World Applied Sciences Journal 9 (11): 1260-1271, 2010
ISSN 1818-4952

Learning Materials Recommendation Using a Hybrid Recommender System with Automated Keyword Extraction

Ehsan Emadzadeh, Azadeh Nikfarjam, Khairil Imran Ghauth, Ng Kok Why

Faculty of Information Technology, Multimedia University, 63100 Cyberjaya, Malaysia

Abstract: A huge amount of learning materials available in e-learning environment has led to a greater challenge to find suitable learning materials for the learners. Recent trend shows that the existing e-learning recommender system can only recommend the items with similar learning content, however other key items with different learning contents may not be recommended. Moreover, to recommend any item using manual recommender systems require the author to provide description for every item that consumes a lot of time and manpower usage. We address all these issues by designing a novel framework of hybrid e-learning recommender system based on combination of collaborative filtering (CF) and content based filtering (CBF), which involves following recommendation strategies: (i) to recommend similar item that remains in the learning context (ii) to extract the keywords automatically from the text-based documents and (iii) to minimize the time taken to provide the keywords for the items. The obtained results show that the proposed recommender system with automated keyword extraction can achieve higher accuracy of about 17.04% for mean absolute error (MAE) and 28.71% for mean square error (MSE) in terms of rating deviation as compared to the manually entered keyword.

Key words: E-learning · Recommender System · Hybrid Filtering · Collaborative Filtering · Content Based Filtering · Keyword Extraction

INTRODUCTION

A huge amount of learning materials available in e-learning environment has led to the difficulty of finding suitable learning materials for the learners. Instead, the researchers are currently moving their focus from authoring the content to recommend the suitable learning content [1-3]. One way of recommending learning materials to learners is done by using a recommender system. Recommender systems have been successfully adopted in many e-commerce website (e.g. ebay.com and amazon.com) to recommend items to the customers. These include recommending new item, interesting item, promotion item and item with similar content. On the other hand, recommender system in e-learning differs from other domains in such a way that the recommender system must consider the learning context and the learning sequence. The recommended items should be of interest to the current viewing item and the recommender system should be able to return to the previous state of learning sequence after viewing the recommended items. Figure 1 shows the recommendation process in e-learning, in which after the learners have viewed the recommended items, it will return to the previous learning state.

The recommender system can be divided into two main recommendation techniques [4]: (i) collaborative filtering (CF) and (ii) content-based filtering (CBF). The CF recommends the items based on other user preferred items that have similar taste to the active user, thus a new item might be recommended to the active user. However, CF suffers two problems: (i) the items that have not received any rating from the users will not be recommended and (ii) the items that have received less rating might give a poor recommendation result. On the other hand, CBF recommends items based on the content similarity, thus items with only similar content will be recommended and other interesting items with different contents are not able to be recommended. To overcome from the above mentioned drawbacks, a combination of CF and CBF, namely, hybrid filtering can be used for item recommendation. In general, hybrid filtering makes use of user similarity to recommend the items when the rating for the item exists and sufficient (exceed the threshold) enough to be recommended. Otherwise, the similarity between the content of the items will be used as recommending the items.

Corresponding Author: Khairil Imran Ghauth, Faculty of Information Technology, Multimedia University, 63100 Cyberjaya, Malaysia, Tel: +603-8312 5403, Email: kghauth@gmail.com.

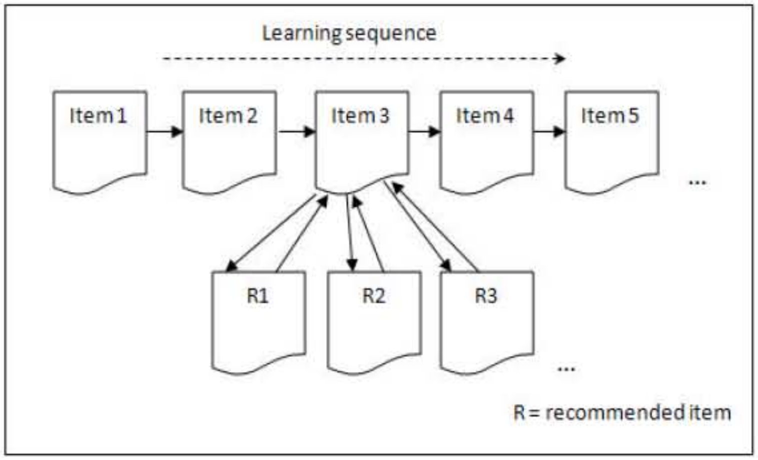

Fig. 1: Recommendation process in e-learning

To recommend any item using content-based recommender system, the author needs to provide the description or keywords for each item. Currently, a lot of learning materials, such as word documents and presentation slides are created without any document description (metadata) [5]. To provide description for every item consumes a lot of time and manpower usage. To overcome from these issues, keyword extraction technique can be used to extract the keywords from the text-based documents. In general, keyword extraction will scan the documents, remove the stop words, apply word stemming and the remaining word will be broken down into individual unique keyword. The tradeoff between manually entered keyword by the author and automatically extracted keyword by keyword extraction is the document description accuracy and time taken to provide the keyword [6]. Manually entered keyword will have better document description accuracy but consumes more time. On the other hand, automatic keyword extraction will consume less time with the expense of document description's accuracy.

