ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 12 (2022) No. 5, pp. 139-154
DOI:10.33168/JSMS.2022.0509

A Hybrid Ontology-based Recommender System Utilizing Data Enrichment and SVD Approaches

Lit-Jie Chew[1], Su-Cheng Haw[2+], Samini Subramaniam, Kok-Why Ng

[1] Faculty of Computing and Informatics, Multimedia University, 63100 Cyberjaya, Malaysia

[2] AirAsia Berhad, KLIA, Selangor, 64000, Malaysia

sucheng@mmu.edu.my

Abstract. A recommender system is a method of filtering data that provides a personalized recommendation list to a user where the user is interested. The semantic relationship from the ontology modelling does help to boost the accuracy of the recommender system based on recent research. In this paper, we propose a hybrid method to predict the unknown rating in the user-item matrix by using the semantic information of the ontology. The rating prediction utilizes the combination of user-based and item-based techniques. The predicted ratings boost the information of the input data of the model used in the recommender system as input data quality plays an important role in constructing the model. Experimental results demonstrated that the proposed approach achieves greater accuracy as compared to the baseline and existing methods.

Keywords: recommender system, recommender technique, ontology, hybrid recommender, singular value decomposition.

1. Introduction

A Recommender System (RS) recommends items to users based on the users' interest and interaction with the system. Over the past decade, the RS has become one of the trends in research and development. The RS can suggest a user based on the user's interest, behaviour, the similarity between other users and so on. There is a need for RS as the information on the internet keeps increasing nowadays, and there are difficulties for the user to find the exact information they want. The successful use cases in various domains have proven that the RS can help to increase the revenue of the company, especially in the online business domain. eBay (Schafer et al., 1999) and Amazon (Linden et al., 2003) are the first few E-commerce companies that use RS in their system. Online video streaming companies like YouTube (Covington et al., 2016) and Netflix (Gomez-Uribe & Hunt, 2016) have also integrated the RS into their system to promote their video and help users to discover more similar videos.

Ontology is a data modeling method that structures the data in nodes and edges, where the node is the attributes, while the edges are the relationships between each node. By using this modeling method, the semantic information of the data is preserved. The ontology implemented in RS usually helps to reduce the cold start issue and improve the performance of the RS (Middleton et al., 2004).

Recently, most of the research on the model-based recommender system are focused on optimizing the model algorithm. However, data quality that acts as input of the model plays an important role. In this paper, we demonstrate the use of an ontology to increase the model-based recommender system performance.

2. Literature Review

The three main grouping of an RS system are Content-based (CB), Collaborative Filtering (CF) and Hybrid-based (HB).

The CB method generates results by calculating the similarity of the item based on their content such as category. This method needs more information and item to produce a better result. Besides, an overspecialization issue often occurs in this method where it only recommends the same type of item to the user (Isinkaye et al., 2015). According to a survey conducted by (Beel et al., 2016), the CB method appeared in 55% of the RS research papers published from the year 1998 to 2016.

The CF method generates results by considering the interaction between items and users. It suggests the items to users that are liked by other users that have similar interests. Due to the construction method, the traditional CF method is mostly facing cold start problems, data sparsity problems, and scalability issues.

A hybrid method combines two or more techniques in the implementation. It is used to mitigate the issue caused by a single method and combine the advantages of multiple methods. It can be a CB pair with a CF or two CF combinations. A recent

survey in 2020 (Chew et al., 2020) stated that the HB method is the current trend of the research direction. Most of the HB RS have at least one CF in the system.

Two more RS categories fall under the CF method are memory-based and model-based. The memory-based CF suggests the user by calculating the existing relationship between item and user. On the other hand, the model-based CF learns the relationship between user and item to build a model which takes most of the implementation time. The input data quality is the key to ensuring the accuracy of the model-based RS (Heinrich et al., 2019).

The implementation of the ontology in RS has brought a positive effect on the performance of the RS. It has been proven to increase the cold start problem (Almabdy, 2018). With ontology implemented in the system, the fake neighbours' problem has been reduced in the proposed method in (Martín-Vicente et al., 2014). An HB RS has been proposed where it uses ontology to model the data and item-based (IB) CF and user-based (UB) CF have been performed in this method (Gohari & Tarokh, 2016). (Tarus et al., 2017) showed the accuracy improvement by using ontology in an e-learning RS. Bagherifard et al. (Bagherifard et al., 2017) proposed enriching the input data of the model-based CF method. The semantic similarity calculated from the ontology has been used in the CB and CF method. The clustering method has been used to cluster the user in the system. Nilashi et al. (Nilashi et al., 2018) proposed to use ontology RS with clustering method to decrease the overgeneralization issue. On the other hand, (Liu & Li, 2019) proposed a hybrid CF RS that uses ontology as a data modelling method and Singular Value Decomposition (SVD) as the model-based CF RS. Data sparsity issue was minimized by predicting the unknown rating using the IB CF method before the model-based CF process. The summary of the review has been listed in Table 1.

