Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 7 No. 3 (2023) 663 - 669 ISSN Media Electronic: 2580-0760

Ontology-based Conversational Recommender System for Recommending Camera

Restu Aditya Rachman[1], Z.K.A. Baizal[2]*
[1,2]Informatics, School of Computing, Telkom University
[1]restuar@student.telkomuniversity.ac.id, [2]baizal@telkomuniversity.ac.id

Abstract

The camera is a product that has developed very quickly in terms of specifications and functions. In addition, the cameras available on the market are becoming increasingly varied, so customers need more time to find a camera that suits their needs. Currently, many recommender systems have been developed to assist users in finding suitable products, especially the conversational recommender system (CRS). CRS is a recommender system that recommends products through conversations between the user and the system. However, many developed CRS still forces users to have knowledge of the product's technical characteristics. In the real world, many people are not familiar with the technical features of products, especially cameras. People interact more easily with CRS by stating the camera function they want. In this study, we call that statement functional requirements. Therefore, we proposed a CRS for recommending cameras that interact with users using functional requirements. This CRS uses semantic reasoning techniques on ontologies. To evaluate system performance, we use two parameters, i.e., user satisfaction and recommendation accuracy. The evaluation results show that the accuracy of the recommendations is at a value of 82.35%, and the level of user satisfaction reaches 0.66. With these results, the system can provide recommendations accurately and satisfy users.

Keywords: *camer; conversational recommender system; knowledge-based recommender system; ontology*

1. Introduction

In the last three years, shopping activity has increased quite significantly [1]. This activity becomes a user requirement in meeting their needs [2]. In addition, the products available on the market are becoming more varied, so customers need more time to study the specifications, especially products that have many technical features [3]. This problem causes difficulties for customers in choosing a product that suits their needs. Therefore, many recommender systems have been developed to make it easier for users to get the right product [4].

Recommender systems can assist users in making decisions by analyzing user preference information and recommending products according to their preferences [5], [6]. To get user preferences, recommender system can take two approaches: implicit and explicit [7]. The implicit approach recommends items based on the user's similarity to the rating data [8]. However, this approach can cause data sparsity and cold start problems. An explicit approach can avoid these problems [9]. The knowledge-based recommender system is an example of an explicit approach. Knowledge-based recommender systems look for solutions that suit user needs based on domain knowledge [10].

CRS is a knowledge-based recommender system that utilizes user-system conversations to get user preferences [11]. To get user preferences, CRS repeatedly asks several questions to users until they get the desired product [12]. Previously a CRS framework had been developed whose interactions were based on functional requirements [13]. This framework is multi-domain, meaning it can be developed for various domains.

There are two types of navigation in CRS: Navigation by asking (NBA) and navigation by proposing (NBP). This navigation strategy determines the interaction mechanism in the CRS. Baizal et. al. [13] utilize ontology and semantic reasoning to combine NBA and NBP in building interactions. The study proposes a knowledge-based recommender system. The combination of NBA and NBP can mimic a conversation between a potential buyer and a professional salesperson. Moreover, Cai et. Al. [14] develop CRS with NBP strategy. He incorporates critical techniques into CRS to make it easier for users

Accepted: 12-01-2023 | Received in revised: 15-04-2023 | Published: 02-06-2023

to search for music. Meanwhile, Habib et. al. [15] created a CRS for movies called IAI MovieBot. The system can display dialogs that flow specific to certain tasks, handle multi-modal communication, and adapt to changes in a user's preferences. Then, Zhang et. al. [16] proposed a conversational search recommender framework called the System Ask–User Respond (SAUR) paradigm. The framework is designed to be used in e-commerce and its main function is to assist in product search and recommendations, by defining its main components and implementation.