In this paper, a new hybrid recommender system is proposed that utilizes the automatic keyword extraction technique to extract the keywords. This paper focuses on two main objectives: (i) to recommend similar item that remains in the learning context and (ii) to minimize the time taken to provide the keywords for the items. The proposed system is evaluated using a training set of learning documents. The results analysis obtained by using the test data show that the proposed recommender system with automatic keyword extraction can achieve higher accuracy of about 17.04% for mean absolute error (MAE) and 28.71% for mean square error (MSE) in terms of rating deviation as compared to the recommender system with manually entered keyword. Moreover, the proposed recommender system takes minimum time of about 12.83 seconds (in average) to extract the keywords automatically when the size of the file increased to 671 KBytes.

In summary, the paper proposes an e-learning recommender system based on the combination of CF and CBF that utilized automatic keyword extraction to extract keywords. The paper also describes the technology to assist users in finding the relevant document based on other learners recommendations and also using the basic document-to-document matching strategy. Nonetheless, keyword extraction is the main idea in this study and the content of this research is more related to information retrieval or natural language processing areas, though the proposed system can usefully apply the e-learning recommend system.

The remaining part of this paper is organized as follows. Section 2 describes the detailed background information of the e-learning recommender system and the focus of this paper. Section 3 introduces overall system architecture. Moreover, this section also includes descriptions of about keyword extraction, similarity calculation and matrixes, rating prediction and document suggestion. Section 4 presents analysis of data and results. This section explains the detailed information of benchmarking experiment setup and details of implementation of recommendation mechanism. The obtained results are also compared with manual recommender system to evaluate the performance of the proposed recommender system. Finally, Section 5 provides the concluding remarks along with applications of the proposed system as well as suggestions for future work.

Related Works: The attempt to discover and recommend accurate documents for the users has been a hot topic since few years ago. Many successful methods have been proposed to meet the users' desired documents. One of the methods is through extracting important keywords from the documents-examples: filtering insignificant words or recognizing high repetitive words as the keywords of the document as concerned. These methods devoted pros and cons to the system. In the following sub-sections, we discuss them in detail and relate them to our proposed method. Besides, we also discuss various techniques used in recommender systems including recommender systems used in e-learning.

Keyword Extraction: Keyword is the main descriptor to capture the essence of the topic of a document. Conventionally, keywords are produced through analyzing the document either manually with subject indexing or automatically with automatic indexing. Subject indexing described a document by index terms to indicate what the document is about, while automatic indexing stores data in a database to ease the information retrieval. In recent years, a more sophisticated method is through keyword extraction. We have surveyed many related research work and in the following, we only discuss a few recent work that are closely related to our area.

Vladislav [7] implemented keyword extractor in their project of Language Technology for e-Learning (LT4eL). They first applied morpho-syntactic annotation-based filtering to filter uncommon keyword types (e.g. isolated prepositions, conjunctions and numerals); then followed by scoring function to select the most probable keyword candidates. They also used TF*IDF measure and ARIDF (adjusted reduced inverse document frequency) measures to extract the information. Furthermore, ARF (average reduced frequency) measure is absorbed into their algorithm to assign regular corpus frequency to words distributed evenly in the entire corpus. In our study, we are not using ARF measure as this will slow down the extraction process and we deem that this does not aid much in our work.

Lemnitzer [8] created keyword extractor to handle documents in 8 languages. Their implementation is similar to Vladislav [7] to filter candidate keywords and rank them based on frequency criteria. The only difference is that, they select the relevant keywords in a document which has been complemented with a linguistic processing step. Lemnitzer [8] performed three tests to validate inter annotator agreement for determining the complexity of the task and the acceptance of the extracted keywords by the users. The authors claimed that the results are still promising. The reader can read more description of the tools created by them for keyword extraction and detection [9].

In [10], the supervised learning approach is suggested for keyword extraction to combine parameterized heuristic rules with a genetic algorithm into a system-GenEx, which automatically extracts the keywords from the text.

Meanwhile in [11], the supervised and unsupervised graph-based approaches are evaluated for cross-lingual keyword extraction to extract summarization from the text documents. The author claimed that, when large training set is available, the supervised approach will produce more accurate salient keywords and vise versa. In our approach, we perform unsupervised learning algorithm.

Frank [12] employed naive Bayes learning scheme to automatically tailoring the extraction process. The quality of the extraction improved significantly when domain specific information is exploited. We do not employ this approach as it has domain dependency.

Wan [13] combined the activity of summarization together with keywords extraction from a single document with the assumption that the summary and keywords of the document can be mutually boosted. However, only the mutual information measure was used to compute the word semantics.

For more principles on the keyword extraction techniques, the author in [14] clearly pointed out the importance of the quantitative techniques, such as concordances, statistical relationship and the organization of the formal linguistic materials and the qualitative techniques, such as semantics analysis for the central of various keyword extraction techniques.

In our experiment, we only deal with mono-lingual input text. Our approach will trade off between the complexity and the accuracy of the output data. We use TF*IDF measure from [7] and CBF measure to find the similarities of the given documents. The detail of these two measures will be elaborated in details in Section 3.2.

