

Chatbot-Based Book Recommender System Using Singular Value Decomposition

Muhammad Attalariq, Z K A Baizal*

School of Computing, Informatics, Telkom University, Bandung

Jl. Telekomunikasi. 1, Terusan Buahbatu - Bojongsoang, Telkom University, Sukapura, Kec. Dayeuhkolot, Kabupaten Bandung, Jawa Barat, Indonesia

Email: ¹attalariqtelu@student.telkomuniversity.ac.id, ^{2,*}baizal@telkomuniversity.ac.id

Correspondence Author Email: baizal@telkomuniversity.ac.id

Submitted: 10/07/2023; Accepted: 28/07/2023; Published: 31/07/2023

Abstract—In the era of information overload, finding the right book that matches one's preferences and interests has become a challenging task for users as many online book provider service websites such as Amazon, Goodreads, and Gramedia provide books of various types and choices. Recommender systems can be used in addressing such issues, it works by filtering information that provides predictions and suggests the best product or service to the user. Currently, various book recommender systems have been developed, but the systems do not provide interaction between the user and the system. Therefore, we propose a recommender system built with a conversational approach so that it can interact with natural language. Recommender system built using matrix factorization method with Singular Value Decomposition (SVD) algorithm, SVD is proven to have advantages for handling large datasets, extracting features, reducing noise and dimensionality so as to speed up computation. We performed two types of evaluation on the system. First, we tested the prediction accuracy using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) metrics. Second, we use questionnaires to measure user satisfaction levels. The evaluation of the system shows that the results of the prediction accuracy obtain an MAE value of 0.6481 and an RMSE value of 0.8287. Then, the accuracy performance of the system found that 83.2% of users get recommendations according to their interests. The user satisfaction with the whole system is 87.9%. The system built can provide a fairly good recommendation performance, and the chatbot can interact well with users based on the evaluation results obtained.

Keywords: Book Recommender System; Chatbot; Collaborative Filtering; Singular Value Decomposition; Matrix Factorization

1. INTRODUCTION

Today's rapid technological development is followed by an increase in the number of information circulating on the Internet. With the amount of information circulating too abundantly, finding the right information quickly and getting satisfactory results becomes increasingly difficult for users [1]. This also happens when searching for information such as books [2], various online book provider service websites such as Amazon, Goodreads, Gramedia, and others provide books with different types and choices. As a result, book lovers often have difficulty in determining the choice of other books to read. To overcome these problems, a system is needed that can help users find and determine book choices precisely, quickly, and of course with a system that is easy to operate.

Recommender system is a system that aims to provide the best suggestions for products and services to its users. This system provides results of items or products that are relevant to the user's interests and needs by finding patterns in the information set by learning the user's choices and behavior [3]. In building a recommender system, various methods such as hybrid filtering, collaborative filtering, and content-based filtering can be used [4]. Recommender systems have proven to be a successful solution to the problem of information overload in recent years, especially for people who do not have sufficient experience to evaluate the potential of a large number of alternative items in the services offered [5].

The recommender system built in this research uses a model-based Collaborative Filtering (CF) method with Singular Value Decomposition (SVD) technique and cosine similarity calculation to measure item similarity. SVD is a matrix factorization technique for identifying latent semantic factors in information retrieval. In its application, the factorization of the user-item evaluation matrix is required, in addition, SVD also has advantages for dealing with large dataset, extracting features, reducing noise and dimensional space in large data to speed up computation [6].

So far, there is research related to the development of recommender systems using model-based CF techniques. Sireesha, et al [7] built recommendations in the book domain using model-based CF and K-nearest neighbor to classify items. The similarity measure used in this study is the cosine distance, to obtain a set of items that are similar to the target item. The research by Christina, and Baizal [8] also discusses the development of a model-based CF recommender system using SVD technique enhanced with the Slope One algorithm. This combination addresses the problem of data sparsity, because the model is trained with more complete data as the Slope One algorithm fills in the empty rating data. Based on the test results with the MAE error metric, it shows that the recommender system built with the combination method of SVD and Slope One algorithm is better than using only one of the algorithms from this method. Pujahari, et al [9] analyzed various CF model-based recommender system techniques factorization approach in the movie domain, it was found that the SVD approach had the best training time. In this case, the computation time performance of the recommender system depends on the dimension of the user-item latent factor matrix.