CRS as a form of knowledge-based recommender system, requires a form of knowledge representation to generate interaction. Rule-based programs, knowledge graphs, and ontologies are often used as knowledge representations [17]. Mentec et. al. [18] utilize ontology in recommending suitable candidates for certain job vacancies, considering the skills of the relevant workers, and providing explanations for these recommendations. Anelli et. al. [19] proposed an ontology to develop knowledge-aware and CRS beyond the traditional focus on accuracy. The aim is to introduce a new generation of algorithms and interactive approaches. Meanwhile, Zhou et. al. [20] propose knowledge-graph for enhanced recommender component for an enhanced recommender component to make accurate recommendations and generate informative keywords. The approach has been shown to be effective through experiments, resulting in improved performance on recommendation.

In this study, we propose a CRS based on functional requirements in the camera domain. We utilize the CRS framework from Baizal et. al. [13]. This framework can be used on multi-domains and elicit user preferences based on functional requirements. Based on evaluation results, this framework has higher user satisfaction than other frameworks that take user preferences from technical features. Furthermore, CRS framework utilizes ontologies and intelligent agents to generate interactions. Adjustments are needed in ontology development so that the system can recognize the functional requirements of the recent camera. We modified the information contained in the ontology to suit current camera products. The camera is a product that has many technical features, so that many people are not familiar with the features [2]. Thus, we hope that this CRS can help non-photographer users who want to find a camera. The CRS built in this study consists of two components, 1) the ontology domain and 2) the interaction generator algorithm. The system utilizes semantic reasoning to recommend cameras [21].

2. Research Methods

Baizal et. al. [13] developed a multi-domain CRS framework based on the functional needs of users. Ontology is used as a representation of knowledge. The CRS framework uses an ontology to map functional requirements to product specifications. However, ontology is fixed knowledge representation. So we need to adjust the formulation of the ontology domain. In this research, we develop this framework for the camera domain.

2.1 System Design

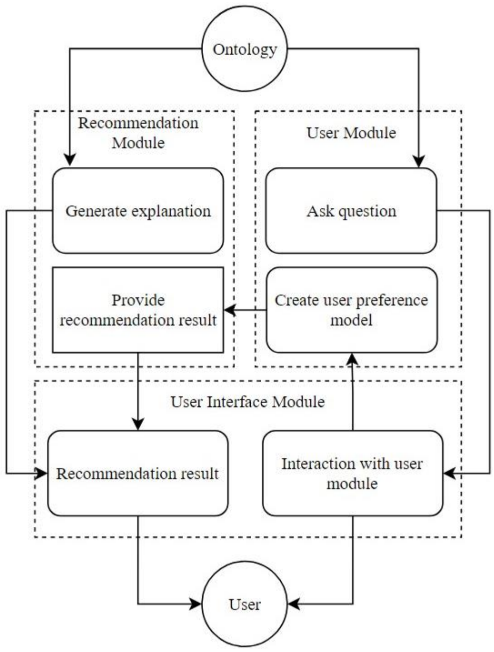

Figure 1. System design overview

In Figure 1, Each module provides a specific purpose. First, the user module connects the user interface with the ontology. This module has a function to prepare questions given to users and create a user preference module from the answers given by users. The second module is recommendations. This module has a function to search for recommendation results based on user preferences that have been made previously and provide an explanation of recommended products so that users can easily understand product specifications, such as what functional requirements this product supports. Third, the user interface module. This module functions to interact directly by displaying questions provided by the user module and recommendations from the recommendation module.

2.2 System Flow

Figure 2 shows the flow of user-system interaction. The interaction begins with the system providing a choice of the product's functional requirements. Then the user chooses the camera according to the user's wishes and can choose more than one. After that, the system determines whether the interaction carried out is sufficient to provide information or not. Otherwise, the system will ask questions about functional requirements again. If the information is sufficient, the system recommends several appropriate products.