Recommender System: With the efficient of keyword extraction technique, the next step is to work with the effective recommender system which will recommend top-N documents for the active document. The recommender system (or more commonly identified as information filtering technique) is an attempt to present information items, which are likely of interest to the users. This approach compares the user's profile to some reference characteristics and predicts the rating given by a user towards an item that they have not yet considered. Among the most popular characteristics are the information item (content-based or item-based approach) and user's social environment (collaborative approach). Adomavicius [4] classified recommendation methods into three main categories: firstly, content-based recommendation (recommended items similar to the ones the user preferred in the past); secondly, collaborative recommendations (recommended items that people with similar tastes and preferences liked in the past); and finally, hybrid approaches (combination of collaborative and content-based method).

Generally, pushing techniques to be more and more accurate require deepening their foundation, while reducing reliance on arbitrary decisions. Item-based approach considers the similarity of those items to the targeted item. A weighted average from the ratings of similarity is computed and sorted to facilitate the top listing of the related items. van Meteren [15] introduced item-based filtering on large web site searching. They proposed PRES (personalized recommender system) to create dynamic hyperlinks for the web site by referring to a list of advises to ease the users finding for their interesting items. In their paper, they claimed that item-based filtering cannot predict future interests accurately due to several terms or many terms may have more than one meaning.

Badrul [16] proposed item-based filtering method by analyzing the user-items. With the relationships, they indirectly computed the recommendations for the users. They claimed that item-based algorithm works better than user-based algorithm, but not significantly large.

Roberto [17] approached the hybrid method to recommend the research papers. In their findings, different algorithms are more suitable for recommending different kinds of papers instead of an individual algorithm. They claimed that user's research experience played the most important influences the way users perceived recommendations. Besides, cultural differences of the users do not have much significant to influence the system, however, if the interface of the system appeared within the user's native language regardless the language of the papers are written, it does make any difference. Einarsson [18] also employed hybrid method to personalize the content of the MobileTV. In his finding, the rating value is of less importance than the order of the preferred items as different users will have different rating values in them but not the order of the items.

More interestingly, Shengchao [19] performed user-based collaborative filtering method to predict user's item ratings. They improved the prediction accuracy based on the statistical prediction errors, namely PEBE (prediction error-based enhancement method). Two strategies were proposed: (i) item similarity strategy and (ii) counting number of common neighbors. They claimed that the strategy of considering number of common neighbors outperformed the standard user-based CF method. Another work on statistical approach is presented by MingLi [20]. They took into consideration of the conditional probability of an item with respect to the user who had pre-known of other item. Their main concern is on the item itself rather than a bunch of items from the conditional perspective. The statistical approaches are impressive but we have not moved into that far in this study.

Of all the different approaches discussed above, we turn our direction into hybrid-based approach as this approach allows recommendation for items that have not yet received any rating (to solve the drawbacks of the collaborative approach) and allows dissimilar contents of items to be recommended (to enrich the content-based approach). The procedure will be described in more detailed in the next section of our proposed methodology.

Proposed Methodology: Figure 2 shows the overall system architecture of the proposed recommender system. Once a user selects a document to read, a set of related documents will be suggested to the user and the user can rate the suggested documents based on the degree of relevancy and interest. For recommending related documents, the proposed method uses a hybrid combination of CF and CBF. We consider CF which is based on similarities between the users (which means how similar the user interests are).

On the other hand, we consider the CBF where the similarities between the documents are calculated and stored in a document-document matrix. The similarities between the documents are calculated based on the common keywords extracted from both documents.

The system gives a predicted rating to all the documents which have not received a rating from the users and the users can rate the suggested documents. The results obtained from the predicted ratings will be compared with the real user ratings to calculate the rating accuracy.

In the following sections, the architecture of the system will be discussed, followed by keyword extraction, similarity calculation and matrixes, rating prediction and document suggestion.

