_{\mathrm{HFCL}}\widehat{p}_{a,j}^{\mathrm{CFC}}, \tag{11}$$

$$\hat{p}_{a,j} = (1-\alpha_{\text{HFCL-J}})\widehat{p}_{a,j}^{\mathrm{CBFC}} + \alpha_{\text{HFCL-J}}\widehat{p}_{a,j}^{\text{CFC-J}}. \tag{12}$$

3.3.2. Hybrid filtering and categorization with sequential combination (HFCS)

The hybrid filtering and categorization with sequential combination method (HFCS) tries to compensate for the sparsity of rating information in collaborative filtering by using the predicted scores from the content-based mechanism as the ratings of unrated items. Thus, the rating function (CDR) in CFC is extended to eCDR derived from CBFC. An extended CDR matrix, eCDR matrix, is generated based on the predicted ratings of unrated documents derived from CBFC (Eq. (7)). For a category c_a containing a document d_j, i.e., $\mathrm{CDR}_{a,j} = 1$, $\mathrm{eCDR}_{a,j}$ is set to 1. For a category c_a that does not contain a document d_j, i.e., $\mathrm{CDR}_{a,j} = 0$, $\mathrm{eCDR}_{a,j}$ is set to 1 if the predicted rating $\hat{p}_{a,j}$ (derived from Eq. (7)) is greater than a predefined threshold; otherwise, $\mathrm{eCDR}_{a,j} = 0$. An extended UDR matrix, eUDR matrix, is generated as follows. If there exists a category c_a and u_x owns c_a such that $\mathrm{eCDR}_{a,j}$ equals 1, then $\mathrm{eUDR}_{x,j} = 1$; otherwise, $\mathrm{eUDR}_{x,j} = 0$. Moreover, the profiling approach described in Phase 3 of Section 3.2.1 can be used to derive non-binary ratings by using the similarity measures of the category/user profile and the document profile. The category/user profile of each category/user is re-generated according to the binary eCDR/eUDR matrix. The similarity measures derived based on the new category/user profile are used for the non-binary ratings.

In the HFCS method, the predicted ratings are derived as follows:

$$\hat{p}_{a,j} = \frac{\sum_{c_b \in c_a\text{'s neighbor}} [(1-\alpha_{\mathrm{HFCS}})\mathrm{sim}(c_a, c_b) \times \mathrm{eCDR}_{b,j} + \alpha_{\mathrm{HFCS}}\mathrm{sim}(u_x, u_y) \times \mathrm{eUDR}_{y,j}]}{\text{Number of } c_a\text{'s neighbors}}. \tag{13}$$

HFCS-J combines CBFC and CFC-J by the sequential approach. The joint coefficient in CFC-J is based on the number of common documents required to compute the similarity measure. HFCS-J uses extended CDR and UDR to derive the predictions, as shown in Eq. (14). Binary eCDR and eUDR are used to compute the Jcof(c_a, c_b) and Jcof(u_x, u_y), respectively. The eCDR/eUDR is generated according to the same approach described in HFCS:

$$\hat{p}_{a,j} = \frac{\sum_{c_b \in c_a\text{'s neighbor}} [(1-\alpha_{\text{HFCS-J}})\text{Jcof}(C_a, C_b) \times \text{eCDR}_{b,j} + \alpha_{\text{HFCS-J}}\text{Jcof}(U_x, U_y) \times \text{eUDR}_{y,j}]}{\text{Number of } c_a\text{'s neighbors}}. \quad (14)$$

4. Experiments and evaluations

We applied the CBFC, CFC, and hybrid methods to recommend relevant academic papers to the researchers in a research institute. In this section, we describe the experiment design, evaluation metrics, and experiment results.