In the various book recommender systems that have been developed [7,8], or that use SVD techniques in other domains such as skincare [10], e-commerce products [11], and music [12], the system does not provide intensive interaction between the system and the users. This limits users' flexibility in searching for books that suit their interests. Therefore, we propose a book recommender system that can interact in natural language through a chatbot. In the built system, books are recommended based on the preferences of other books favored by the user. One method of obtaining such information is to have a conversation with the user through a Conversational Recommender System (CRS) [13]. This system allows a more direct interaction with the user to get the needed information, similar to how users have daily conversations [14].

Our research also relates to several other studies on the topic of conversation-based recommender systems, as done by Theosaksomo, et al. [15] built a chatbot conversation recommender system that provides recommendations based on users' functional needs. The results of usability tests conducted on the chatbot get good results, the main aspects evaluated were the user experience of getting recommendations as in daily life and the ease of adding functional requirements. The development of CRS has also been carried out in the music domain by Narducci, et al. [16], a system built using content-based recommendations, provide explanatory facilities, implement critiques and adaptive strategies. Users interact with the system using different methods such as natural language, buttons, or a combination of both. The evaluation results showed that users prefer interaction modes that combine buttons and natural language. In research by, Jeffrey Dalton, et al. [17] built a chatbot-based recommender system in the movie domain using Dialogflow. The recommender system is built with a collaborative filtering method, then this chatbot can receive voice input in the interaction process. The movie recommendation results produced by this system are very accurate. Fajari and Baizal [18] also built a CRS to recommend culinary tours. Built with the Named Entity Recognition method which functions to recognize or retrieve entities such as preferences, names, ages in user input. TF-IDF and cosine similarity techniques are used to generate recommendation items.

In this research, we focus on building a chatbot using the Dialogflow framework with natural language processing that can receive user preference information and provide recommendations based on it. This chatbot-based book recommender system is built on the Telegram platform. Then, the prediction accuracy performance of the system is evaluated using RMSE and MAE. Then, to measure the level of user satisfaction is evaluated using a questionnaire. With this system, it is expected to facilitate users in finding books according to their wishes.

2. RESEARCH METHODOLOGY

Baizal, et al. [19] developed a CRS in the movie domain using a natural language processing technique. The built system's recommendation mechanism utilizes content-based filtering. The Dialogflow framework, which is implemented on the Telegram platform, is used for chatbot development. Based on this research, we used the idea of these components that have been redesigned and adapted to the domain of the book.

2.1 Research Stages

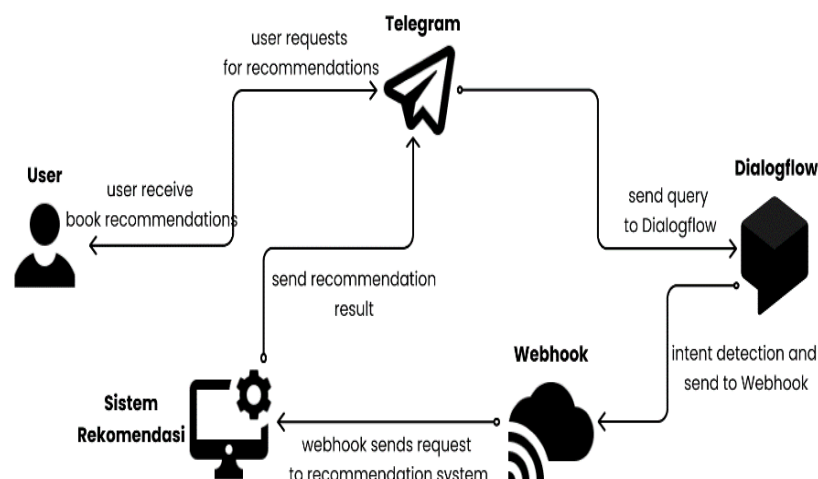


Figure 1. Systems overview

In Figure 1, an overview of the system architecture is shown. In the first stage, intent detection is carried out from queries that have been given by users with a natural language processing approach using the Dialogflow framework. The second stage is forwarding queries containing user preferences to the recommender system via webhooks. The next stage is to generate book recommendations based on the book title preferences. The final stage is to provide recommendation results to users via the Telegram platform.

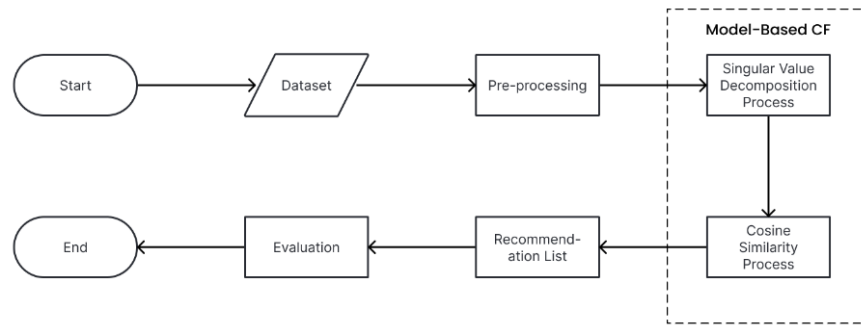


Figure 2. Recommender system flowchart

Recommender systems aim to provide accurate and personalized information based on user needs, done by filtering large amounts of information using various methods and algorithms [20]. In Figure 2, the stages of building a recommender system are shown. The first phase is dataset collection, then dataset preprocessing is the second phase. The third phase calculates singular value decomposition on the preprocessed dataset. The next stage calculates cosine similarity to produce a number of recommendation items. The recommendation results are then evaluated using error metrics. A questionnaire is also used to measure user satisfaction with the system as a whole.

2.2 Dialogflow Components

Dialogflow is a natural language understanding-based platform developed by Google. The platform allows developers to build conversational interfaces such as chatbots [21]. In its working process, Dialogflow requires two main component factors [19], namely agents and intents as shown in Figure 3. The agent here is a virtual agent that handles and processes conversations with users. Meanwhile, intents are useful for categorizing conversations. Intents represent the mapping between what the user says and what action the system needs to take. A set of example phrases known as "training phrases" are used to train each intent. These phrases show various ways a user might express the same intention. We increase the intent's ability to accurately match user input by offering a variety of training phrases.

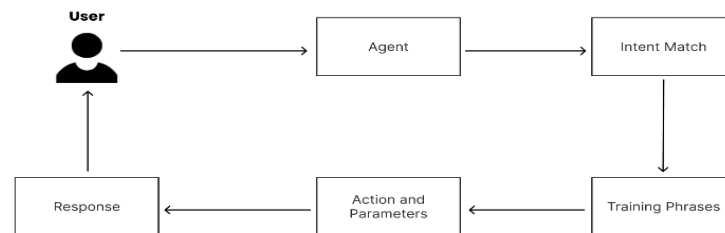


Figure 3. Intent matching flow

2.3 Dataset

The data used in this study is the Goodbooks-10k dataset from Kaggle, which is data on book product reviews on Goodreads e-commerce. The dataset consists of two csv format files namely books, and ratings. The books file has 10000 rows and 23 columns, then the ratings file has 981756 rows and 3 columns.

2.4 Preprocessing Data

Preprocessing is the process of transforming raw data, which usually has an incomplete and irregular data structure, into a form that is easy to analyze. Preprocessing in this study includes deleting unused columns, deleting data rows with empty values, and filtering data based on criteria. To select valid book data, we chose books that were rated by at least 30 users. This is done to avoid book titles that are very unfamiliar to the users. Then, the data attributes needed for this research are user ID, book ID, and book rating. After these various stages, the final number of rating data becomes 981713 rows with 9961 unique book titles, and 53031 unique users. In Table 1 shows sample data results after the preprocessing stage.

Table 1. Dataset sample after preprocessing

User ID	Book ID	Rating
21684	315	4
30112	1120	5
8089	46	4

User ID	Book ID	Rating
4553	2668	4
7065	421	3

2.5 Singular Value Decomposition

SVD is a technique that can be used in developing model-based CF recommender systems. This technique is one of the approaches in the Matrix Factorization method. In this technique, we have a set of users, items, and user rating, usually represented by a user-item rating matrix. Based on this, the SVD algorithm derives the latent factor and makes recommendations based on the user-item matrix.

$$A = U \cdot S \cdot V^T \quad (1)$$

It can be seen in that the matrix decomposes into three other matrices as follows: A is an $m \times n$ utility matrix, U is an $m \times m$ orthogonal matrix, S is an $r \times r$ diagonal matrix, and V^T is an $r \times n$ matrix resulting from the transposition of orthogonal matrix. Thus, matrix U is representing users with latent factors, matrix V^T is representing items with latent factors, while the diagonal matrix S is the singular value.

In this study, we use the right singular vector to perform item-based collaborative filtering. Item-based CF with SVD uses item vectors obtained from SVD to compute item similarity and generate recommendations based on user-item interaction. This approach is efficient for handling large datasets and can provide accurate recommendations by exploiting the latent features captured by SVD. The matrix used has a number of k latent factors for each item, as shown in Table 2.