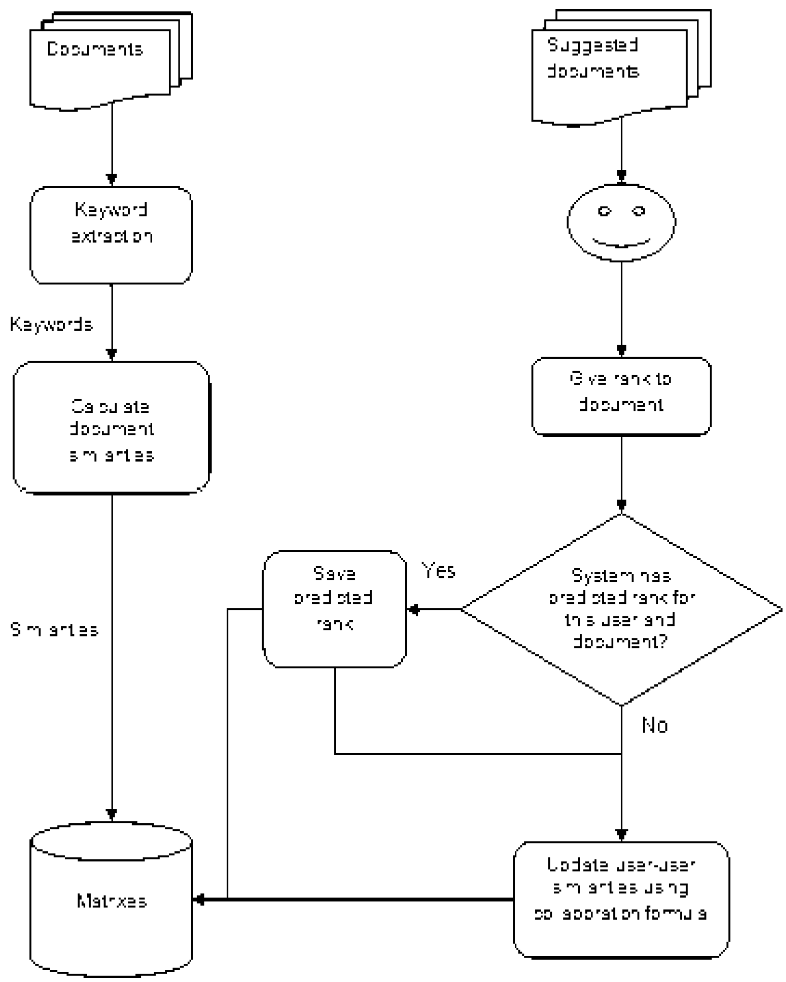

Fig. 2: Overall system architecture of the e-learning recommender system

Recommendation Framework: The recommendation framework is depicted in Figure 3. There are four main matrixes involves in this study: user-user, document-document, user-document and rank-difference matrixes. The user-user matrix keeps user similarities, the document-document matrix contains document similarities, the user-document stores real or predicted users` ratings and the rank-difference matrix stores the difference between predicted rating and real rating. Detailed information about how to construct the user-user and the document-document matrixes will be discussed in 'similarity calculation and matrixes' section, followed by a description about predicted ratings in 'rating prediction' section.

The document-document matrix stored the similarity values between the documents which will be populated after all the documents have added to the system. When a user gives a rating to a document, it will be saved in user-document matrix and the user-user matrix will be updated. Also, the predicted rating of the user for a document will be stored in a user-document matrix with. After users have given the ratings to the documents, the real rating will overwrite with the predicted rating and the value is stored in the user-document matrix. In addition, the difference between the real rating and the predicted rating will be saved in a rank-difference matrix for further analysis.

Extraction of Keywords: As mentioned earlier, the research goal is to compare the results of the system's prediction based on the extracted keyword method. The keyword extraction will be done in two ways as follows: (i) manual extraction and (ii) automatic extraction.

For the manual keyword extraction, the instructor will prepare the keywords and upload the keywords together with learning materials to the system.

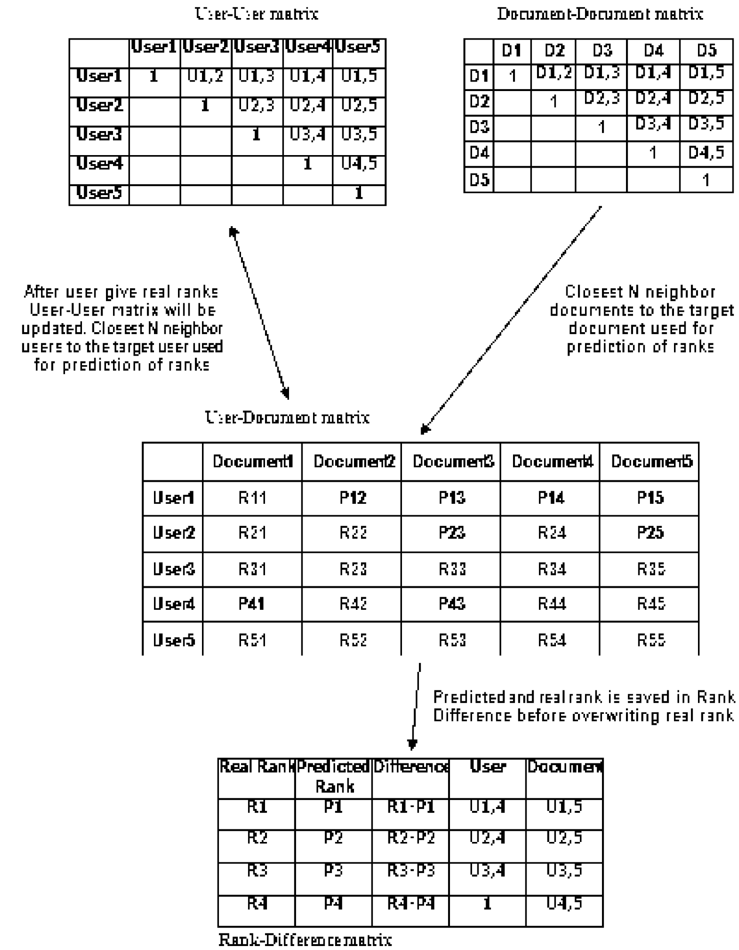

Fig. 3: The recommendation process flow

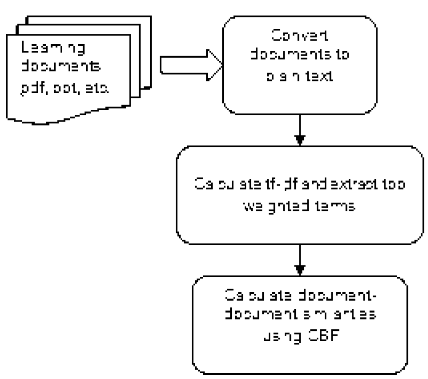

Fig. 4: Automatic term extraction

On the other hand, automatic extraction of keyword (as shown in Figure 4) consists of the following steps:

* Extracting all the words from the document.
* **Removing the Stop Words:** Most commonly used words, such as articles, pronouns and prepositions will be removed from the document.
* **Stemming:** Stemming is based on the Porter's algorithm in which the main idea is that, all the suffixes in English should consist of simpler suffixes. The algorithm consist of five stages, in each stage a set of rules will be tested on the word and if a rule condition is satisfied based on the way defined by the rule, the suffix will be removed from the word. The resulting stem will be returned after passing all five stages. For more information about Porter's algorithm, we may refer to [21].
* Returning the remained words as the extracted keywords: The keywords will be saved in the database.