Inspired by (Liu & Li, 2019), we have previously proposed a HB RS that uses ontology modeling to enrich the input data information (Chew et al., 2021). The result shows that the system outperformed the baseline and existing method in the MovieLens 100K dataset. Our proposed method for this paper is the extension work of the proposed method in (Chew et al., 2021).

Table 1: Advantages of the recent ontology-based RS.

Publication	RS Type	Advantages
(Martín-Vicente et al., 2014)	CF	Reduce fake neighborhoods' problem.
(Tarus et al., 2017)	CF	Implicit feedback was considered in this method.
(Bagherifard et al., 2017)	Hybrid (CB and CF)	Clustering was done on user with the used of ontology. Therefore, the computational time was reduced.
(Gohari & Tarokh, 2016) (Nilashi et al., 2018)	Hybrid (IB and UB CF)	(Gohari & Tarokh, 2016) User demographic is used. (Nilashi et al., 2018) Overcome overgeneralization by using clustering method on item and user.

3. Methodology

Inspired by (Liu & Li, 2019), the data enrichment method has been proven and led us to propose the enhanced method in (Chew et al., 2021). The information in the original dataset is enriched by the rating prediction, which is its main contribution. The data enrichment process helps the model-based CF archive better results. In this paper, we focus on enhancing the previously proposed method. We have changed a new dataset which is similar to the MovieLens 100K dataset, but it is larger in scale and the data is sparser. Some part of the algorithm of the process has also been optimized to reduce the processing time.

Fig. 1 illustrates the process flow diagram. There are 4 main parts in the system:

- Retrieving extra Book attributes from Google Books APIs then construct the ontology
- Predict the empty rating by using IB and UB CF
- Predicted rating combines process
- Model-based CF.

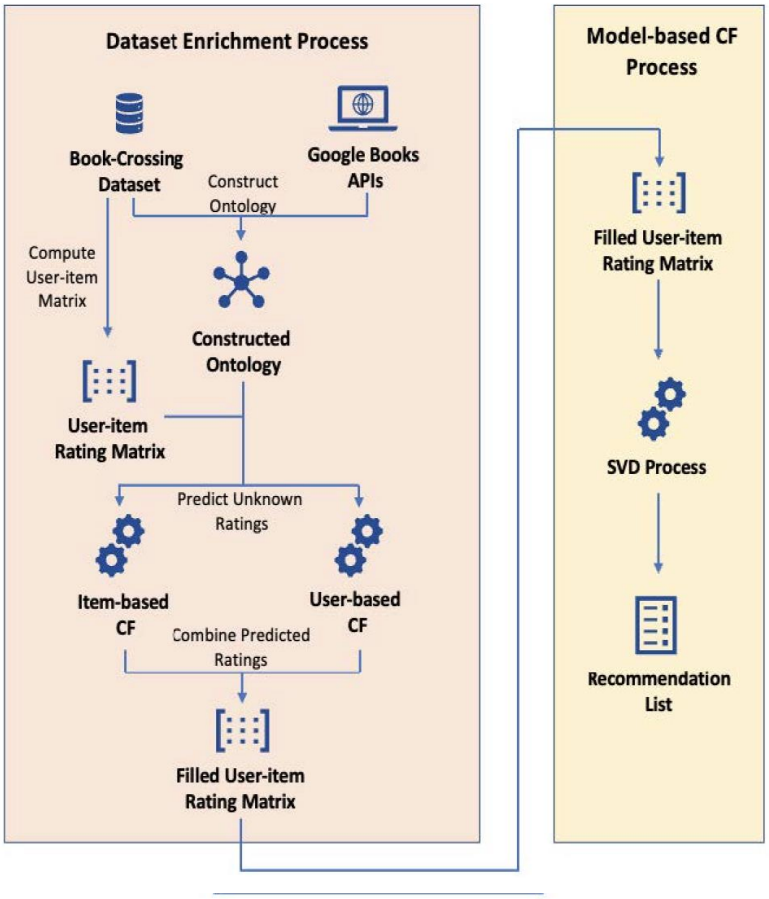

Fig. 1: Process Flow diagram.

3.1. Dataset

The dataset we used is the Book-Crossing Dataset (Ziegler et al., 2005). The dataset includes 278,858 users with demographic information, 271,379 books and a total 1,149,780 ratings.