4.1. Experiment setup

Since the experiments were conducted in a real application domain, namely, a research institute laboratory, there were few participants; hence, the size of the dataset was small. Knowledge workers have their own folders to store documents (research papers) that assist them in writing theses or accomplishing research projects. There are 11 users, 35 categories and 1062 documents. The sparsity in the data sets is 99.962% (749 non-zero entries in 506 × 35 matrixes). Personal folders are translated into one level of categories, as described in Section 3. The data set is divided as follows: 80% for training and 20% for testing. The training set includes documents stored in workers' personal folders, and is used to generate a recommendation list. Test data is used to verify the recommendation quality of the various methods.

Two metrics, precision and recall, are commonly used to measure the quality of recommendations. These metrics are also used extensively in information retrieval (Salton and McGill, 1983; Van Rijsbergen, 1979). Recall is the ratio of relevant documents that can be located, as shown in the following equation:

$$\text{Recall} = \frac{\text{number of correctly recommended documents}}{\text{number of relevant documents}}. \quad (15)$$

Precision is the ratio of recommended documents (predicted to be relevant) that are actually relevant to workers, as shown in the following equation:

$$\text{Precision} = \frac{\text{number of correctly recommended documents}}{\text{number of recommended documents}}. \quad (16)$$

Documents relevant to a target user u are the documents owned by u in the test set. Each relevant document is associated with its corresponding category owned by u. This is called *a relevant document–category* pair of u. Correctly recommended documents are those in the recommended document–category pairs that match the relevant document–category pairs of u.

Increasing the number of recommended documents tends to reduce the precision and increase the recall. The F1-metric is used to achieve a trade-off between precision and recall (Van Rijsbergen, 1979) by assigning equal weights to them as follows:

$$\text{F1} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}. \quad (17)$$

Each metric is computed for each researcher. Then, the average value computed for all researchers is taken as the measure of the recommendation quality.

4.1.1. Parameter selection

We conduct pilot experiments to determine the parameter values of various methods (equations). In the experiments, we systematically adjust the values of the parameters in increments of 0.1. The F1 metric (given in Eq. (17)) is chosen as the performance measure to evaluate the effectiveness of the methods. The optimal parameter values with the best results (the highest average F1 values computed over various top-N) are chosen as the parameter settings of the proposed equations.

4.2. Experiment results

We perform experiments based on the CBFC, CFC, and hybrid methods, including HFCL and HFCS. The F1 metric is used to compare the recommendation quality of the methods for various values of α and top-N recommendations. The top-N approach recommends N document–category pairs with N highest rankings of the predicted ratings.

4.2.1. Experiment one: comparison of CBFC and CBFC-CP methods

To evaluate the effectiveness of CBFC, we compare it with CBFC-CP. The CBFC approach (Eq. (7)) derives recommendations via the category classifier (CC), which uses tf–idf to distinguish between categories, whereas CBFC-CP uses the category profile (CP), which is derived by the *centroid* approach. Eq. (7) is also used to derive the CBFC-CP method by replacing CC with CP and parameter α_{CBFC} with $\alpha_{\text{CBFC-CP}}$. The parameter α_{CBFC} is used to tune the weight of predicted ratings produced by the category classifier and the user profile. We tune α_{CBFC} to between 0 and 1 by systematically adjusting the value of α_{CBFC} in increments of 0.1 and examine its effect on the F1 metrics.

The value of α_{CBFC} is determined according to the highest average F1 value computed over various top-N. The other parameters in the following experiments are decided similarly. The highest average F1 value of CBFC is achieved when $\alpha_{CBFC} = 0$, while the highest average F1 value of CBFC-CP is achieved when $\alpha_{CBFC\text{-}CP} = 0.1$. Fig. 4 shows the F1 values of CBFC and CBFC-CP for various top-N recommendations by setting α_{CBFC} of CBFC and $\alpha_{CBFC\text{-}CP}$ of CBFC-CP to 0 and 0.1, respectively. The setting $\alpha_{CBFC} = 0$ indicates that the category classifier is powerful enough to determine the correct categories for documents. The results show that, in general, CBFC outperforms CBFC-CP. The category classifier provides better quality recommendations than the category profile because it can distinguish between categories.