Similarity Calculation and Matrices: In order to obtain high quality recommendations, the method proposed in this study uses a combination of CBF and CF. The following sections describes

Table 1: User-User similarities matrix based on CF.

	User 1	User 2	User 3	User 4	User 5
User 1	1	0.25	0.62	0.77	0.56
User 2		1	0.09	0.88	0.36
User 3			1	0.21	0.12
User 4				1	0.51
User 5					1

```
maximum_rank=5
for each user(i) and user(j) {
      accumulative_rank_difference=0;
      number_of _commonly_ranked=0;
      for each user(i),user(j) commonly rated
      documents(k){
            accumulative_rank_difference =
            accumulative_rank_difference +
                  absolute(rank user(i) for
                  document(k) - rank user(j)
                  for document(k))
            number_of _commonly_ranked =
            number_of _commonly_ranked + 1
      }
      max_match_value = number_of
      _commonly_ranked * maximum_rank;
      similarity user(i) and user(j) = (
      max_match_value -
      accumulative_rank_difference)/
      max_match_value
      store similarity between user(i) and
      user(j);
}
```

Fig. 5: Algorithm for similarity calculation

the relation between the users and the documents, followed by an explanation on how the document-document and the user-user matrixes are constructed.

Similarity Between Users: As part of CF, the first step is to calculate the similarities between the users, based on the users' ratings on documents (as illustrated in Table 1). The ratings history of each user can be considered as an N-dimension vector, where N is the number of documents in the system. First of all, the algorithm calculates the distance between the two users by using Manhattan distance [22] and normalizes the distance to a range between "0" and "1"; subsequently calculates similarity by reducing the distance from "1" (which is the maximum possible normalized distance).

After constructing the user-user matrix, the top-N similar users to a target user can be identified. The algorithm for similarity calculation is shown in Figure 5.

The number of documents that have rated by both the users has a great influence on the accuracy of the similarity results. The commonly rated documents are selected and hence, the similarity will be calculated from equation (1) as follows.

$$sim(user(i), user(j)) = \frac{(N * R_{\max}) - \sum_{x=1}^{N} | r_i(x) - r_j(x) |}{N * R_{\max}} \quad (1)$$

where N is the number of commonly rated documents, R_{max} is the maximum rated value (the maximum rated value is assumed to 5), $r_i(x)$ is the rating of $user_i$ for document x and $r_j(x)$ is the rating of $user_j$ for document x.

Similarity Between Documents: For calculating similarity between the documents, the proposed method is based on the algorithm reported in [16] with some modifications. In this approach, different attributes of a document, such as the author, the title and the keywords are considered. A weight is assigned to each of these attributes and then the similarity between the documents will be calculated using equation (2) as below.

$$sim(document(i), document(j)) = \sum_{x=1}^{N} w_x * \frac{a_x(i,j)}{b_x(i,j)} \quad (2)$$

where:

- ***N*:** Number of attribute types (number is assumed to 3: author, title and keyword)
- **w_x:** Weight for attribute of type x
- **a_x:** Number of common type of x attributes for document i and j
- **b_x:** Minimum number of attributes of type x between document i and document j

By calculating the similarities between each pair of documents, the system can construct the document-document matrix. After adding each document to the system, the similarities between the newly added document and the other documents in the system will be calculated and will be added to the document-document matrix (Figure 3).

Rating Prediction: As discussed earlier, the proposed recommender system stored the user ratings for documents in the user-document matrix. As can be seen from Figure 3, there are white and grey cells. The white cells denote the ratings are the real users' ratings while grey cells denote the predicted ratings by the system (and will be overwritten once users rate the document).Futhermore, it can be seen that some of the documents have not received any rating by some users. Thus, the final recommendations will be solely based on the rating predictions. To calculate the predicted rating of *user*$_i$ for *document*$_j$ will consists of the following steps:

- Find the top-N similar users to *user*$_i$, who have rated *document*$_j$
- Find the top-N similar documents to *document*$_j$ which have received rating from *user*$_i$
- Calculate prediction based on user similarities (refer equation (3))
- Calculate prediction based on document contents' similarities (refer equation (4))
- Calculate final predicted rating by combining the step iii and iv prediction results (refer equation (5))

The proposed recommender system will apply the threshold in selecting similar users and documents to improve the quality of prediction. The method proposed in this study is relatively common like other recommendation systems reported in [23].

In predicting user ratings, the similarity value between the users will be used as weighting constants.

Firstly, the proposed recommender system will considers CF in which the top-N similar neighbours to *user*$_i$ will be selected from the user-user matrix. The CF calculations will be done based on the following equation.

$$PCollaborative_{i,j} = \frac{\sum_{n=1}^{N} S_{i,n} * R_{j,n}}{\sum_{n=1}^{N} S_{i,n}} \tag{3}$$

where:

$S_{i,n}$: Similarity value between *user*$_i$ and *neighbor*$_n$

$R_{j,n}$: The rating of *neighbor*$_n$ to *document*$_j$

From equation (3), the predicted rating of CF will be calculated. In order to calculate the predicted rating, there should be at least one neighbor for *user*$_i$ who has given rating to *document*$_j$. After a user has given a rating to a document, the neighborhoods and then collaborative predicted rating will be updated.