Upon review of the dataset, some data pre-processing has been done. Some records are removed such as the rating of the user is 0. The dataset is then split into trainset and test set with the ratios of 80% and 20%

The book attribute in the dataset is limited which will limit the performance of the IB CF. To have more details for the book information, we use the Google Book API (*Google Books APIs*, n.d.) to retrieve further information by matching the ISBN. Fig. 2 shows the processed book dataset.

	isbn	book_title	book_author	year_of_publication	publisher	Language	Category
0	0195153448	Classical Mythology	Mark P. O. Morford	2002.0	Oxford University Press	en	Social Science
1	0002005018	Clara Callan	Richard Bruce Wright	2001.0	HarperFlamingo Canada	en	Actresses

Fig. 2: Processed book dataset preview.

3.2. Ontology construction

The ontology has been constructed after the data pre-processing and gaining extra information from other sources (see Fig. 3). The ontology was created based on the relationship of the data attributes then stored in the Neo4j graph database. The total number of nodes and relationships created are represented in Table 2. In the ontology representation, the relationship edges connect all the nodes. The book and user nodes are the main nodes in the system.

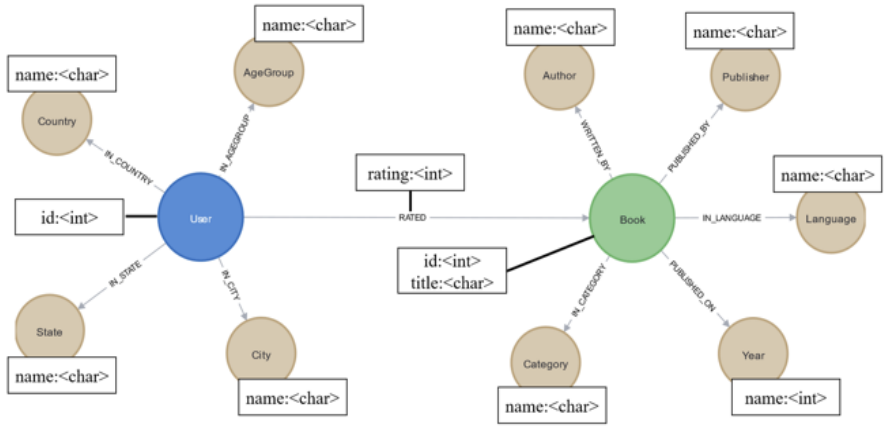

Fig. 3: Ontology constructed based on the book-crossing dataset.

3.3. Unknown rating prediction

With the ontology constructed in the system, we can calculate the semantic similarity by the connection between nodes. In the predicting process, both IB CF and UB CF was included in the process of predicting the unknown rating from rating matrix. Relationships between items are the main consideration in the IB CF. Fig. 4 shows the flow chart for the IB CF process.

The Jaccard similarity index was used in our proposed method to calculate the semantic similarity. It measures the intersection ratio or sets of data. The formula is depicted in Equation (1).

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (1)$$

where J: the Jaccard similarity index, A: set A, B: set B.

Multiple item-item matrices by attributes will be generated where contains the similarity matrix between each item (see Table 3). The weighted average algorithm was used to produce the final item-item similarity matrix. The variables used in the algorithm were decided based on the accuracy evaluation after.

Table 2: Number of nodes and relationships created.

Nodes		Relationships	
Name	Count	Name	Count
[User]	86927	IN_STATE	86927
[City]	14193	IN_COUNTRY	86927
[State]	1912	IN_AGEGROUP	86927
[Country]	360	IN_CITY	86927
[AgeGroup]	9	RATED	173730
[Book]	136226	WRITTEN_BY	136226
[Author]	66684	PUBLISHED_BY	136226
[Year]	12411	IN_LANGUAGE	136226
[Publisher]	11746	IN_CATEGORY	136226
[Language]	3329	PUBLISHED_ON	136226
[Category]	7036	IN_STATE	86927
Total	340833		1202568

Table 3: An example of an item-item similarity matrix.