4.2.2. Experiment two: comparison of CFC-Binary, CFC-Profile and CFC-J methods

This experiment compares different methods of CFC: CFC-Binary, CFC-Profile, and CFC-J. CFC-Binary/CFC-Profile use binary/profiling ratings respectively, as described in Section 3.2.1, while CFC-J uses the joint coefficient approach described in Section 3.2.2. The parameter α is used to tune the weights of the ratings of the category similarity and the user similarity. Based on the highest average F1 values computed over various top-N, the α values for CFC-Binary, CFC-Profile, and CFC-J, are 0.5, 0.0, and 0.2, respectively. This indicates that the ratings for the similarity of user profiles improve the recommendation quality.

Fig. 5 compares CFC-Binary, CFC-Profile, and CFC-J under different top-N by setting $\alpha_{CFC\text{-}binary}$ to 0.5, $\alpha_{CFC\text{-}Profile}$ to 0, and $\alpha_{CFC\text{-}J}$ to 0.2. CFC-Binary outperforms the CFC-Profile, which indicates that the rating function of the latter cannot provide useful rating information. This may be due to the fact that the similarity rating between a category and a document does not reflect the user's document ratings accurately. Consequently, we adopt the CFC-Binary method rather than the CFC-Profile method to represent the CFC method in further comparisons and implementations of the hybrid approach. The results also show that CFC-J performs better when top-N is smaller, while

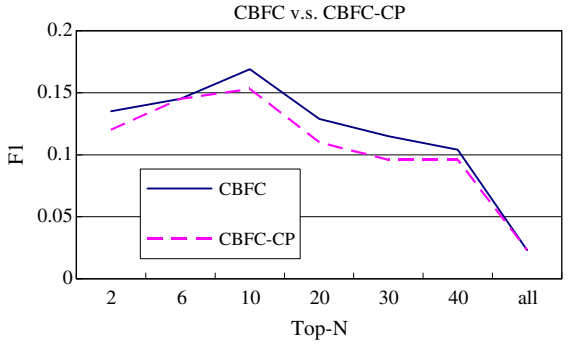

Fig. 4. Comparison of CBFC and CBFC-CP for various top-N recommendations.

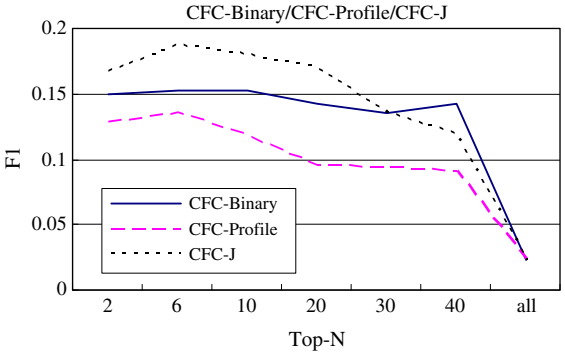

Fig. 5. Comparison of CFC-Binary, CFC-Profile, and CFC-J for various top-N recommendations.

CFC-Binary works better when top-N is larger. Since the number of overlapping documents among different categories is usually small, CFC-J's performance deteriorates as the number of recommended documents increases.

4.2.3. Experiment three: comparison of linear hybrid methods

This experiment compares two hybrid methods with linear combination, HFCL and HFCL-J. The parameters α_{HFCL} and $\alpha_{HFCL\text{-}J}$ are used to adjust the contribution of the predicted ratings from CBFC and CFC/CFC-J, respectively. Based on the highest average F1 values, these parameters are set to 0.4 and 0.6, respectively. Fig. 6 compares HFCL and HFCL-J under different top-N, by setting α_{HFCL} to 0.4 and $\alpha_{HFCL\text{-}J}$ to 0.6. The HFCL method performs better than the HFCL-J method.

4.2.4. Experiment four: comparison of sequential hybrid methods

This experiment compares two sequential hybrid methods, HFCS and HFCS-J. Based on the highest average F1 values, the parameters for HFS and HFS-J are set to 0.2 and 0.0, respectively. Fig. 7 compares HFCS and HFCS-J under different top-N by setting α_{HFCS} to 0.2 and $\alpha_{HFCS\text{-}J}$ to 0. HFCS performs better than HFCS-J.