The next step is to select the top-N similar neighbor documents from the document-document matrix which have received ratings from the *user*$_i$ and is calculated as follows.

$$PContent_{i,j} = \frac{\sum_{n=1}^{N} S_{i,n} * R_{j,n}}{\sum_{n=1}^{N} S_{i,n}} \tag{4}$$

where:

$S_{i,n}$: Similarity value between *document*$_i$ and document *neighbor*$_n$

$R_{j,n}$: The rating of *user*$_j$ to document *neighbor*$_n$

By using equation (4), the system will find another prediction factor of hybrid system that is based on content based prediction. To predict based on the content, user should give at least one rating to similar documents. The system will update the prediction after users have given a rating to a new document.

Finally, the system combines the results by using weighted average. It uses the number of neighbors as weighting factors. Equation (5) shows the calculation.

$$P = \frac{PCollaborative * N_1 + PContent * N_2}{N_1 + N_2} \tag{5}$$

where:

N_1: Number of selected user neighbors

N_2: Number of selected document neighbors

The equation (5) will calculate the predicted rating for *user*$_i$ and *document*$_j$ by combining predicted ratings based on the content of documents (equation (4)) and user's behavioural (equation (3)) similarity.

Document Suggestion: As mentioned in introduction section, when a user reads a document, the system suggests a set of related documents. The aim is to suggest documents that satisfy two conditions: i) the user has not seen the suggested document and ii) the suggested documents are the most related documents to the viewing document. To meet these conditions, the documents are selected based on the value in the user-document matrix, where the documents have not received a real rating by the user and followed by selecting *N* documents that have the highest predicted ratings.

Both the similarity between users and the similarity between documents are used to calculate the predicted user ratings for a particular document. Document suggestion is based on the predicted ratings and both relevancy and user interests are considered for documents suggestion.

RESULTS

The proposed system is evaluated using a prepared training set of learning documents. The documents heading and file name are changed to reduce the probability of prejudgment of users for the respective documents, so that the users need to read the contents to before giving ratings. This will avoid the user to give rating base on file's name or heading rather than the content itself. The training set includes 40 documents, which are in different forms but in a text-based, such as lecture notes, slides and articles.

We have divided our experiment into two parts: (i) automatic keyword extraction and (ii) manual keyword extraction. For the manual keyword extraction, three lecturers have been assigned to extract the keywords from the document. And for automatic keyword extraction, the documents are converted into plain texts and become as the inputs to the system for extracting the keywords.

The system is tested using 68 software engineering students. Out of 68 students, 34 users are assigned to test the automatic extracted keyword system while the remaining users are assigned to test the manual extracted keyword system. The students (users) normally read the documents and rate them according to the relevancy and usefulness. Figure 6 shows the prototype of our implemented recommendation system.

To measure the rating deviation between the predicted and real rating from users, the MAE and MSE are calculated by using the following equation 6 and 7 respectively.

Table 2: Obtained results of MAE and MSE

	Mean Absolute Error (MAE)	Mean Square Error (MSE)
Automatic Keyword Extraction	0.924791	0.161154
Manual Keyword Extraction	1.114709	0.226062

$$MAE = \frac{1}{N} * sum(absolute(realrank(i) - predictedrank(j))) \quad (6)$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N} e_i^2 \quad (7)$$

The result obtained for MAE and MSE calculations are presented in Table 2.

As we can see, the automated keyword extraction gives better prediction than the results obtained by using manual keyword extraction. For better view, the obtained results are summarized in Figure 7 and Figure 8, respectively.

Figure 7 shows the average of the predicted ratings against the real ratings (ranges from 1 to 5). In Figure 8, the deviation of average predicted ratings from real ratings is depicted. It is easy to see from Figure 8 that, the deviation for real ratings of 3 and 4 are the lowest for both automatic and manually extracted keyword. However, automated keyword extraction technique gives better accuracy as compared to manual keyword extraction. It is further revealed that automatically extracted keyword performs better for higher rating value of 5 while manually extracted keyword is not achieving better results for higher rating. For lower rating of 1 and 2, both the algorithms have high deviation from the real ratings.