	Item1	Item2	Item3
Item1	1	0	0.86
Item2		1	0.2
Item3			1

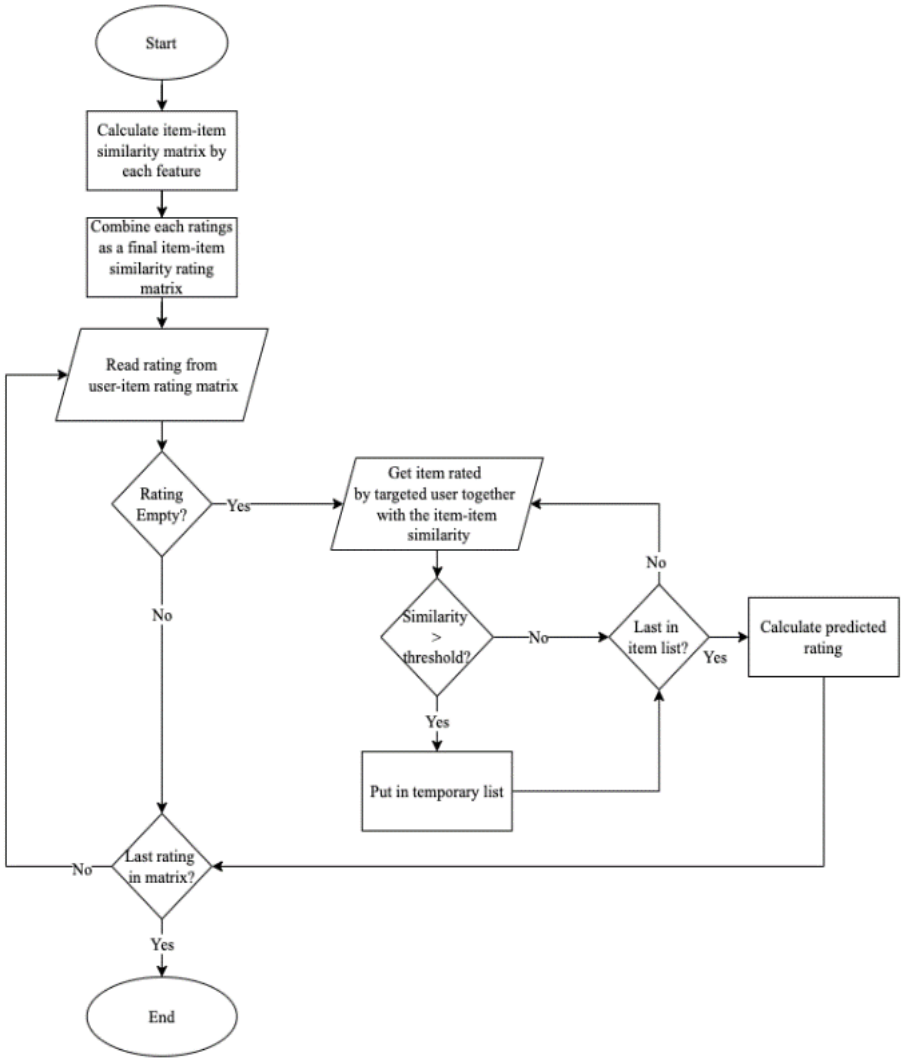

Fig. 4: Flow chart of the IB CF process.

With the final item-item similarity matrices, we can start predicting the unknown rating. In short, this process sums up the rating of the related item which has been rated by the targeted user. Equation (2) shows the formula used.

$$Predicted_{u,m} = \frac{\sum FinalSim_i \times a_{u,i}}{\sum FinalSim_i} \quad (2)$$

where a: rating, u: user, m: item, i: the item rated by the user

During the process, the system will filter the item rated by the targeted user. Then the system will sum up the rating by a weighted average formula where the similarity between the item and the targeted item is the weight. The predicted value will be put in temporary memory and then combine with the rating predicted by UB RS later.

Similar steps from the IB CF above will be applied in the UB CF. The flow chart of the UB CF process is shown in Fig. 5.

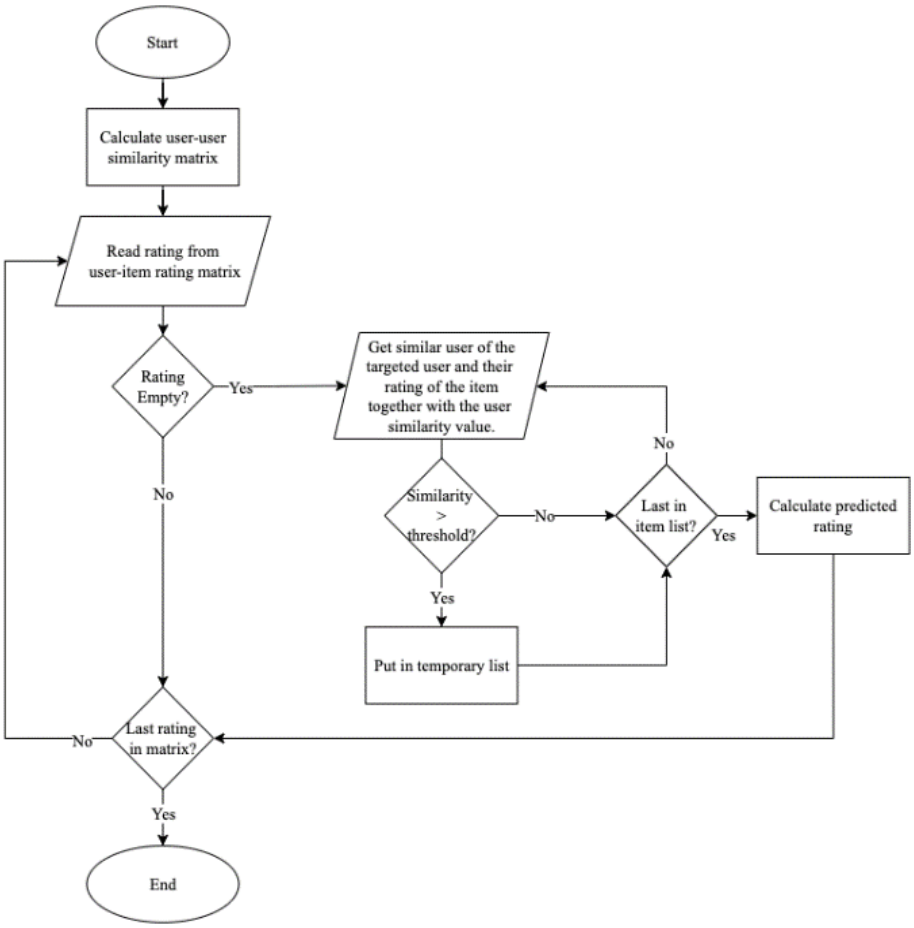

Fig. 5: Flow chart of the UB CF process.

First, the similarity between each user has been calculated. Next, the empty item rating was predicted by filtering the similar users to the targeted user. The steps are similar to the IB CF above. The rating of the similar users to that targeted item was summed and combined by the weight formula. Again, the similarity of the user to the target user is the weight in the formula.

3.4. Combine predicted ratings

In the process of combining the predicted rating, the weighted average formula is applied to produce the final rating. The unknown value of the user-item matrix will be filled with the predicted rating and then combined as an enriched dataset. The dataset will then pass to the model-based CF.

3.5. Model-based CF

The enriched dataset will be used in the model-based CF. SVD was chosen as the model-based CF in our system. SVD is one of the famous model-based CF in RS. It decomposed the matrix into two lower dimensionality matrix and having the abilities to extract the hidden latent features. After the SVD process is done, then the system is ready to recommend items to the user.

3.6. System flow enhancement

As the process flow is shown above, we need to iterate each of the cells in the matrix which make the whole process take an O(n2) process time. This makes the system not able to handle large datasets and has scalability issues.

To solve this issue, we reduce the iteration by getting the related item of each user before applying the IB and UB CF. By doing this, we only iterate the related item based on the related item list while the previous implementation will iterate all item lists. The related item list of the targeted user is extracted from the user-item matrix, where including all similar user's rated item. It executes the similar item filtering by similarity threshold before the rating prediction to avoid spending time predicting the rating of the non-similar item. With this, we can reduce the computational time complexity to O(n) without affecting the result of the prediction.

Besides, multiple thread processing has also been implemented to speed up the process. The matrix will be split according to the predefined number of threads before the iterating process. After the iterating process is done, the matrix will be merged back.

By doing the above changes and some minor code changes, the performance has increased more than 4 times as compared to the previously proposed method. Fig. 6 shows the flow chart of the enhanced process flow.

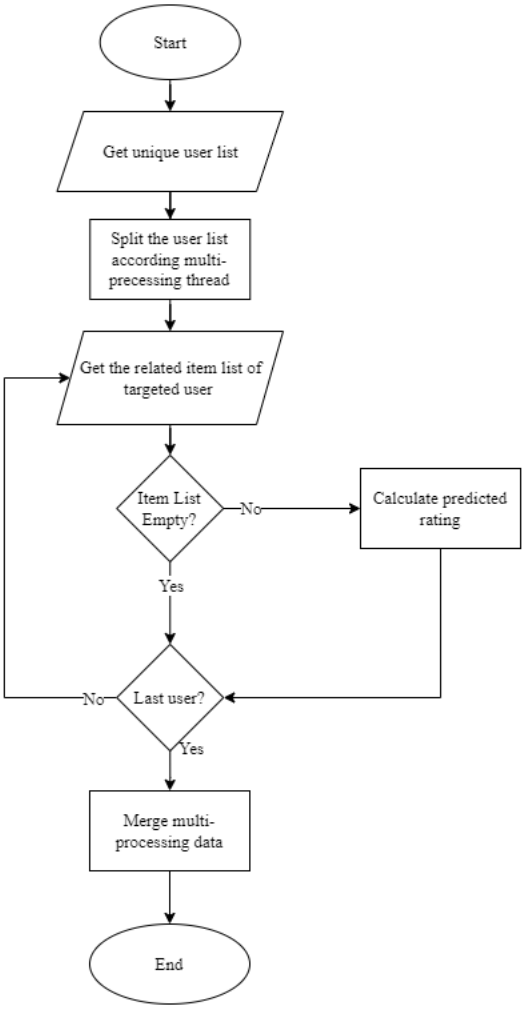

Fig. 6: Flow chart of the enhanced system flow.