4.3. Comparing all methods

Fig. 8 compares all the methods under different top-N. The results show that CFC (CFC-Binary) and CFC-J out-

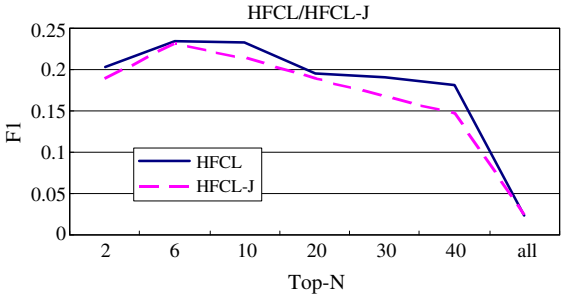

Fig. 6. Comparison of HFCL and HFCL-J.

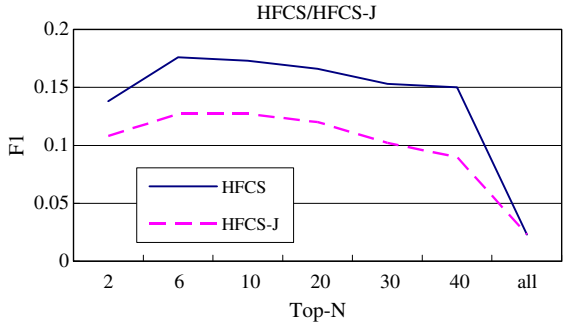

Fig. 7. Comparison of HFCS and HFCS-J for various top-*N* recommendations.

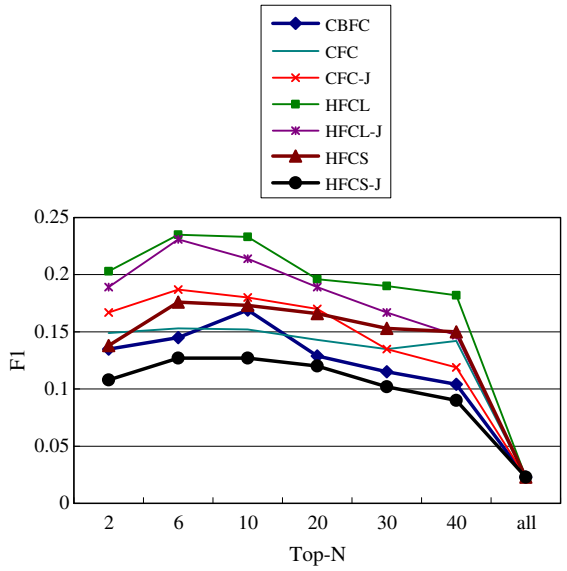

Fig. 8. Comparison of all methods.

perform CBFC. CFC-J performs better when top-*N* is smaller, but CFC's performance is better when top-*N* is larger. The linear and sequential hybrid methods, HFCL, HFCL-J, and HFCS achieve relatively satisfactory results because they combine the advantages of CBFC and CFC. In general, hybrid approaches perform better than pure content-based or collaborative filtering and categorization. Both HFCL and HFCL-J outperform all the other approaches. Although HFCS outperforms CFC, HFCS-J does not outperform CFC-J. In fact, HFCS-J performs even worse than the CBFC method. The sequential hybrid approach does not perform as well as expected. This may be due to the poor construction of the extensible matrix, which is derived from the predicted ratings of the CBFC method.

5. Conclusion and future work

In this paper, we have investigated the issue of recommending documents to appropriate categories in personal folder environments where knowledge workers use their own folders (categories) to organize and store documents. We propose document recommendation methods that facilitate the recommendation and sharing of explicit codified knowledge within a personal folder environment. Recommendations made to such environments need to consider the appropriate category for a recommended document. Most conventional recommendation methods focus on recommending items to users without addressing the issue of recommending items to the target user's appropriate document category. Some methods have addressed the issue by assuming a common category schema without considering personalized categories, or by making recommendations to users first and then determining the categories of the recommended documents. The proposed methods combine recommendation and text categorization techniques to recommend documents to a knowledge worker's personalized categories.