Table 2. Right-singular matrix with k latent factors

Book ID	0	1	2	...	k
0	-0.046799	-0.037289	-0.037823	...	0.035537
1	0.037493	-0.175064	0.097511	...	-0.015152
2	-0.030098	-0.050828	0.017948	...	0.027037
...
9959	0.039308	0.034704	-0.070687	...	0.014746
9960	-0.027186	-0.028970	-0.076146	...	0.008337

2.6 Cosine Similarity

Cosine similarity is a metric for measuring the degree of similarity between two vectors, usually in a high-dimensional space [22]. Cosine similarity determines the cosine of the angle among two items or products vector obtained from dot product of the two vectors, then divided by magnitudes of the two vectors. After that, it will produce a similarity score between 0 and 1, where the value 1 indicates perfect similarity, and 0 indicates no similarity. The cosine similarity calculation is solved

$$\text{Similarity}(p, q) = \cos \theta = \frac{p \cdot q}{\|p\| \cdot \|q\|} \quad (2)$$

After obtaining the item-latent factor matrix described in Section 2.5 Singular Value Decomposition, then calculate the cosine similarity between each items using feature vectors. Finally, the items with the highest cosine similarity scores are identified as similar items to the target item. These similar items are the "neighbors" of the target item. In this case, the target item is the book item preferred by the user.

2.7 Evaluation Metrics

To evaluate the recommender system in this study, we used error metrics to measure the accuracy of system predictions with MAE and RMSE calculations.

a. Mean Absolute Error

MAE is one of the popular and well-established error metrics for evaluating the accuracy performance of recommender systems. MAE measures the difference of the average absolute deviation between the rating predicted by the recommender system and observed values, in this case is user's actual rating [23]. The accuracy of the recommender system is good if it gets a relatively low MAE value. MAE calculates the average error value by giving equal weight to all data. The calculation of MAE is shown in equation

$$MAE = \left(\frac{1}{n} \right) \sum_{i=1}^n |p_{u,i} - r_{u,i}| \quad (3)$$

Where $p_{u,i}$ is the rating value predicted by the recommender system and $r_{u,i}$ is the actual rating, n is number of data, u and i represent users and items.

b. Root Mean Squared Error

RMSE is a metric commonly used for evaluate the accuracy or performance of a predictive model. RMSE calculates average squared difference among user's actual and predicted rating. In the calculation, if the RMSE

value gets a lower value it shows better accuracy and closer similarity between prediction and actual values. The RMSE formula is shown in equation

$$RMSE = \sqrt{\left(\frac{1}{n}\right) \sum_{i=1}^n (p_{u,i} - r_{u,i})^2} \quad (4)$$

Where n is the number of data, then $p_{u,i}$ is the predicted rating by the recommendation's algorithm, and the actual rating value on item is denoted by $r_{u,i}$, u and i represent users and items.

2.8 User-System Conversation Flow

In Figure 4 shows the conversation flow between user and chatbot system. First, the user asks the system for a book recommendation. At this stage, there are two possibilities: 1) the user does not have a preferred book title, or 2) the user does have a preferred book title. In the first case, the system provides several popular book titles. The user can select one of these titles as a preferred book. For the second, the system asks for the user's preferred title and searches the database. If available, the system will immediately provide a set of book recommendations consisting of book titles and covers. If the user is not satisfied with the recommendation results, the user can repeat the process and try with other preferred book titles. If the user responds favorably to the recommendation result, the system ends the interaction with the user.

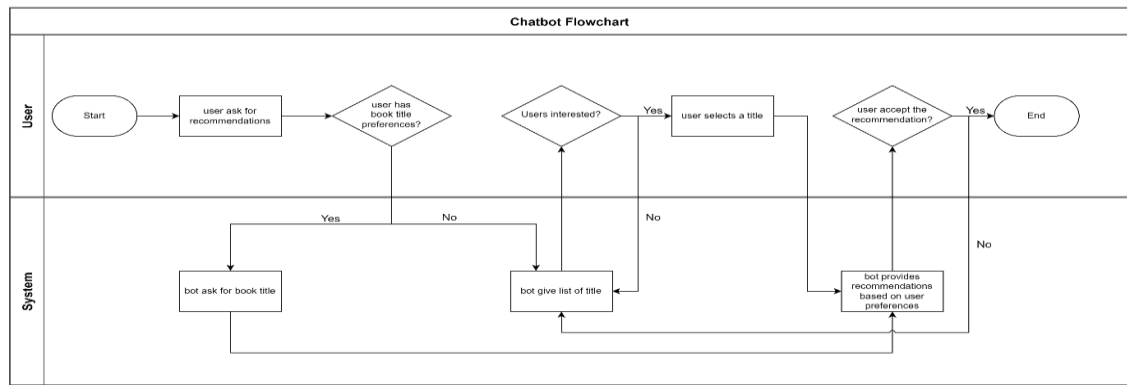


Figure 4. User-system conversation flowchart

3. RESULT AND DISCUSSION

In the performance evaluation of the CRS proposed, a combination of evaluation metrics is used to obtain more comprehensive results. To evaluate the predicted ratings, we use existing metrics such as MAE and RMSE to measure the accuracy of the ratings predicted by the recommender system compared to the actual ratings by the users. In addition, we included a questionnaire to obtain user feedback after they interacted with the system. By asking several questions to measure the level of user satisfaction, accuracy of recommendations, and additional suggestions for future development of the system. By combining quantitative and qualitative evaluation approaches, we aim to obtain a comprehensive evaluation of the system's performance.

3.1 Systems Performance

In evaluating the SVD algorithm for rating prediction, we run tests using k-fold cross validation with $k = 5$, epochs = 20, with a total of 981713 data. Then, the dataset was separated into 20.0% for the test data and 80.0% for the training data. Evaluation is done using MAE and RMSE metrics, the smaller the value, it indicates good predicting accuracy. Testing results of this prediction accuracy evaluation are detailed in Table 3.

Table 3. Rating prediction performance results on the SVD algorithm using various n-latent factors

Metric	n-factors	k-fold					Average
		1	2	3	4	5	
MAE	10	0.6516	0.6513	0.6540	0.6511	0.6533	0.6523
	25	0.6512	0.6521	0.6502	0.6516	0.6491	0.6509
	50	0.6497	0.6506	0.6481	0.6489	0.6498	0.6494
RMSE	10	0.8328	0.8339	0.8375	0.8337	0.8362	0.8348
	25	0.8331	0.8349	0.8322	0.8341	0.8317	0.8332
	50	0.8320	0.8337	0.8287	0.8295	0.8324	0.8313

In SVD algorithm, "n-factors" refers to the number of dimensions or factors retained during the decomposition process. The degree of approximation or dimensionality reduction achieved by SVD is determined

by the number of n-factors chosen. Furthermore, k-fold indicates the number of iterations performed on each metric. Finally, average indicates the average of all iterations. As can be seen in Table 3, from different numbers of n-factors and k-folds, the SVD algorithm produces the best MAE value at n-factor = 50 at k-fold = 3 iterations with MAE = 0.6481. While the best RMSE value is obtained at n-factor = 50 and k-fold = 3 iterations with RMSE = 0.8287. However, if we look at the best average with MAE = 0.6494 and RMSE = 0.8313, we see that the SVD algorithm is optimal at n-factor = 50. These results show that the SVD algorithm used to recommend books in this research can predict ratings well based on low MAE and RMSE values.

3.2 User Satisfaction

In this test, we used a questionnaire to get feedback and measure the level of user satisfaction. The questionnaire was administered online to 25 respondents, whose age ranged from 20 to 25, most of whom were students. We chose this age range because at this age, users are considered to be familiar with digital environments such as chatbots and are still enthusiastic about reading books. There are 7 statements in the questionnaire, grouped into 6 factors, namely informative (INF), easy to use (ETU), perceived recommendation quality (PRQ), ease of understanding (EOU), trust (TR), and perceived efficiency (PE) [24]. the calculation to get the final score for each statement in the questionnaire is shown.

$$\text{Final Score} = \frac{\text{Total score}}{\text{Maximum score}} \times 100 \quad (5)$$

Table 4 shows a number of statements given to users with each factor. Each statement is then given a rating based on a predetermined weight, as shown in Table 5.

Table 4. Statements of the questionnaire

ID	Factor	Information
P1	ETU	You can easily navigate and interact with the chatbot system
P2	ETU	The instructions and guidance provided by the system are clear and intuitive
P3	EOU	Easy for you to understand the recommendations provided by the system
P4	PE	You are satisfied with the speed and responsiveness of the system
P5	PRQ	Recommended books suites to your preferences and interests
P6	INF	The system provides sufficient information about each recommended book
P7	TR	You will trust the recommendations from the system for future book selection

Table 5. Questionnaire assessment weights

Weight	Description	Score
DS	Disagree	1
SD	Somewhat Disagree	2
NT	Neutral	3
SA	Somewhat Agree	4
ST	Strongly Agree	5

In Table 6, displays the results of testing with questionnaires that have been carried out, showing that the proposed chatbot-based recommender system can provide satisfactory results for users, with a final user satisfaction level of 87.9%. In statements P1 and P2 with the ETU factor getting positive results from users, indicating that the flow of interactions and instructions contained in the chatbot system can be understood by users well. Users are also satisfied with the speed and responsiveness of chatbots that respond to user requests in a short time. Then, the accuracy of the book recommendations received a score of 83.2% based on the score on statement P5. Therefore, with these performance results we find that the proposed system can provide fairly accurate recommendations to users. However, some users still have problems when they find that their preferred book titles cannot be recognized by the system due to problems with the availability of book data in the system. This will affect the performance of the system in providing recommendations to users. In addition, problems also occur when users do not have or know the preferred book title. Based on additional suggestions given by users in the questionnaire, it is recommended that the system asks for the genre or author's name if the user does not have a specific title preference, in this way the user feels more flexible and can improve the performance of the recommender system.

Table 6. User satisfaction questionnaire results

ID	Assessment results					Final Score
	DS	SD	NT	SA	ST	
P1			9	44	55	(108/123) * 100 = 86,4%
P2				36	80	(116/125) * 100 = 92,8%
P3			3	36	75	(114/125) * 100 = 91,2%
P4			9	44	55	(108/125) * 100 = 86,4%

ID	Assessment results					Final Score
	DS	SD	NT	SA	ST	
P5			12	52	40	$(104/125) * 100 = 83,2\%$
P6			6	40	65	$(111/125) * 100 = 88,8\%$
P7			6	52	50	$(108/125) * 100 = 86,4\%$
Final Result						87,9%

3.3 Application Implementation

A chatbot is a computer program or artificial intelligence (AI) model designed to simulate human conversation through text or voice interaction. A chatbot can interpret and understand user input and provide appropriate responses in a conversational manner [14]. Many applications such as customer support, interactive messaging platforms, virtual assistants, and information retrieval are built using chatbots [21]. In this research, a book recommender system is implemented on the Telegram platform using conversational interaction through a chatbot built with Dialogflow with a natural language processing approach.

In Figure 5, the chatbot flow is shown if the user has a preferred book title that will be used as the target item in the recommendation process. The chatbot system provides book recommendations in the form of a list consisting of book titles and cover images. Then, Figure 6 displays the flow if the user does not have a preference item. In that scenario, the chatbot will provide a list of books to be selected by the user. The selected book will become the user's preference item, and then the chatbot will provide recommendations based on that item.

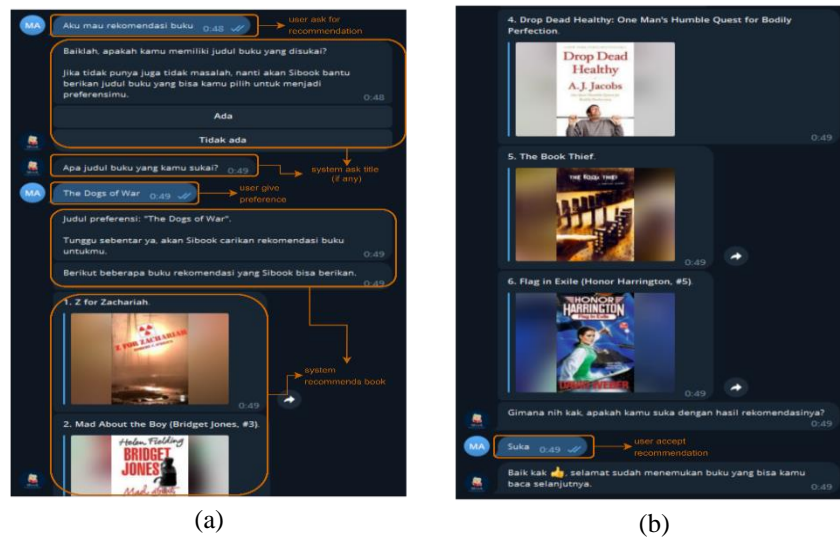


Figure 5. Chatbot flow if the user has a preference item

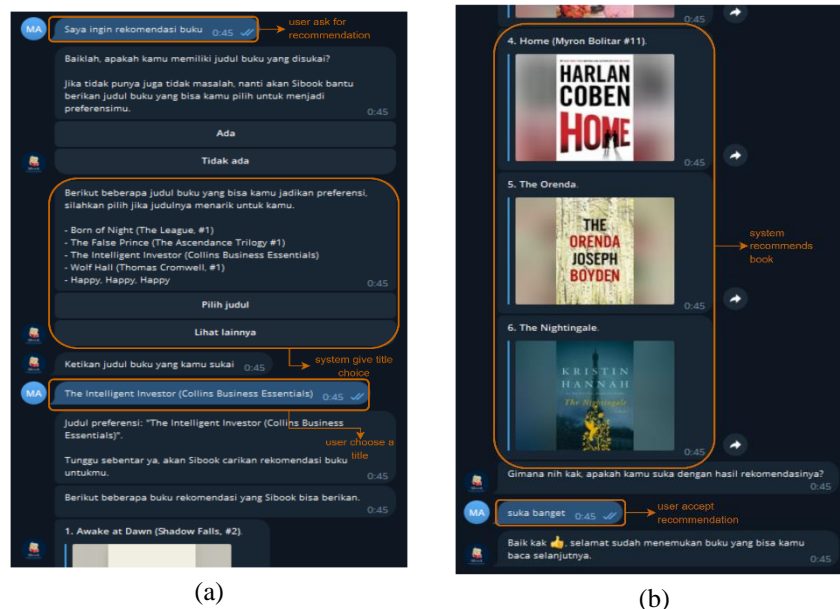


Figure 6. Chatbot flow if user has no preference item



4. CONCLUSION

In this research, we successfully developed a conversational recommender system through a chatbot in the book domain. The recommender system built using the SVD algorithm can provide quite good performance. Based on the evaluation of the rating prediction test with error metrics, and the measurement of user satisfaction through questionnaires. The evaluation results we get from testing the prediction accuracy performance of the recommender system and user satisfaction found that the book recommender system built has an RMSE of 0.8287 and MAE of 0.6481. These results show that the recommender system can provide good prediction accuracy. Then, based on the user satisfaction questionnaire, the accuracy of the system is 83.2%. The questionnaire results show that users give a positive response to the system based on the six factors asked: informative (INF), easy to use (ETU), perceived recommendation quality (PRQ), ease of understanding (EOU), trust (TR), and perceived efficiency (PE) with a satisfaction level of 87.9%. The results show that the conversation-based recommender system built, successfully carries out interactions with users effectively and has good accuracy in providing book recommendations to users. The drawback of this research is that the system can only accept book title preferences in providing recommendations. Thus, we suggest that further research can add other functional requirements such as genre or author name, and use the improved singular value decomposition or other matrix factorization algorithm to see a wider comparison of accuracy prediction results.

REFERENCES

- [1] Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R. Kashef, "Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities," *Applied Sciences*, vol. 10, no. 21, p. 7748, Nov. 2020, doi: 10.3390/app10217748.
- [2] H. Alharthi, D. Inkpen, and S. Szpakowicz, "A survey of book recommender systems," *J Intell Inf Syst*, vol. 51, no. 1, pp. 139–160, Aug. 2018, doi: 10.1007/s10844-017-0489-9.
- [3] K. Falk, *Practical recommender systems*. Shelter Island, NY: Manning, 2019.
- [4] P. G. Padti, K. Hegde, and P. Kumar, "Hybrid Movie Recommender System," *International Journal of Research in Engineering, Science and Management*, vol. 4, no. 7, pp. 311–314, Jul. 2021.
- [5] F. Ricci, L. Rokach, and B. Shapira, Eds., *Recommender Systems Handbook*. New York, NY: Springer US, 2022. doi: 10.1007/978-1-0716-2197-4.
- [6] N. Bhalse and R. Thakur, "WITHDRAWN: Algorithm for movie recommendation system using collaborative filtering," *Materials Today: Proceedings*, p. S2214785321003242, Feb. 2021, doi: 10.1016/j.matpr.2021.01.235.
- [7] V. Sireesha, N. P. Hegde, K. Sreenija, and B. Thindhu, "An Enhanced Book Recommendation System Using Hybrid Machine Learning Techniques," in *Smart Intelligent Computing and Applications, Volume 2*, S. C. Satapathy, V. Bhateja, M. N. Favorskaya, and T. Adilakshmi, Eds., in *Smart Innovation, Systems and Technologies*, vol. 283. Singapore: Springer Nature Singapore, 2022, pp. 171–179. doi: 10.1007/978-981-16-9705-0_17.
- [8] Christina and Z. K. A. Baizal, "Book Recommender System Using Singular Value Decomposition Combined with Slope One Algorithm," in *2022 10th International Conference on Information and Communication Technology (ICoICT)*, Bandung, Indonesia: IEEE, Aug. 2022, pp. 346–350. doi: 10.1109/ICoICT55009.2022.9914884.
- [9] A. Pujahari and D. S. Sisodia, "Model-Based Collaborative Filtering for Recommender Systems: An Empirical Survey," in *2020 First International Conference on Power, Control and Computing Technologies (ICPC2T)*, Raipur, India: IEEE, Jan. 2020, pp. 443–447. doi: 10.1109/ICPC2T48082.2020.9071454.
- [10] F. Nissa, A. H. Primandari, and A. K. Thalib, "COLLABORATIVE FILTERING APPROACH: SKINCARE PRODUCT RECOMMENDATION USING SINGULAR VALUE DECOMPOSITION (SVD)," *Medstat*, vol. 15, no. 2, pp. 139–150, Apr. 2023, doi: 10.14710/medstat.15.2.139-150.
- [11] S. Akansha, G. S. Reddy, and C. N. S. V. Kumar, "User Product Recommendation System Using KNN-Means and Singular Value Decomposition," in *2022 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, Bengaluru, India: IEEE, Dec. 2022, pp. 211–216. doi: 10.1109/CENTCON56610.2022.10051544.
- [12] D. Shahane, A. Nerurkar, S. Pansare, R. Ingale, and R. Narkar, "Music Fiesta-The Recommendation System," in *2022 1st International Conference on Technology Innovation and Its Applications (ICTIIA)*, Tangerang, Indonesia: IEEE, Sep. 2022, pp. 1–6. doi: 10.1109/ICTIIA54654.2022.9936009.
- [13] Y. Sun and Y. Zhang, "Conversational Recommender System," 2018, doi: 10.48550/ARXIV.1806.03277.
- [14] M. Nugraha, Z. K. A. Baizal, and D. Richasdy, "Chatbot-Based Movie Recommender System Using POS Tagging," *Building of Informatics, Technology and Science (BITS)*, vol. 4, no. 2, pp. 624–630, Sep. 2022, doi: 10.47065/bits.v4i2.1908.
- [15] D. Theosaksomo and D. H. Widyantoro, "Conversational Recommender System Chatbot Based on Functional Requirement," in *2019 IEEE 13th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, 2019, pp. 154–159. doi: 10.1109/TSSA48701.2019.8985467.
- [16] F. Narducci, M. De Gemmis, P. Lops, and G. Semeraro, "Improving the User Experience with a Conversational Recommender System," in *AI*IA 2018 – Advances in Artificial Intelligence*, C. Ghidini, B. Magnini, A. Passerini, and P. Traverso, Eds., in *Lecture Notes in Computer Science*, vol. 11298. Cham: Springer International Publishing, 2018, pp. 528–538. doi: 10.1007/978-3-030-03840-3_39.
- [17] J. Dalton, V. Ajayi, and R. Main, "Vote Goat: Conversational Movie Recommendation," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, Ann Arbor MI USA: ACM, Jun. 2018, pp. 1285–1288. doi: 10.1145/3209978.3210168.



- [18] A. N. Fajari and A. Baizal, “Chatbot-based Culinary Tourism Recommender System Using Named Entity Recognition,” *jipi. jurnal. ilmiah. penelitian. dan. pembelajaran. informatika.*, vol. 7, no. 4, pp. 1131–1138, Nov. 2022, doi: 10.29100/jipi.v7i4.3210.
- [19] Z. Abdurahman Baizal, N. Ikhsan, I. Muslim Karo Karo, R. Kenneth Darmawan, and R. Dwi Hartanto, “Movie recommender chatbot based on Dialogflow,” *IJECE*, vol. 13, no. 1, p. 936, Feb. 2023, doi: 10.11591/ijece.v13i1.pp936-947.
- [20] S. Reddy, S. Nalluri, S. Kuniseti, S. Ashok, and B. Venkatesh, “Content-Based Movie Recommendation System Using Genre Correlation,” in *Smart Intelligent Computing and Applications*, S. C. Satapathy, V. Bhateja, and S. Das, Eds., Singapore: Springer Singapore, 2019, pp. 391–397.
- [21] A. Singh, K. Ramasubramanian, and S. Shivam, *Building an Enterprise Chatbot: Work with Protected Enterprise Data Using Open Source Frameworks*. Berkeley, CA: Apress, 2019. doi: 10.1007/978-1-4842-5034-1.
- [22] Inderprastha Engineering College, AKTU *et al.*, “Movie Recommendation System using Cosine Similarity and KNN,” *IJEAT*, vol. 9, no. 5, pp. 556–559, Jun. 2020, doi: 10.35940/ijeat.E9666.069520.
- [23] T. Silveira, M. Zhang, X. Lin, Y. Liu, and S. Ma, “How good your recommender system is? A survey on evaluations in recommendation,” *Int. J. Mach. Learn. & Cyber.*, vol. 10, no. 5, pp. 813–831, May 2019, doi: 10.1007/s13042-017-0762-9.
- [24] D. Jannach, “Evaluating conversational recommender systems: A landscape of research,” *Artif Intell Rev*, vol. 56, no. 3, pp. 2365–2400, Mar. 2023, doi: 10.1007/s10462-022-10229-x.