4. Evaluation And Discussion

The development and the evaluation environment were done in Ubuntu 18.04, with Jupyter Notebook 6.4.0 and Python 3.9 installed in the system. The data ontology is stored in Neo4j graph database.

For the computational speed evaluation, we compare the computational speed with the previously proposed method (Chew et al., 2021). The dataset used in this evaluation is the Movielens 100K dataset and Book-Crossing Dataset. The average computational time is taken from the average of the 5 repeated computational times. The result is shown in Fig. 7 and Fig. 8.

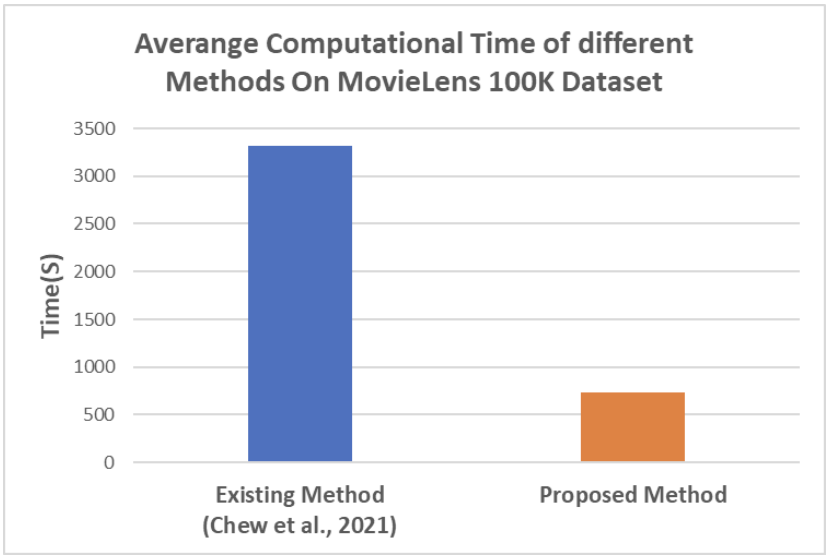

Fig. 7: Average computational time of different methods on MovieLens 100k Dataset.

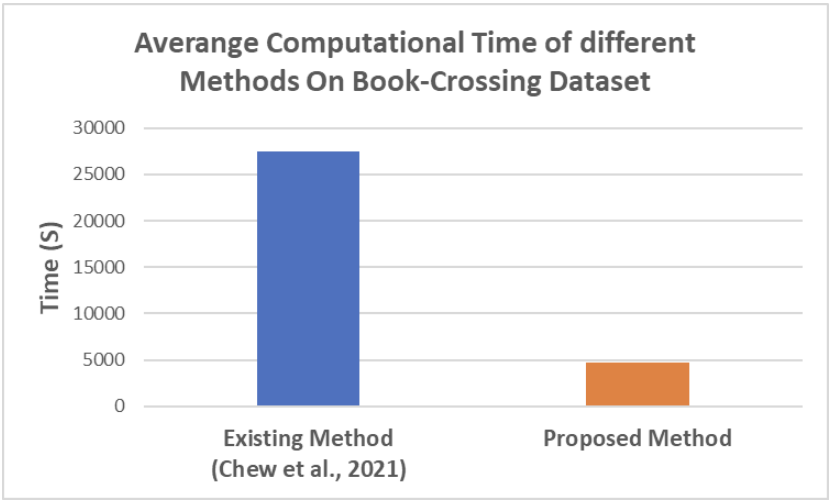

Fig. 8: Average computational time of different methods on book-crossing dataset.

From the Fig. 8 and Fig. 9, we observed that the computational time for our newly proposed method has significantly decreased in both MovieLens 100K and Book-Crossing datasets. The newly implemented iteration method of the user-item matrix has successfully reduced the processing time as it does not require iterating the whole user-item matrix cell.

In the accuracy performance evaluation, we compared the baseline method, the existing method proposed by (Liu & Li, 2019), and our proposed method. The baseline method we compared was only using the SVD method to predict rating,

without the data enrichment process. The existing method proposed by (Liu & Li, 2019) uses only the IB CF to predict the rating in the rating prediction process.

The Root Mean square error (RMSE) formula (see Equation (3)) was used in the evaluation to evaluate the accuracy of the system. It is widely used in the predictive RS evaluation process (Silveira et al., 2019). RMSE calculate error in a way that giving the weight according to the error size. We can consider the system is more accurate if the RMSE value is lower than others.

$$RMSE = \sqrt{\frac{1}{n}\sum_{u,i}(P_{u,i} - r_{u,i})^2} \tag{3}$$

Where n: total number of the training set P: predicted value r: actual value

For the weight used in the weighted average formula when combining the IB CF and UB CF predicted rating, we conducted a test to obtain the best weight to use in the system. The tested weight ratio of UB CF to IB CF varies from 0.3 to 0.7.