Several existing recommendation methods are adopted and modified by integrating them with text categorization techniques to design the following document recommendation methods: content-based filtering and categorization (CBFC), collaborative filtering and categorization (CF) and hybrid filtering and categorization (HFC) methods. Experiments were conducted to evaluate and compare the performance of these methods using data collected from a research institute laboratory. The experiment results demonstrate that CBFC outperforms CBFC-CP, while CFC-J achieves the best performance among the CFC methods when top-*N* is smaller. Moreover, HFCL outperforms HFCL-J, and HFCS performs better than HFCS-J. Among the hybrid methods, HFCL achieves the best recommendation quality. The hybrid methods, including HFCL and HFCL-J, outperform the pure content-based methods as well as the collaborative filtering and categorization methods.

The proposed recommendation methods can be used to proactively notify knowledge workers about relevant documents from peers and to facilitate push-mode knowledge sharing. Consequently, workers can learn from one another and thereby reduce the effort and manpower involved in searching for documents needed to improve productivity and efficiency when performing knowledge-intensive tasks.

In our future work, we will conduct experiments on a larger data set, i.e., more documents, categories, and users. Currently, the lack of rating information means that ratings in the collaborative filtering method must be presented in binary form. The collection of ratings from users should improve the performance of the collaborative filtering and hybrid methods. The adoption of different classifiers, such as probabilistic models, to determine a user's information needs precisely and route relevant documents to the right folders will also be addressed in our future work. Moreover, document semantics considering the implied meaning of co-occurred keywords in documents will be helpful to facilitate knowledge sharing and document understanding (Zhuge and Luo, 2006). We will adopt document semantics to further improve the recommendation quality in future work. Furthermore, our proposed methods do not utilize

the hierarchical relationships between categories to recommend and allocate documents to folders. In the category hierarchy, the lower level of a category contains documents on more specific subjects, while the upper level contains documents on more general subjects covered by the category. Thus, when recommending documents to the appropriate level of a category, the system needs to consider the subject covered by the category. For example, if the recommendation scores of a document for two sibling nodes are both high, it may be more appropriate to allocate the document to their parent node, since allocating the document to either one of the sibling nodes would not really reflect the subject matter of the document. In our future work, we will extend our scheme by considering the hierarchical relationships and the subjects covered by categories to improve the quality of recommendations.

Acknowledgement

This research was supported in part by the National Science Council of the Taiwan (Republic of China) under the Grant NSC 95-2416-H-009-002.

References

Agostini, A., Albolino, S., Boselli, R., De Michelis, G., De Paoli, F., Dondi, R., 2003. Stimulating knowledge discovery and sharing. In: Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work, pp. 248–257.

Baeza-Yates, R., Ribeiro-Neto, B., 1999. Modern Information Retrieval. Addison-Wesley, Boston, MA.

Bighini, C., Carbonaro, A., Casadei, G., 2003. InLinx for document classification, sharing and recommendation. In: Proceedings of the Third IEEE International Conference on Advanced Learning Technologies, pp. 91–95.

Breese, J.S., Heckerman, D., Kadie, C., 1998. Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence, pp. 43–52.

Burke, R., 2002. Hybrid recommender systems: survey and experiments. User Modeling and User-Adapted Interaction 12 (4), 331–370.

Chen, P.-M., Kuo, F.-C., 2000. An information retrieval system based on a user profile. The Journal of Systems and Software 54 (1), 3–8.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M., 1999. Combining content-based and collaborative filters in an online newspaper. In: Proceedings of the ACM SIGIR Workshop on Recommender Systems: Algorithms and Evaluation.

Davenport, T.H., Prusak, L., 1998. Working Knowledge: How Organizations Manage What They Know. Harvard Business School Press, Boston, MA.

Delgado, J., Ishii, N., Ura, T., 1998. Intelligent collaborative information retrieval. Lecture Notes in Computer Science 1484, 170–182.

Glance, N., Arregui, D., Dardenne, M., 1998. Knowledge pump: community-centered collaborative filtering. In: Proceedings of the Fifth DELOS Workshop on Filtering and Collaborative Filtering, pp. 83–88.

Kamba, T., Bharat, K., Albers, M.C., 1995. The Krakatoa Chronicle: an interactive personalized newspaper on the web. In: Proceedings of the Fourth International World Wide Web Conference, pp. 159–170.

Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J., 1997. GroupLens: applying collaborative filtering to usenet news. Communications of the ACM 40 (3), 77–87.

Lang, K., 1995. NewsWeeder: learning to filter netnews. In: Proceedings of the 12th International Conference on Machine Learning, pp. 331–339.

Langari, Z., Tompa, F.W., 2001. Subject classification in the oxford English dictionary. In: Proceedings of the IEEE International Conference on Data Mining (ICDM), pp. 329–336.

Larkey, L.S., Croft, W.B., 1996. Combining classifiers in text categorization. In: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 289–297.

Lei, Z., Shouju, R., Xiaodan, J., Zuzhao, L., 2000. Knowledge management and its application model in enterprise information systems. IEEE International Symposium on Technology and Society, 287–292.

Lewis, D., Ringuette, M., 1994. A comparison of two learning algorithms for text categorization. In: Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94), pp. 81–93.

Li, Q., Kim, B.M., 2003. An approach for combining content-based and collaborative filters. In: Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages, pp. 17–24.

Liu, D.-R., Shih, Y.-Y., 2005. Hybrid approaches to product recommendation based on customer lifetime value and purchase preferences. The Journal of Systems & Software 77 (2), 181–191.

Melville, P., Mooney, R.J., Nagarajan, R., 2002. Content-boosted collaborative filtering for improved recommendations. In: Proceedings of the 18th National Conference on Artificial Intelligence, pp. 187–192.

Meso, P., Smith, R., 2000. A resource-based view of organizational knowledge management systems. Journal of Knowledge Management 4 (3), 224–234.

Middleton, S.E., Shadbolt, N.R., De Roure, D.C., 2004. Ontological user profiling in recommender systems. ACM Transactions on Information Systems 22 (1), 54–88.

Mooney, R.J., Roy, L., 2000. Content-based book recommending using learning for text categorization. In: Proceedings of the Fifth ACM International Conference on Digital Libraries, pp. 195–204.

Nonaka, I., 1994. A dynamic theory of organizational knowledge creation. Organization Science 5 (1), 14–37.

Rucker, J., Polanco, M.J., 1997. Siteseer: personalized navigation for the web. Communications of the ACM 40 (3), 73–76.

Salton, G., Buckley, C., 1988. Term-weighting approaches in automatic text retrieval. Information Processing & Management 24 (5), 513–523.

Salton, G., McGill, M., 1983. Introduction to Modern Information Retrieval. McGraw-Hill, New York.

Shapira, B., Shoval, P., Hanani, U., 1999. Experimentation with an information filtering system that combines cognitive and sociological filtering integrated with user stereotypes. Decision Support Systems 27 (1/2), 5–24.

Shardanand, U., Maes, P., 1995. Social information filtering: algorithms for automating "Word of Mouth". In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'95), Denver, Colorado, United States, pp. 210–217.

Van Rijsbergen, C.J., 1979. Information Retrieval, second ed. Butterworth, London.

Woodruff, A., Gossweiler, R., Pitkow, J., Chi, E.H., Card, S.K., 2000. Enhancing a digital book with a reading recommender. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 153–160.

Zack, M.H., 1999. Managing codified knowledge. Sloan Management Review 40 (4), 45–58.

Zhuge, H., Luo, X., 2006. Automatic generation of document semantics for the e-science Knowledge Grid. The Journal of Systems & Software 79 (7), 969–983.