DOI: https://doi.org/10.29207/resti.v7i3.4852

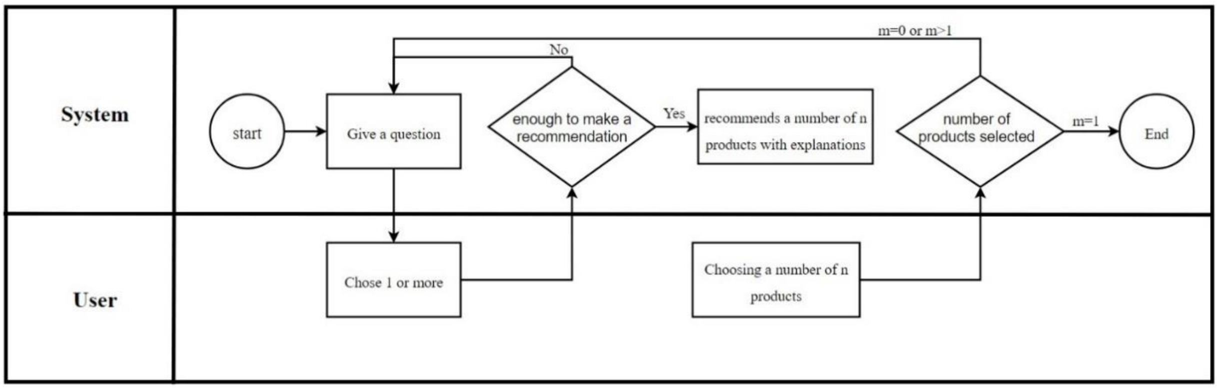

Figure 2. User-system interaction flow

In the final stage, the user selects the recommended product. If the user does not choose a product or more than one product, the system again asks about functional requirements. If the user chooses a product, the system has successfully provided a recommendation and is finished.

2.3 Camera Ontology

The ontology is separated into three categories: functional needs, detailed product descriptions, and actual products. Each class consists of several subclasses that form a hierarchy. Camera specifications are taken from the camera review site www.dpreview.com. We validate the data that has been obtained from the photographer user. After getting validation, we will adjust this data to be used in the ontology.

In Figure 3 the camera domain's functional requirements are organized into a class hierarchy. Questions and explanations are constructed based on this hierarchy.

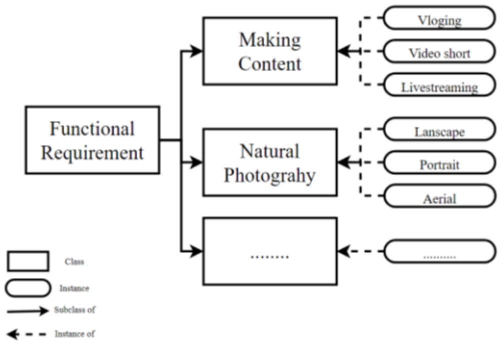

Figure 3. Functional requirements hierarchy [13]

Figure 4 is a specification class hierarchy for determining functional requirements with products that comply with specifications. Classifying product specifications based on their level of quality. Example for battery specification. Batteries of 1000 mAh, 1800 mAh, 2000 mAh, and others are expressed as high battery, medium battery, and low battery. Figure 5 is a product hierarchy that serves to present the types of products in the camera domain. This hierarchy groups cameras by body-type.

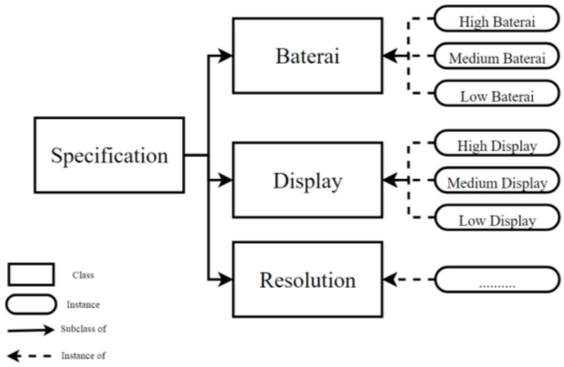

Figure 4. Specifications hierarchy [13]

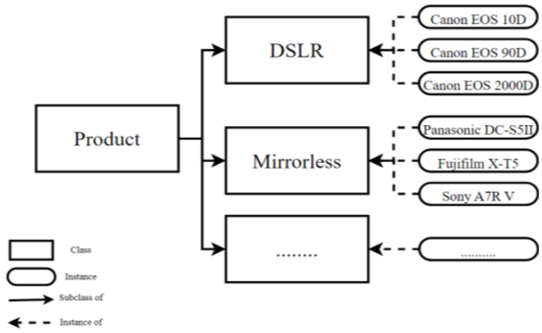

Figure 5. Product hierarchy [13]

To implement the camera ontology, we use the Protégé application version 5.5.0. We build an ontology design based on the 3 class hierarchies that have been described previously. This class hierarchy has several subclasses in it. The smallest class is called an instance. The results of the camera ontology are shown in Figure 6.

Each instance has an object property and a data property. Object property that aims to map each instance to a class specification. Otherwise, we use the data property to perform filtering at the beginning of the user-system interaction. Figure 7. Indicates an object property and data property that we use to support an ontology.

DOI: https://doi.org/10.29207/resti.v7i3.4852

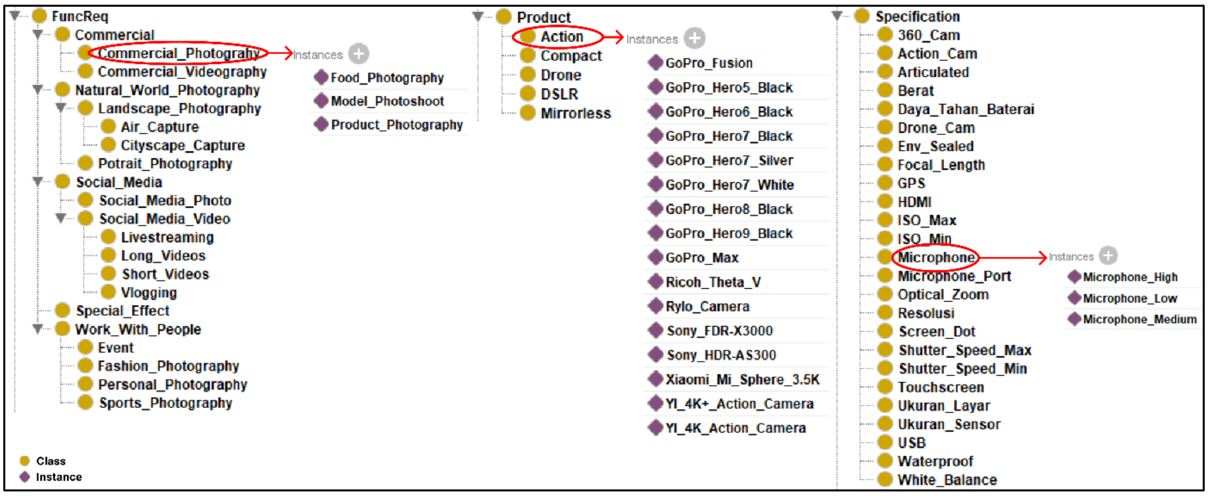

Figure 6. Implementation of ontology classes

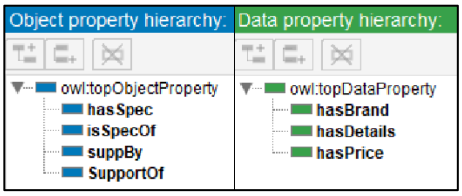

Figure 7. Object property and data property

There are four object properties that we use. Each property object has its role. `hasSpec` for assigning the product class to the specification class. `hasSpec` can be illustrated as follows "Product A has specification B". While `isSpecOf` is the opposite of `hasSpec`. In addition, `suppBy` assigns class functional requirements to class specifications. `suppBy` can be illustrated as follows "to satisfy functional requirements A requires specification B". While `supportOf` is the opposite of `suppBy`.

2.4 Query Refinement

Conversations in a CRS aim to improve the user's ability to express their preferences and to assist them in specifying those preferences more clearly [22]. Practically, users provide functional requirements that are still common when they first interact with the system. Thus, it is difficult for the system to recommend suitable cameras based on these requirements because there will be too many recommended products. Therefore, need a query refinement to narrow down the functional requirements. This can be overcome with query refinement [3]. The query refinement process enhances the user's preferences by inquiring about more specific functional requirements and potentially satisfying the user's wishes.

Algorithm 1: Query Refinement [3]

Data: *uModel, maxFr*
// *maxFr* number of questions related to functional requirements
Result: a group of inquiry from nodes (*nextFr*)

depthUser ← the depth of *uModel*;
uPref ← nodes [*FrMand* ∪ *FrOps*] at level *depthUser* of *uModel*;
subPref ← children of *uPref*;
if *subPref* ≠ *empty* **then**
  candQuestion ← *subPref*;
else
  candQuestion ← the undiscovered nodes that may be desirable;

In the process, Algorithm 1 requires the input parameter of the user model (`uModel`) and the maximum number of questions related to functional requirements (`maxFr`). In addition, there is a depth of user model (`depthUser`), user preferences (`uPref`), mandatory functional requirements (`FrMand`), optional functional requirements (`FrOps`), subclasses of user preferences (`subPref`), and candidate questions that are asked next (`candQuestion`) as constraints. Must be met when performing a refinement query. The resulting output is in the form of further questions asked to the user (`nextFr`) to obtain more detailed functional requirements.

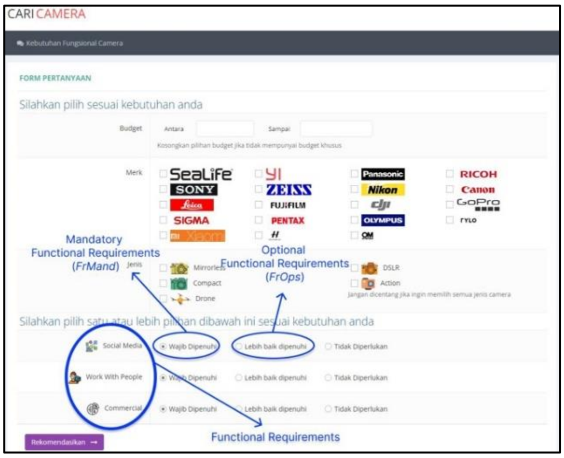

Figure 8. Initial user-system interaction

DOI: https://doi.org/10.29207/resti.v7i3.4852

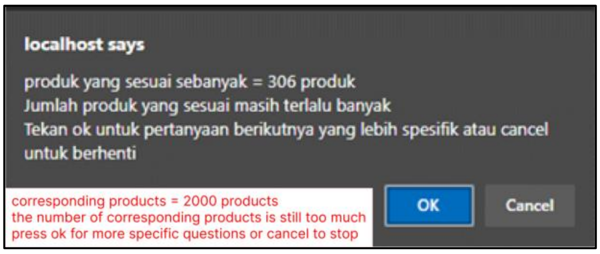

Figure 9. the system hasn't gotten enough user preferences

Figure 8 shows the initial interaction between the user-system. The questions asked are the functional requirements that were first generated by the system. Figure 9 is a message that appears if the system has not received enough user preferences. The system performs query refinement by Algorithm 1 to get more detailed user preferences. Algorithm 1 generates the functional requirement questions from the previous functional requirements subclass, as shown in Figure 10. Finally, Figure 11 illustrates the results of camera recommendations when the system has obtained sufficient user preferences.

Figure 10. The result of the refinement query

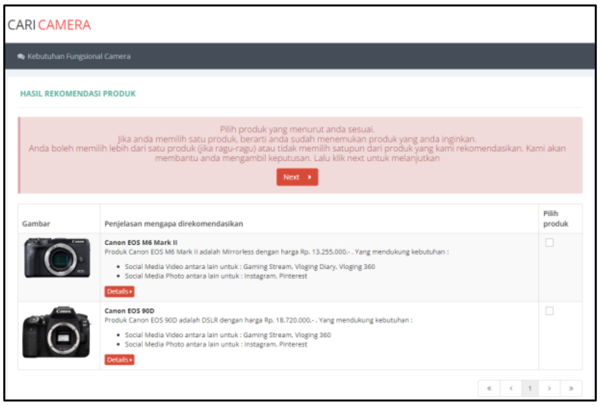

Figure 11. Recommendation results

3. Results and Discussions

For results and discussions, we tested the system on 60 users. This system comes from the difficulty of an unfamiliar user in choosing the right camera. So, we divide users into two categories: photographer users (familiar) and non-photographer users (unfamiliar) to determine “whether this system will more help the unfamiliar user”.

3.1 System Performance

System performance is measured by how the user operates the camera recommendation website that has been provided. When the user has succeeded in getting the camera they want, the user is directed to fill in the questionnaire by giving a value from 1 to 5. Values 4 and 5 indicate that the system has successfully recommended the camera, while values 1, 2, and 3 indicate that the system has failed to recommend the camera the user wants. A formula is needed to calculate recommendation accuracy as a measure of the success of a recommender system. The recommendation accuracy formula can be stated as formula 1.

$$Accuracy = \frac{succeessful\ recoomendations}{total\ recoomendations} \tag{1}$$

Figure 12 shows the average percentage of accuracy of 82.35%. From the gender category, the system gets 80.77% for males. While for females, it gets a percentage of 87.50%. Females get a higher percentage of accuracy than males. According to consumer behavior studies, Women tend to be more meticulous when selecting a product but can be easily influenced to choose products with various features and comprehensive information. This can happen because the system has explanation facilities to describe a product.

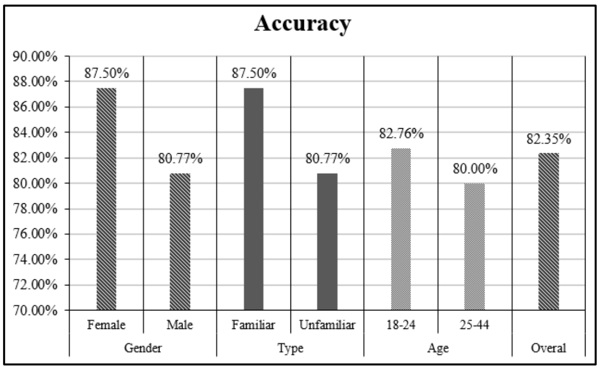

Figure 12. Recommendation system accuracy data

When viewed from the 44 unfamiliar users, the system gets an accuracy of 80.77%. This indicates that this system helps unfamiliar users. Also, the 16 familiar users get an accuracy of 87.50%. This system does not only help unfamiliar users. However, it also helps familiar users.

3.2 User Satisfaction

The questionnaire is used to measure user satisfaction. There are ten questions given in the questionnaire. To assist analysis, each statement is grouped into six factors: 1 perceived efficiency (PE), 2) Informative (INF), 3) trust (TR), 4) easy to use (ETU), 5) ease of understanding (EOU), 6) perceived quality of recommendation (PRQ) [12]. Details of each question can be seen in Table 1.