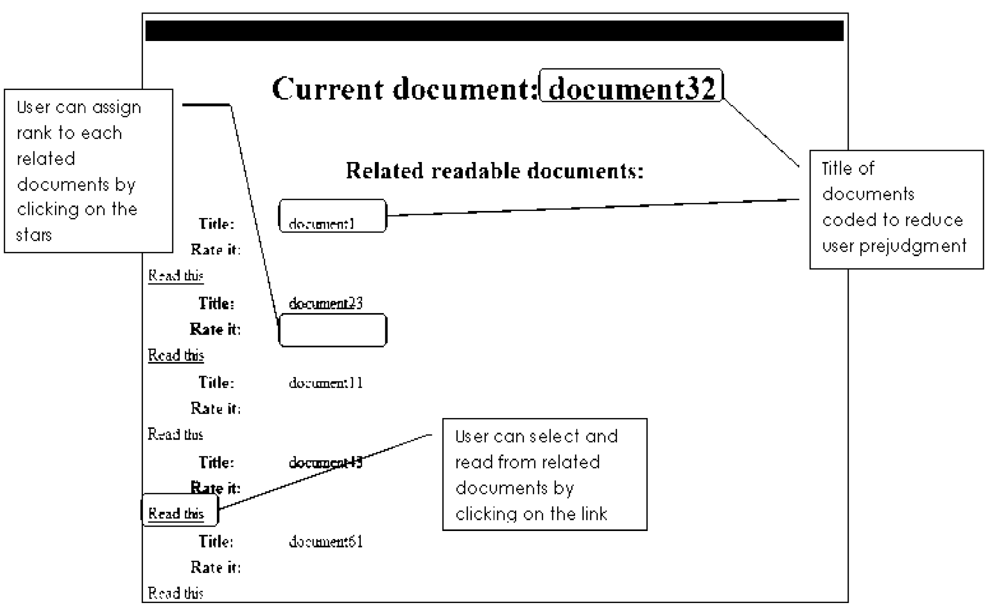

Fig. 6: Prototype of the recommendation system

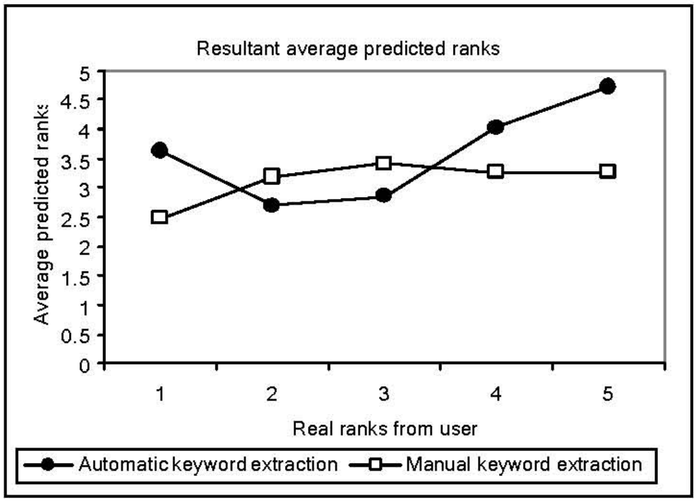

Fig. 7: Average predicted ranks of automated and manually extracted keyword with respect to real ranks

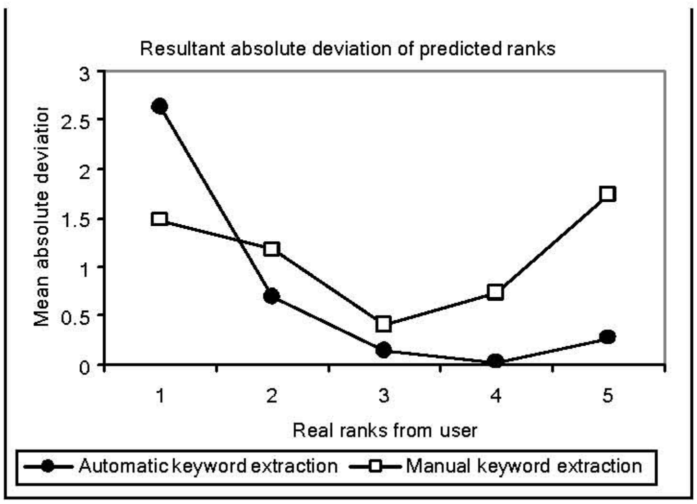

Fig. 8: Absolute deviation of predicted ranks of automated and manually extracted keyword with respect to real ranks

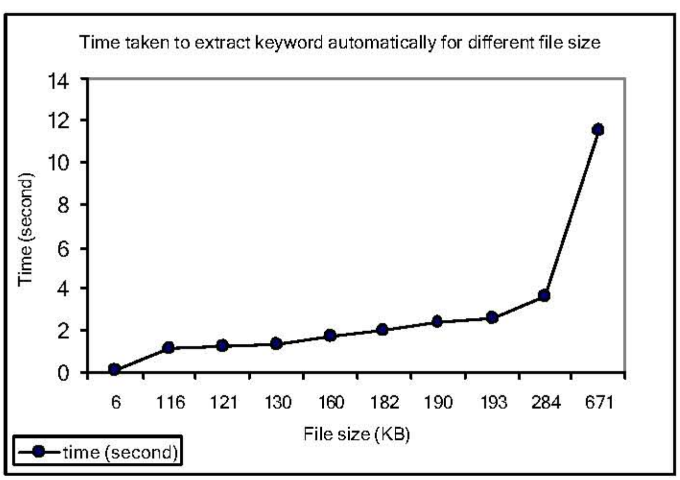

Fig. 9: Time taken for automatic extraction of keyword with respect to different file size

Figure 9 shows the time taken to extract the keywords automatically by the proposed system with respect to different file size. The amount of time requires to extract the keywords automatically tends to increase gradually as the size of the file increases. It is found that the proposed recommender system takes minimum time of about 12.83 seconds (in average) to extract the keywords automatically for file size of about 671 KB.

CONCLUSION

In this paper, we propose a new hybrid recommender system that utilizes automatic keyword extraction technique. Learners are typically drowned with volumes of information and documents and searching the relevant document is time consuming. Recommendation of similar content learning materials with key items, extraction of the keywords automatically from the text-based documents and minimization of the time taken to provide the keywords automatically for every item-all these are crucial yet challenging problems in current e-learning systems. The proposed system addresses all of them. In this study, a combination of CF and CBF is used for item recommendation to facilitate the learners during the learning process and also to enhance the learning performance. The CF will help to improve the accuracy in document searching as it uses some human intelligence as part of the judgment as well as reduce the searching time. However, this paper also devises some new similarity functions, such as user-to-user and doc-to-doc to address the above mentioned problems. More specifically, performance analysis and benchmarking comparison shows that the proposed recommender system with automated keyword extraction can achieve higher accuracy of about 17.04% for MAE and 28.71% for MSE in terms of rating deviation as compared to the manually entered keyword system.

The technique presented in this paper can be useful to create an effective online learning site such as moodle, a Course Management System (CMS), also known as a Learning Management System (LMS). In future, it is of interest to test the proposed framework on real learning environment and to evaluate the learner's performance after using the proposed system to conduct their study.

REFERENCES

1. Liang, G., K. Weining and L. Junzhou, 2006. Courseware Recommendation in E-Learning System. Advances in Web Based Learning-ICWL2006, Springer Berlin/Heidelberg, pp: 10-24.
2. Soonthornphisaj, N., E. Rojsattarat and S. Yim-ngam, 2006. Smart E-Learning Using Recommender System. Computational Intelligence, Springer-Verlag Berlin Heidelberg, pp: 518-523.
3. Tang, T.Y. and G. McCalla, 2003. Smart Recommendation for an Evolving E-Learning System: Architecture and Experiment. Intl. J. E-learning, 4(1): 105-129.
4. Adomavicius, G. and A. Tuzhilin, 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, IEEE Transactions on Knowledge and Data Engineering, 17(6): 734-749.
5. Han, H., C.L. Giles, E. Manavoglu, H. Zha, Z. Zhang and E.A. Fox, 2003. Automatic document metadata extraction using support vector machines. Proceedings of the 2003 Joint Conference on Digital Libraries, pp: 37-48.
6. Shah, P.K., C. Perez-Iratxeta, P. Bork and M.A. Andrade, 2003. Information extraction from full text scientific articles: Where are the keywords?. BMC Bioinformatics, 4(20).
7. Vladislav, K. and S. Miroslav, 2008. Multilingual Approach to e-Learning from a Monolingual Perspective. American Association for Artificial Intelligence.
8. Lemnitzer, L. and P. Monachesi, 2008. Extraction and evaluation of keywords from Learning Objects-a multi-lingual approach. In Proceedings of the Language Re-sources and Evaluation Conference.
9. Lemnitzer, L. and L. Deg´orski, 2006. Language technology for elearning-implementing a keyword extractor. The fourth EDEN Research Workshop Research into online distance education and eLearning Making the Difference.
10. Peter, D.T., 2000. Learning Algorithms for Keyphrase Extraction. Information Retrieval, 2(4): 303-336.
11. Marina, L. and L. Mark, 2008. Graph-Based Keyword Extraction for Single-Document Summarization. Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization, pp: 17-24.
12. Frank, E., G.W. Paynter, I.H. Witten, C. Gutwin and C.G. Nevill-Manning, 1999. Domain-specific keyphrase extraction. Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, pp: 668-673.

13. Wan, X., J. Yang and J. Xiao, 2007. Towards an Iterative Reinforcement Approach for Simultaneous Document Summarization and Keyword Extraction. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp: 552-559.
14. Hunyadi, L., 2001. Keyword Extraction: aims and ways today and tomorrow. In Proceedings of the Keyword Project: Unlocking Content through Computational Linguistics, 2001, pp: 1-6.
15. van Meteren, R. and M. van Someren, 2000. Using content-based filtering for recommendation. In Proceedings of the Machine Learning in the New Information Age. MLnet/ECML2000 Workshop.
16. Badrul, S., K. George, K. Joseph and R. John, 2001. Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th international conference on World Wide Web, pp: 285-295.
17. Roberto, D.T.J., 2004.Combining Collaborative and Content-based Filtering to Recommend Research Papers, PhD thesis, Porto Alegre.
18. Einarsson, O.P., 2007. Content Personalization for Mobile TV Combining Content-Based and Collaborative Filtering, (Master thesis) Technical University of Denmark, DTU.
19. Shengchao, D., Z. Shiwan, Y. Quan, Z. Xiatian, F. Rongyao and B. Lawrence, 2008. Boosting Collaborative Filtering Based on Statistical Prediction Errors. Proceedings of the second ACM International Conference on Recommender Systems, pp: 3-10.
20. Ming, L., M.B. Dias, W. El-Deredy and J.G. Paulo, 2007. A probabilistic model for item-based recommender systems. Proceedings of the 2007 ACM conference on Recommender systems, pp: 129-132.
21. Porter, M.F., 1980. An algorithm for suffix stripping. Program, 14(3): 130-137.
22. Laurent, C., M. Frank and B. Marc, 2007. Comparing state-of-the-art collaborative filtering systems. In Proceedings of 5th International Conference on Machine Learning and Data Mining in Pattern Recognition, pp: 548-562.
23. Resnick, P., N. Lacovou, M. Suchak, P. Bergstrom and J. Riedl, 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work, pp: 175-186.