Ontology-based conversational recommender system for recommending laptop

M S Ayundhita¹, Z K A Baizal², Y Sibaroni³
School Of Computing, Telkom University, Bandung, Indonesia

E-mail: ¹ maulidyasurya@students.telkomuniversity.ac.id, ² baizal@telkomuniversity.ac.id, ³ yuliant@telkomuniversity.ac.id

Abstract. Currently there are many recommender systems for various products, but the recommender system still uses questions that refer to product specifications. For example, on laptop, the question is how much hard drive capacity is needed, what type of processor, etc. With the recommender systems previously described, customers who are not familiar with product’s specifications confused in choosing product to match with their needs. To solve these problems, a recommender system is required to prioritizes the functional requirements (high level requirement). Previously, we have developed a multi-domain framework for developing a conversational recommender systems (CRS) based on functional requirements. This framework comprises interaction generate method and ontology. The ontology aims to map functional requirements with technical features of the product. In this paper, we operationalize that framework and propose an ontology based on that framework for developing a CRS in a laptop domain. This CRS interacts with an iterative conversation to find out what the customer needs (e.g., customers need a laptop to watch video). This recommender system does the same conversation as a customer with a professional seller. The users involved in this test show that a recommender system that prioritizes functional requirements is more helpful in product selection than the recommender system commonly used in e-commerce.

1. Introduction

When buying products, many customers have difficulty in choosing the product because of there are many products available [1,2], that makes it difficult for customers to decide which products that match with their requirements. Sometimes, many buyers don’t have time to study laptop specifications that match their needs. Today, most people already have a laptop and it becomes one of the basic needs for urban society. Laptop is one of product that have many functional activities and many people are not familiar with the technical specifications [3]. With this problem, recommender system application has developed to make customers can easily choose products that match with their requirements [4].

There are previous researches that develop multi-domain framework for developing conversational recommender system (CRS) based on functional requirements [3,4,5,6]. This framework consists of interaction generate method and ontology. The main contribution of the work in this paper is to operationalize CRS framework and propose an ontology based on that framework for develop a CRS in a laptop domain. CRS is a recommendation system that doing conversation activities so that the system can find out the functional needs that the user wants [3,7]. The user-system interaction will end if user has chosen a product that matches with their requirements. The questions dialogue provided by the system
that aims to direct the specifications of the product, service, or information like user wants [5]. However, most CRS still use questions that refer to product technical specifications [4]. Such recommender system makes it difficult for customers who do not know the specifications of the laptop. For example, on laptop, the question that being asked is how much hard drive capacity is needed, what type of processor is desired, etc. Therefore, a recommender system is needed that asks about the functional requirements that customer needed. For example, customers need laptops to watch videos. Watching videos is one of the functional requirements of a laptop. CRS like this does the same conversation as customers with professional sellers [5]. The ontology structure aims to map functional requirements with technical features of the product and to become the foundation for constructing user profiles [6,8]. The recommendation is based on semantic relation on the ontology [6].

2. Overview of General System

In this research, we develop a recommender system for a laptop using conversational recommender system based on our previous CRS framework [3,4,5,6]. The framework uses navigation by asking and navigation by proposing [4]. In navigation by asking approach, users will be occupied with question and answer interactions where they are asked to answer specific questions about the features of thing need [9]. For the example, in recommending a laptop, the system asks user about RAM, camera, price range or processor that user wants. In navigation by proposing approach, users are presented with recommendations that have been suggested, then user is asked to provide feedback regarding the suitability of the recommendations previously given [9]. