

# Document Recommender Agent Based on Hybrid Approach

Khalifa Chekima, Chin Kim On, Rayner Alfred, and Patricia Anthony

**Abstract**—As Internet continues to grow, user tends to rely heavily on search engines. However, these search engines tend to generate a huge number of search results and potentially making it difficult for users to find the most relevant sites. This has resulted in search engines losing their usefulness. These users might be academicians who are searching for relevant academic papers within their interests. The need for a system that can assist in choosing the most relevant papers among the long list of results presented by search engines becomes crucial. In this paper, we propose Document Recommender Agent, that can recommend the most relevant papers based on the academicians's interest. This recommender agent adopts a hybrid recommendation approach. In this paper we also show that recommendation based on the proposed hybrid approach is better than the content-based and the collaborative approaches.

**Index Terms**—Document recommender agent, agent technology, information retrieval.

## I. INTRODUCTION

As Internet continues to grow, more results are presented to users via search engines as there are many potential relevant sites that exist. This has caused search engines to lose their usefulness. According to [1], in the search process, users are often overwhelmed with information overload coming from different sources. The large number of information makes it difficult for users as to decide on the most appropriate and closely related information that they are seeking. Recommender system is able to offer the right and feasible solution for this kind of problem.

Recommender systems, originally, were defined as systems in which users input recommendation, which then are aggregated and directed to the appropriate recipients [2]. The broader implication to the term refers to any system producing more personalized recommendations for users searching within an environment where the amount of available online information surpasses any individual's ability to explore it [3]. CDNow and Amazon.com are among the largest e-commerce online sites that use recommender systems [4].

Another definition of Recommender Systems by [5], in their book titled 'Recommender System Handbook', defines Recommender Systems (RSs) as software techniques and

tools that suggest, to users, items to be of use depending on their search for items to buy, online news to read and music to listen to. The main purpose of recommender systems is to assist individuals who lack adequate knowledge and experience to evaluate the overwhelmingly large amount of choices and alternatives available on the web.

The rests of the paper is organized as follows. In Section II, we give an overview of recommender systems, followed by existing systems and techniques in Section III. Our Document Recommender Agent is described in Section IV. The experimental setup and evaluations are reported in Sections V and VI respectively, and finally the conclusion and future works are elaborated in Section VII.

## II. OVERVIEW OF RECOMMENDER SYSTEMS

Systems using recommendations have been developed in various research projects. One example where recommender system has been deployed is in commercial domain. Amazon.com and ebay.com are good examples of commercial domains using recommender systems to help users during their browsing and purchasing experiences. Many online communities within the movie domain use recommender systems to gather user opinions on movies, and then produce recommendations based on these opinions. MovieFinder2 and Movielens3 are among the online movie communities where viewers' feedback and opinions are used to recommend movies for the users.

Ref. [5] listed six different classes of recommendation approaches, namely content-based approach, collaborative filtering approach, demographic approach, knowledge-based approach, community-based approach and hybrid recommendation approach.

According to [6], content-based and collaborative filtering are the more popular recommendation strategies. Collaborative filtering recommendations are motivated by the observation that we look for our acquaintances for recommendations. On the other hand, content-based filtering depends on rich content descriptions of the items that are being recommended.

## III. EXISTING SYSTEMS AND TECHNIQUES

In [1], they proposed paper recommender system in e-learning domain by considering pedagogical factors, such as the learning background and the paper overall popularity and acceptance to recommend papers to users. In their research, they proved that pedagogical factors improved the recommendation process.

Faculty members and students in educational and

Manuscript received October 20, 2013; revised December 22, 2013. This project is funded by the Ministry of Higher Education, Malaysia under FRG0303-TK-1-2012.

Khalifa Chekima, Chin Kim On, Rayner Alfred are with the Center of Excellence in Semantic Agents, Universiti Malaysia Sabah, Jalan UMS, 88400, Kota Kinabalu, Sabah, Malaysia (e-mail: k.chekima@gmail.com; kimonchin@ums.edu.my; ralfred@ums.edu.my).

Patricia Anthony is with the Department of Applied Computing, Faculty of Environment, Society and Design, Lincoln University, Christchurch, New Zealand (e-mail: patricia.anthony@lincoln.ac.nz).

educational institutions need to find the most relevant paper to their research topics and projects. To help facilitate and simplify the search process, a research paper recommender system would be an ideal tool. Unfortunately, research paper recommender systems have not received enough attention [7]. [7] proposed a system that recommends papers to academicians with topic analysis, which is solely based on content based recommendation approach. In this project, the proposed system managed to score a total of 68.9% in its recommendation accuracy.

As social web sites become popular, researchers and developers have taken this opportunity to develop social websites meant for researchers and scientists. CiteULike is one of the websites of choice among researchers. This social-tagging tool helps academicians and scientists to tag academic papers that interest them, as well as to connect to other CiteULike users that share the same interests. Richard Cameron developed CiteULike in November 2004. For recommendation technique, CiteULike focuses on recommending papers based on like-minded user, which refers to users that share common interest.

Many researchers have shown interest in CiteULike and its effectiveness in helping academicians. Among these researchers are [8] where they conducted an initial evaluation to compare CiteULike with search engine using abstract, title, and tag. Another researcher, [9] studied the usage of tag for research paper recommendations. At the end of the study, the results indicated that the accuracy of the proposed research paper recommendations was 79%. The downside of CiteULike is when an academic user did not explicitly tag and specify his/her interest, no recommendation will be provided by the system.

Other web social tagging websites that had helped academicians to discuss, share and exchange academic papers and opinions are Flickr (developed in 2004) and del.icio.us (developed in 2003). According to Wikipedia, these are free social bookmarking websites that allow members to store searchable copies of web pages and share them with others. Academicians took advantages of these websites to share academic papers as well as to discuss academic related topics among other academicians worldwide. Again, these websites provide recommendation to users based on their interest by exploiting the information provided by the users explicitly. Without explicit information from users on their interests, no recommendation can be given by the system.

A system called BibSonomy is based on Folksonomy. [10] stated that "The term folksonomy refers to a system of classification derived from the practice and method of collaboratively creating and managing tags to annotate and categorize content". Its main purpose is to incorporate team-oriented publication management as well as the features of bookmarking systems. It was developed to help academicians mainly in engineering and information technology to share papers and exchange opinions on computer science and engineering fields. BibSonomy is similar to CiteULike, where it recommends items to users based on like-minded users. The base users' profiles are obtained by tagging their behaviors. Many researchers have shown interest in BibSonomy such as [11] where they analyzed the publication sharing behavior in BibSonomy.

Another group of researchers presented the tag recommendation framework of BibSonomy to evaluate and compare different tag recommendation algorithm in online setting [12].

Ref. [13] developed software to help student researchers. They developed a model of cross language retrieval of information technology domain papers. The system helps students find papers and articles in their field of interest. In their research, they used ACM digital library as their IT classification guide. Their proposed algorithm managed to return a recommendation accuracy of up to 71%. The advantage of this algorithm is, it is straightforward and easy to implement. However, their system has a drawback in the way the corpus is collected and used, as using the training corpus as it is without removing noise leads to wrong classification and recommendation.

Ref. [14] proposed a recommender system that recommends academic papers to academicians within the same lab based on their interest. The recommendation was based on like-minded user, whereby the system monitors users' activities. Users' profiles are generated implicitly, and then mapped with other users' profiles to find similarities among them. The algorithm of subspace clustering approach proposed by these researchers was considered in the proposed agent system due to its simplicity and effectiveness. Yet, there was an obvious disadvantage of the system, where the system functionality would drop every time the number of its users increases and will cause a delay in information updating as well as it might cost additional host and network resource. This is due to their implementation of traditional information retrieval system. The system also suffers from the cold-start recommendation. According to [5], a cold-start recommendation is when a system is unable provide recommendation to users as users did not provide enough feedback and ratings to compute resemblance to other users. Most of systems that rely on collaborative approach will suffer from cold-start recommendation.

Social-network based recommendation shares the same drawback as other recommendation systems that implements collaborative approaches as their recommendation technique. According to [5], social-network based recommendations have the same level of accuracy to those derived from traditional approaches except for cases such as in the cold-start situation, where, for instance, users fail to provide enough ratings to compute similarities to others; another case is when the user's ratings are highly varied.

#### IV. THE DOCUMENT RECOMMENDER AGENT

This recommender agent is in charge of recommending papers to users based on their current interest. It uses a hybrid technique which is a combination of the collaborative filtering and content based filtering techniques.

In the following section, the techniques used in developing the recommender agent are discussed in details, starting from how the user's profile is generated until the recommendation is made based on the user's current interest.