DOI: https://doi.org/10.29207/resti.v7i3.4852

Table 1. Statements of questionnaire

ID	Factors	Information
P1	PE	I find product that I want
P2	INF	I can find information of product easily
P3	TR	I really want to buy the product that I choose on system later
P4	TR	If I want to buy camera someday, I will use this system again
P5	ETU	I found it hard enough to find products that I really want
P6	ETU	It is easy to use this system
P7	EOU	The given questions or options are easy to understand
P8	EOU	I really understand all the question and option that I got
P9	PRQ	I like the product that I selected
P10	PRQ	I don't prefer this kind of interaction system

Based on Figure 13, ID P3 and ID P6 get minus values, which means that most users disagree with this statement. The result gave positive value for other IDs, which means that most users agree. It shows promising results for the six factors asked: EOU, PRQ, PE, INF, TR, and ETU.

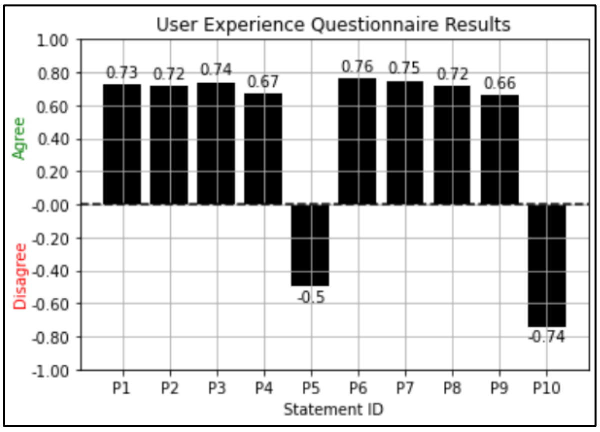

Figure 13. User satisfaction questionnaire data

4. Conclusion

Based on the results of system performance and user satisfaction that have been evaluated, the accuracy value of system performance is 82.35%, and The questionnaire results revealed positive user satisfaction based on six factors asked: EOU, PRQ, PE, INF, TR, and ETU. These results indicate that the recommender system has successfully provided accurate recommendations and appropriately interacted with users.

References

1] M. Z. Irawan and P. F. Belgiawan, "Ride-hailing app use for same-day delivery services of foods and groceries during the implementation of social activity restrictions in Indonesia," *International Journal of Transportation Science and Technology*, Mar. 2022, doi: 10.1016/J.IJTST.2022.03.004.

[2] R. N. Chandra, F. Febriyan, and T. H. Rochadiani, "Single camera body tracking for virtual fitting room application," in *ACM International Conference Proceeding Series*, 2018. doi: 10.1145/3192975.3192991.

[3] Z. K. A. Baizal, D. H. Widyantoro, and N. U. Maulidevi, "Query refinement in recommender system based on product functional requirements," *2016 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2016*, pp. 309–314, Mar. 2017, doi: 10.1109/ICACSIS.2016.7872760.

[4] C. Gao, W. Lei, X. He, M. de Rijke, and T. S. Chua, "Advances and challenges in conversational recommender systems: A survey," *AI Open*, vol. 2, pp. 100–126, Jan. 2021, doi: 10.1016/J.AIOPEN.2021.06.002.

[5] Q. Shambour, "A deep learning based algorithm for multi-criteria recommender systems," *Knowl Based Syst*, vol. 211, p. 106545, 2021, doi: https://doi.org/10.1016/j.knosys.2020.106545.

[6] A. Laksito and M. R. Saputra, "Content Based VGG16 Image Extraction Recommendation ," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 3, pp. 370–375, Jun. 2022, doi: 10.29207/resti.v6i3.3909.

[7] A. Gazdar and L. Hidri, "A new similarity measure for collaborative filtering based recommender systems," *Knowl Based Syst*, vol. 188, p. 105058, Jan. 2020, doi: 10.1016/J.KNOSYS.2019.105058.

[8] L. Quijano-Sánchez, I. Cantador, M. E. Cortés-Cediel, and O. Gil, "Recommender systems for smart cities," *Information Systems*, vol. 92. 2020. doi: 10.1016/j.is.2020.101545.