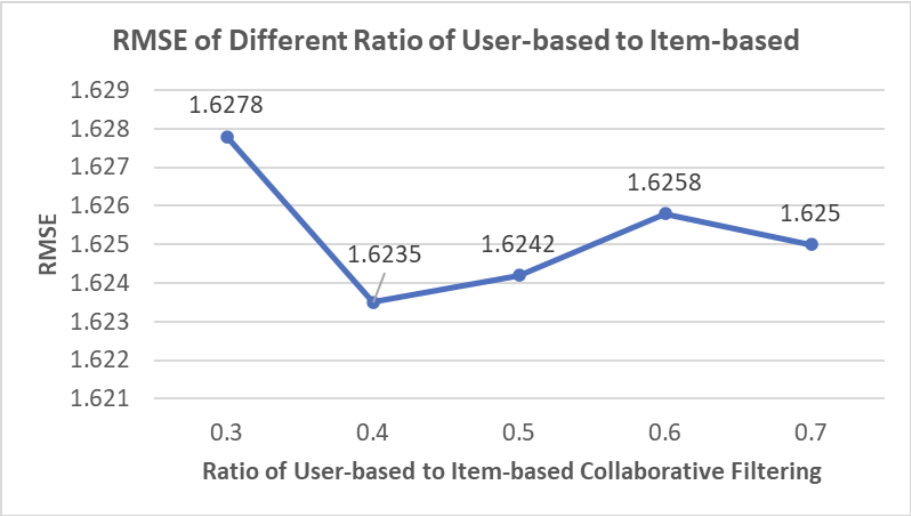

Fig. 9: RMSE of the various ratio of UB CF to IB CF method.

Fig. 9 shows that the RMSE is the lowest, hence accuracy is the highest when the weight ratio of UB CF is 0.4 to the IB CF.

Next, similarity threshold testing was conducted to get the optimized threshold value. The similarity threshold can help to prevent and filter less relevant items and users included in the similar list while running the IB CF and UB CF. The result is illustrated in Fig. 10.

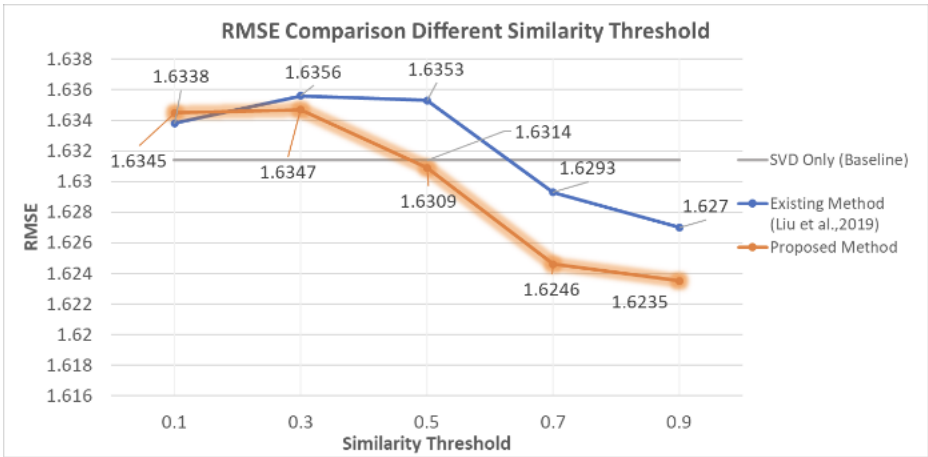

Fig. 10: RMSE comparison of different similarity thresholds and methods.

From the result illustrated in Fig. 10, It shows that our system has the lowest RMSE while the similarity threshold is at 0.9. Some of the results are worsened than using only the baseline model. The possible cause of this situation is the original matrix information has been destroyed by the rating prediction produced by UB and IB CF using less relevant user and item.

From the evaluation above, we observed that the extra information get from Google Books API has increased the accuracy of the system as it contained more information compared to the original dataset. On the other side, our proposed method achieves the same system performance while changing the dataset to a denser and different domain dataset as compared to our previously proposed method (Chew et al., 2021).

5. Conclusion and Future Work

In this paper, we reviewed the recent proposed and developed ontology-based RS. We have also proposed an improvement data enrichment method that is based on our previous works that increasing the model-based CF accuracy by enriching the input data information by IB and UB CF. The newly proposed method increases the computational speed by 5 times without scarifying the accuracy performance of the system. The dataset used in this proposed system is sparser than the previous. In our evaluation results, we can observe the accuracy performance has been maintained as compared to the previously proposed method although our system is running faster. The newly proposed method outperforms as compared to the existing method and baseline method.

Future work includes changing to a level-based semantic similarity calculation in the help of ontology structure can be conducted in the future.

References

Almabdy, S. (2018). Comparative analysis of relational and graph databases for social networks. *1st International Conference on Computer Applications and Information Security, ICCAIS*, 2, 509–512. DOI: https://doi.org/10.1109/CAIS.2018.8441982.

Bagherifard, K., Rahmani, M., Nilashi, M., & Rafe, V. (2017). Performance improvement for recommender systems using ontology. T*elematics and Informatics*, 34(8), 1772–1792. DOI: https://doi.org/10.1016/j.tele.2017.08.008.

Beel, J., Gipp, B., Langer, S., & Breitinger, C. (2016). Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 17(4), 305–338. DOI: https://doi.org/10.1007/s00799-015-0156-0.

Chew, L.-J., Haw, S.-C., & Subramaniam, S. (2020). Recommender system for retail domain. *Proceedings of the 12th International Conference on Computer Modeling and Simulation*, 9–13. DOI: https://doi.org/10.1145/3408066.3408101.

Chew, L.-J., Haw, S.-C., & Subramaniam, S. (2021). A hybrid recommender system based on data enrichment on the ontology modelling. *F1000Research*, 10, 937. DOI: https://doi.org/10.12688/f1000research.73060.1

Covington, P., Adams, J., & Sargin, E. (2016). Deep Neural Networks for YouTube Recommendations. Proceedings of the 10th ACM Conference on Recommender Systems, 191–198. https://doi.org/10.1145/2959100.2959190

Gohari, F. S., & Tarokh, M. J. (2016). A New Hybrid Collaborative Recommender Using Semantic Web Technology and Demographic data. *International Journal of Information and Communication Technology Research*, 8(2), 51–61. DOI: http://journal.itrc.ac.ir/article-1-70-en.html.

Gomez-Uribe, C. A., & Hunt, N. (2016). The netflix recommender system. *ACM Transactions on Management Information Systems*, 6(4), 1–19. DOI: https://doi.org/10.1145/2843948.

Google Books APIs. (n.d.). https://developers.google.com/books.

Heinrich, B., Hopf, M., Lohninger, D., Schiller, A., & Szubartowicz, M. (2019). Data quality in recommender systems: The impact of completeness of item content data on prediction accuracy of recommender systems. *Electronic Markets*. DOI: https://doi.org/10.1007/s12525-019-00366-7.

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3), 261–273. DOI: https://doi.org/10.1016/j.eij.2015.06.005.

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1), 76–80. DOI:

https://doi.org/10.1109/MIC.2003.1167344.

Liu, W., & Li, Q. (2019). Collaborative filtering recommender algorithm based on ontology and singular value decomposition. Proceedings - *2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC*, 2, 134–137. DOI: https://doi.org/10.1109/IHMSC.2019.10127.

Martín-Vicente, M. I., Gil-Solla, A., Ramos-Cabrer, M., Pazos-Arias, J. J., Blanco-Fernández, Y., & López-Nores, M. (2014). A semantic approach to improve neighborhood formation in collaborative recommender systems. *Expert Systems with Applications*, 41(17), 7776–7788. DOI: https://doi.org/10.1016/j.eswa.2014.06.038.

Middleton, S. E., de Roure, D., & Shadbolt, N. R. (2004). Ontology-based recommender systems. In Handbook on Ontologies, 477–498. Springer Berlin Heidelberg. DOI: https://doi.org/10.1007/978-3-540-24750-0_24.

Nilashi, M., Ibrahim, O., & Bagherifard, K. (2018). A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Systems with Applications*, 92, 507–520. DOI: https://doi.org/10.1016/j.eswa.2017.09.058.

Schafer, J. ben, Konstan, J., & Riedi, J. (1999). Recommender systems in e-commerce. *Proceedings of the 1st ACM Conference on Electronic Commerce - EC '99*, 158–166. DOI: https://doi.org/10.1145/336992.337035.

Silveira, T., Zhang, M., Lin, X., Liu, Y., & Ma, S. (2019). How good your recommender system is? A survey on evaluations in recommendation. *International Journal of Machine Learning and Cybernetics*, 10(5), 813–831. *https://doi.org/10.1007/s13042-017-0762-9.*

Tarus, J., Niu, Z., & Khadidja, B. (2017). E-learning recommender system based on collaborative filtering and ontology. *International Journal of Computer and Information Engineering*, 11(2), 400–405.

Ziegler, C.-N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005). Improving recommendation lists through topic diversification. *Proceedings of the 14th International Conference on World Wide Web - WWW '05, 22*. DOI: https://doi.org/10.1145/1060745.1060754.