##### A. Recommender Agent Technique

This section is divided into two parts. The first part, focuses

on the techniques involved in constructing an accurate user’s profile to help the Recommender Agent recommends papers to users. The second part discusses the recommendation techniques.

1) User profile generation technique

User profile is the main issue of concern in the process of analyzing how a personal agent assesses a user or makes recommendations [15]. Due to that, constructing an accurate user profile is vital for both collaborative approach and content-based approach.

Learning as much as possible from the user helps the recommender agent provide satisfactory results. However, in reality users are not usually excited about spending time to share the needed information. Moreover, the user’s information and interests may, with time and under certain circumstances, change and in cases be irrelevant. These issues can become huge obstacles in the line of creating and maintaining user profiles in the development of intelligent agent systems. The degree of automation in the attainment of user profiles can be in the form of manual input, semi-automatic procedures, or the automatic recognition by the recommender agents.

In the proposed system, an empty profile structure is implemented. Through the interaction between the user and the system, the profile structure will be filled through an automatic recognition process. When a user downloads or access a PDF paper, the Paper’s ID will be added into user’s profile. The concept of using paper’s ID to construct user’s profile is very effective when dealing with academic domain. The same technique has been adopted by [14], in their system ScuBa and it was very effective in constructing their users’ profiles.

A technique called history-based model was implemented in the proposed system to generate user’s profile. This model is commonly used in e-commerce domains such as ebay and amazon, in which a system keeps a list of purchased items as user profile. As in the proposed system, instead of keeping a history of purchased items by users, the system keeps the users’ downloads of PDF documents in a database as shown in Fig. 1. This information is retrieved by monitoring the users’ downloading behaviors as they download PDF academic papers.

userid	docid	tmstamp
10.6.41.57	132	2011-03-15 14:18:26
10.6.41.57	131	2011-03-15 14:13:53
10.6.41.57	130	2011-03-15 14:08:08
10.6.41.57	137	2011-03-31 08:55:46
10.6.41.57	127	2011-03-15 13:55:10
10.6.41.57	129	2011-03-15 14:05:42
Boo	131	2011-03-15 14:17:43
Boo	132	2011-03-15 14:19:45
debrah	128	2011-03-15 13:57:10
debrah	128	2011-03-15 13:58:18
Gan	129	2011-03-15 14:10:42
Gan	128	2011-03-15 14:05:42
jacob	127	2011-03-15 13:59:10
jacob	127	2011-03-15 13:58:10
khalifa	132	2011-03-15 14:19:57
Khalifa	131	2011-03-15 14:13:53

Fig. 1. ‘userpapers’ table.

Fig. 1 shows the information stored in the ‘userpapers’ table. There are three important information found in ‘userpapers’ table, User’s ID, document’s ID they

downloaded, and the time and date in which the document was downloaded.

2) Recommendation techniques

An intelligent recommender agent uses reliable information such as profile of users to make recommendations, including actions and products, to the user.

In the recommender system, a hybrid approach was implemented that combines both content based filtering approach that is based on content similarity, and collaborative filtering approach that is based on like-minded users to overcome the shortcoming of the two approaches.

This section, will first explain how recommendation based on content similarity/content based filtering is generated, followed by, how the like-minded/collaborative filtering recommendation approach is generated. Lastly, an explanation on how a hybrid approach (combination of content based filtering, collaborative filtering) is developed to produce the recommendation results to users.

In order to generate recommendations based on content similarity, we used the concept proposed by [16]. First, we need to identify the similarity degree between user’s current interest (current paper the user accessed) with the newly downloaded academic paper by other users or papers that exists in the local repository. First, we developed a local repository where each downloaded paper will reside. Next, we used ACM Computing Classification system to identify 11 main classes for computer science paper. These classes cover all computer science topics, each of the class primary keywords are kept in the database in bigrams (The bigrams refer to the primary keys constructed from document’s remaining words after removing noise from a document.) to be compared later with newly downloaded paper’s primary keys.

When a user downloads a paper, first the paper primary keywords will be extracted in terms of bigrams to be compared with all the available classes to identify to which class the paper belongs to. Once the class is identified, and if the newly downloaded paper matches any of the other user’s current interest, the paper will be recommended to these users. As an example, assume that we have two classes, namely Class\_A and Class\_B, and we have a newly downloaded document X. Document X belongs to class A if  $X \cap \text{Class\_A} > X \cap \text{Class\_B}$  as shown in Fig. 2.

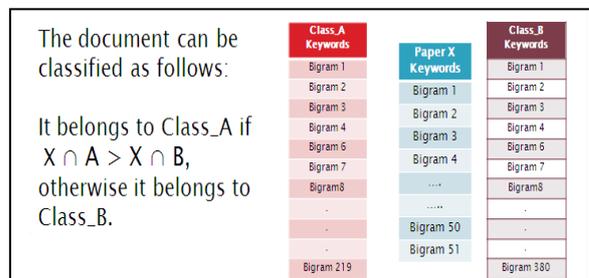


Fig. 2. Paper classification.

As the like-minded user recommendation approach is straightforward; we find people that access similar papers and group them together based on papers they accessed. To be able to find these people, we used the algorithm proposed by [14]. First, we monitor users’ and the paper they accessed. Each of the papers downloaded has a unique ID called

Paper\_ID, which are saved in the repository as shown in Fig. 3.

	primkey	docid	catid
<input type="checkbox"/>	212	127	ACM_Category_D
<input type="checkbox"/>	213	128	ACM_Category_D
<input type="checkbox"/>	214	129	ACM_Category_D
<input type="checkbox"/>	215	130	ACM_Category_E
<input type="checkbox"/>	216	131	ACM_Category_G
<input type="checkbox"/>	217	132	ACM_Category_D
<input type="checkbox"/>	218	133	ACM_Category_D
*	{NULL}	{NULL}	{NULL}

Fig. 3. 'doccat' table in database

Fig. 4 shows users and the papers downloaded. From there we can locate similar users by the same categories or clusters. This can be done by comparing the papers accessed by each user with the rest of the users. Once similarity is identified a recommendation is generated.

userid	docid	timestamp	primkey	
<input type="checkbox"/>	jacob	127	2011-03-15 13:58:10	94
<input type="checkbox"/>	debrah	128	2011-03-15 13:58:18	95
<input type="checkbox"/>	jacob	127	2011-03-15 13:59:10	96
<input type="checkbox"/>	10.6.41.57	129	2011-03-15 14:05:42	97
<input type="checkbox"/>	Gan	128	2011-03-15 14:05:42	98
<input type="checkbox"/>	10.6.41.57	130	2011-03-15 14:08:08	99
<input type="checkbox"/>	Gan	129	2011-03-15 14:10:42	100
<input type="checkbox"/>	10.6.41.57	131	2011-03-15 14:13:53	101
<input type="checkbox"/>	Khalifa	131	2011-03-15 14:13:53	102
<input type="checkbox"/>	Boo	131	2011-03-15 14:17:43	103
<input type="checkbox"/>	10.6.41.57	132	2011-03-15 14:18:26	104
<input type="checkbox"/>	10.6.41.57	133	2011-03-15 14:19:43	105
<input type="checkbox"/>	Boo	132	2011-03-15 14:19:45	106
<input type="checkbox"/>	khalifa	132	2011-03-15 14:19:57	107
*	{NULL}	{NULL}	{NULL}	{NULL}

Fig. 4. 'userpapers' database table.

In the hybrid approach, a recommendation weights is assigned to each of the recommendation approaches based on their recommendation contribution. A weight of 0.75 is assigned to collaborative approach compared to 0.25 assigned to content-based approach. This weight allocation is based on the experiments conducted, where different weight allocation was assigned to both content and collaborative, and a reasonable weight allocation falls between the ratio of collaborative and content respectively of 7:3 and 8:2. The average of this is taken which is 7.5: 2.5 as shown in Fig. 5.

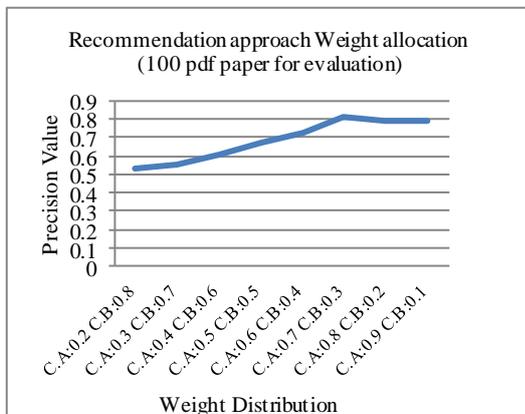


Fig. 5. Recommendation weight allocation.

First, the recommender agent maps user's current interest with similar users in the system to find recommendation based on like-minded users. Once the papers recommendation list

has been identified, the recommender agent will check the category of these papers before recommending them to user. Whenever a category of a paper from collaborative recommendation matches the category of user's current interest, the paper will be recommended to the user based on the weight specified.

As an example, assume a paper is to be recommended to user called User X. This user has accessed paper IDA that belongs to Category\_D (ACM category) based on content based filtering. First, the system will identify the recommendation from like-minded users, say the recommendation retrieved based on like-minded users are paper IDs 8, 9, 10, 11, 12, 13 and 14, assuming these papers are arranged based on their overlapping degree from most relevant to less relevant. These papers will be given a weight of 0.75 to the most relevant assuming 1 is 100%. Say paper ID8 is 80% to be likely recommended to users A based on like-minded users followed by paper ID9 with 75%. At this point, based on the proposed formula, recommendation degree from like-minded users X weight (0.75), paper ID 8 carries 60% to be likely recommended followed by paper ID 9 that carries 56.25%. Next, the system will check the category of these papers. For those papers that has similar category as the user's current paper which is Category\_D, it will be given an additional weight of 0.25. As an example, we take paper IDs 8 and 9 that carry recommendation degrees of 60% and 56.25% respectively. Assume doc ID8 belongs to Category\_D which is similar to user's paper current category. At this point, an additional 25% will be added to paper ID8 to be likely recommended to the user based on 0.25 weight specified in the proposed algorithm making the total percentage of paper ID8 to be recommended increases from 60% to 85%.

This algorithm is designed to increase the accuracy of recommending papers to users to prevent it from recommending something that might not be of interest to the user. In the proposed algorithm, more weights have been given to like-minded users compared to collaborative users. Before assigning the correct weight, a few tests were run to identify the right weight value to enhance the accuracy of recommendation precision. The ratio of 75 to 25 is the best match.

### B. The Recommending Processes

This section explains the steps involved in recommending a paper to user. First, the Recommender Agent will check user's current interest. The Recommender Agent will check for recommendation type. If recommendation based on hybrid approach is available, the recommender system will rank papers based on most relevant and recommends them to user. In case of no recommendation available from the hybrid approach, the Recommender Agent will look into either content based filtering approach or collaborative approach. If collaborative approach has a recommendation based on like-minded users, papers will be ranked based on the most relevant paper at top of the list, then recommendation will be displayed to users. Refer to Fig. 6.

In the case when there is no recommendation based on like-minded users, the system will look at the content based filtering approach and finds similar papers to user's current

interest based on the content similarity of the paper. To prevent from returning a zero recommendation to the user, the system will return the most downloaded paper from the repository based on category similarity. In case there is no similarity found, the recommender system will not return any recommendation to users.

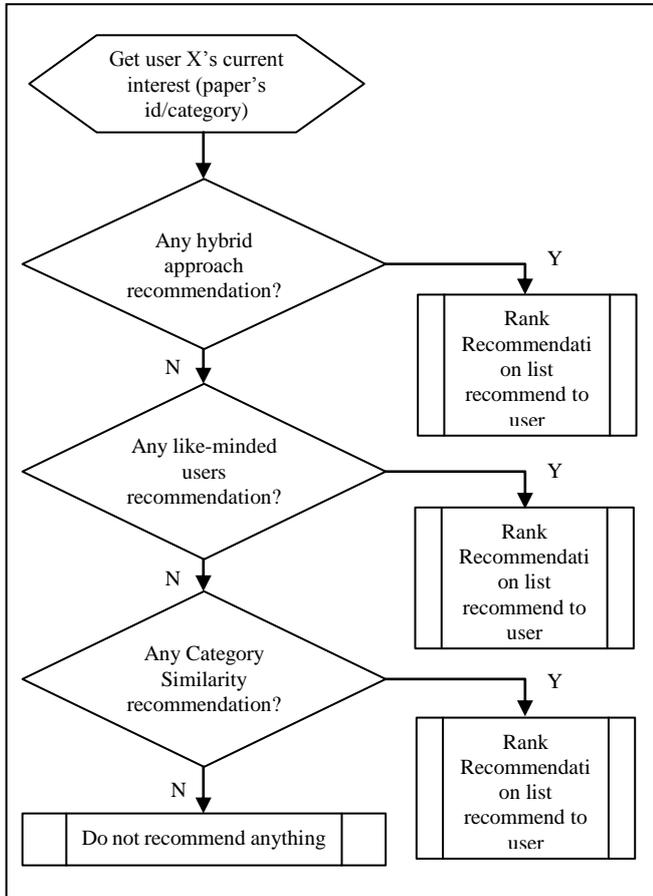


Fig. 6. Document recommendation process.

V. EXPERIMENTAL SETUP

In this experiment, the quality of recommendation is measured using precision matrix which is the ratio between number of relevant results returned and the total number of returned results, such that precision = (correctly recommended articles/total recommended articles). The reason for using precision compared to using precision and recall is due to the system's main the goal of recommending a small amount of relevant information from a vast source of information.

To compare the best performance of the three recommendation approaches namely the content-based approach, the collaborative approach and the hybrid approach, 100 papers were considered as the test data. First, the 100 document was evaluated by the user to identify which among these papers are relevant to the user, and this result was taken as the actual result. Next, the same 100 documents were processed by the Recommender Agent using three different recommendation approaches. The recommendation result from the Recommender Agent is then compared with the actual result retrieved from the user's manual evaluation to find out which among the three recommendation approach has

the nearest value to the actual value.

VI. EVALUATION RESULTS

In Fig. 7, the axis X indicates the number of documents considered in the experiment. Axis Y indicates the value of precision, the value 1 under precision is considered the most relevant and 0 is considered not relevant. The precision curve in Fig. 7 shows that the hybrid approach showed better recommendation then the other two approaches. The three methods showed drops in their precision values as the number of papers increases. This is due to two major factors. The first one is due to the existence of noise or non-computer science paper, and the second factor is as number of documents increases, the chance of the system to be likely recommending irrelevant documents increases as well. However, as can be observed the hybrid approach did not drop as much as the other two approaches.

VII. CONCLUSIONS AND FUTURE WORKS

The hybrid recommendation technique implemented in the proposed agent system has improved the recommendation accuracy compared to recommendation systems that uses single recommendation techniques (either collaborative recommendation or content based recommendation).

While recommendation based on hybrid showed good result, there is a need to widen the recommendation to cover more academic topics rather than limiting the recommendation to just Computer Science.

ACKNOWLEDGMENT

This project is supported by Center of Excellence in Semantic Agents (COESA), Universiti Malaysia Sabah and in collaboration with MIMOS Berhad.

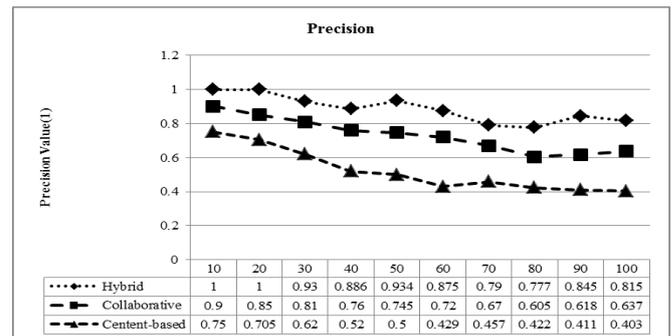


Fig. 7. Comparison of precision values between various recommendation approaches.

REFERENCES

- [1] T. Y. Tang and G. McCalla, "A Multidimensional Paper Recommender: Experiments and Evaluations," *IEEE Internet Computing*, vol. 13, no. 4, pp. 34-41, 2009.
- [2] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56-58, 1997.
- [3] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331-370, 2002.
- [4] N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, "Combining collaborative filtering with personal agents for better recommendations," *AAAI/IAAI*, pp. 439-446, 1999.

- [5] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," *Recommender Systems Handbook*, US: Springer, 2011, pp. 1-35.
- [6] J. O'Donovan, and B. Smyth, "Trust in recommender systems," in *Proc. the 10th International Conference on Intelligent User Interfaces*, 2005, pp. 167-174.
- [7] C. Pan and W. Li, "Research paper recommendation with topic analysis," in *Proc. the International Conference on Computer Design and Applications*, 2010, vol. 4, pp. v4-264.
- [8] P. Jomsri, S. Sanguansintukul, and W. Choochaiwattana, "A comparison of search engine using "tag title and abstract" with CiteULike—An initial evaluation", in *Proc. the International Conference on Internet Technology and Secured Transactions*, 2009, pp. 1-5.
- [9] W. Choochaiwattana, "Usage of tagging for research paper recommendation," in *Proc. the International Conference on Advanced Computer Theory and Engineering*, 2010, vol. 2, pp. V2-439.
- [10] P. Isabella, *Folksonomies: Indexing and Retrieval in Web 2.0 Handbook*, Walter de gryter GmbH & Co, 2009.
- [11] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme, "Tag recommendations in folksonomies," in *Proc. the Knowledge Discovery in Databases*, pp. 506-514, Springer, 2007.
- [12] L. B. Marinho, A. Hotho, R. Jäschke, A. Nanopoulos, S. Rendle, L. Schmidt-Thieme, and P. Symeonidis, *Social Tagging Systems in Recommender Systems for Social Tagging Systems*, US: Springer, 2012, pp. 3-15.
- [13] G. Kembellec, I. Saleh, and C. Sauvaget, "A model of Cross Language Retrieval for IT domain papers through a map of ACM computing classification system," in *Proc. the International Conference on Multimedia Computing and Systems*, 2009, pp. 162-168.
- [14] N. Agarwal, E. Haque, H. Liu, and L. Parsons, "Research paper recommender systems: a subspace clustering approach," *Advances in Web-Age Information Management*, UK: Springer, pp. 475-491, 2005.
- [15] M. Montaner, B. López, and J. L. De La Rosa, "A taxonomy of recommender agents on the internet," *Artificial Intelligence Review*, vol. 19, no. 4, pp. 285-330, 2003.
- [16] K. Chekima, K. O. Chin, R. Alfred, K. S. Gan, and P. Anthony, "Document categorizer agent based on ACM hierarchy," in *Proc. International Congress on Control System, Computing and Engineering*, 2012, pp. 23-25.



**Khalifa Chekima** received his master degree in computer science recently with the Center of Excellent in Semantic Agents under School of Engineering and Information Technology, in Universiti of Malaysia Sabah, Sabah, Malaysia. The author's research interests include agent technologies, and document categorizer.



**Chin Kim On** received his PhD in artificial intelligence with the Universiti of Malaysia Sabah, Sabah, Malaysia. The author's research interests included gaming AI, evolutionary computing, evolutionary robotics, neural networks, image processing, semantics based visual information retrieval, agent technologies, evolutionary data mining and biometric security system with mainly focused on fingerprint and voice recognition.

He is currently working as a senior lecturer at the Universiti Malaysia Sabah in the School of Engineering and Information Technology, Sabah, Malaysia. He has authored and co-authored more than 60 articles in the forms of journals, book chapters and conference proceedings. He is a member of IEEE and IAENG societies.



**Rayner Alfred** was born in Kota Kinabalu, Sabah. He completed a PhD in 2008 looking at intelligent techniques to model and optimize the complex, dynamic and distributed processes of knowledge discovery for structured and unstructured data. He holds a PhD degree in computer science from York University (United Kingdom), a master degree in computer science from Western Michigan University, Kalamazoo (USA) and a Computer Science degree from Polytechnic University of

Brooklyn, New York (USA).

Dr. Rayner leads and defines projects around knowledge discovery and information retrieval at Universiti Malaysia Sabah. One focus of Dr. Rayner's work is to build smarter mechanism that enables knowledge discovery in relational databases. His work addresses the challenges related to big data problem: How can we create and apply smarter collaborative knowledge discovery technologies thvcat cope with the big data problem.

Dr. Rayner has authored and co-authored more than 75 journals/book chapters and conference papers, editorials, and served on the program and organizing committees of numerous national and international conferences and workshops. He is a member of the Institute of Electrical and Electronic Engineers (IEEE) and Association for Computing Machinery (ACM) societies.



**Patricia Anthony** received her PhD in computer science from the University of Southampton in 2003. She is currently working as a senior lecturer at the Department of Applied Computing, Lincoln University, New Zealand. Her research interest is in semantic agents and multi-agent systems and how these agents can interact with each other within an open domain to solve problems. She is also interested in investigating how agents can communicate with each other at the semantic level using semantic technology. To date, she has published more than 80 articles in the forms of journals, book chapters and conference proceedings. She is a member of IEEE, ACM and IACSIT.