[9] S. Natarajan, S. Vairavasundaram, S. Natarajan, and A. H. Gandomi, "Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data," *Expert Syst Appl*, vol. 149, p. 113248, Jul. 2020, doi: 10.1016/J.ESWA.2020.113248.

[10] F. U. D. Laseno and B. Hendradjaya, "Knowledge-Based Filtering Recommender System to Propose Design Elements of Serious Game," in *Proceedings of the International Conference on Electrical Engineering and Informatics*, 2019, vol. 2019-July. doi: 10.1109/ICEEI47359.2019.8988797.

[11] W. Lei *et al.*, "Interactive Path Reasoning on Graph for Conversational Recommendation," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020. doi: 10.1145/3394486.3403258.

[12] Z. K. Abdurahman Baizal, Y. R. Murti, and Adiwijaya, "Evaluating functional requirements-based compound critiquing on conversational recommender system," in *2017 5th International Conference on Information and Communication Technology, ICoIC7 2017*, 2017. doi: 10.1109/ICoICT.2017.8074656.

[13] Z. K. A. Baizal, D. H. Widyantoro, and N. U. Maulidevi, "Computational model for generating interactions in conversational recommender system based on product functional requirements," *Data Knowl Eng*, vol. 128, p. 101813, Jul. 2020, doi: 10.1016/J.DATAK.2020.101813.

[14] W. Cai, Y. Jin, and L. Chen, "Critiquing for Music Exploration in Conversational Recommender Systems," in *International Conference on Intelligent User Interfaces, Proceedings IUI*, 2021. doi: 10.1145/3397481.3450657.

[15] J. Habib, S. Zhang, and K. Balog, "IAI MovieBot: A Conversational Movie Recommender System," *International Conference on Information and Knowledge Management, Proceedings*, pp. 3405–3408, Oct. 2020, doi: 10.1145/3340531.3417433.

[16] Y. Zhang, X. Chen, Q. Ai, L. Yang, and W. Bruce Croft, "Towards conversational search and recommendation: System Ask, user respond," *International Conference on Information and Knowledge Management, Proceedings*, vol. 10, no. 18, pp. 177–186, Oct. 2018, doi: 10.1145/3269206.3271776.

[17] A. F. RIfai and E. B. Setiawan, "Memory-based Collaborative Filtering on Twitter Using Support Vector Machine Classification," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 5, pp. 702–709, Oct. 2022, doi: 10.29207/RESTI.V6I5.4270.

[18] F. Mentec, Z. Miklós, S. Hervieu, and T. Roger, "Conversational recommendations for job recruiters," Sep. 2021, Accessed: Jan. 12, 2023. [Online]. Available: https://hal.inria.fr/hal-03537355

DOI: https://doi.org/10.29207/resti.v7i3.4852

[19] V. W. Anelli *et al.*, "Knowledge-aware and conversational recommender systems," *RecSys 2018 - 12th ACM Conference on Recommender Systems*, pp. 521–522, Sep. 2018, doi: 10.1145/3240323.3240338.

[20] K. Zhou, W. X. Zhao, S. Bian, Y. Zhou, J. R. Wen, and J. Yu, "Improving Conversational Recommender Systems via Knowledge Graph based Semantic Fusion," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1006–1014, Aug. 2020, doi: 10.1145/3394486.3403143.

[21] Z. K. A. Baizal, D. H. Widyantoro, and N. U. Maulidevi, "Design of knowledge for conversational recommender system based on product functional requirements," in *Proceedings of 2016 International Conference on Data and Software Engineering, ICoDSE 2016*, 2017. doi: 10.1109/ICODSE.2016.7936151.

[22] H. Xie *et al.*, "Incorporating user experience into critiquing-based recommender systems: a collaborative approach based on compound critiquing," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 5, pp. 837–852, May 2018, doi: 10.1007/S13042-016-0611-2/METRICS.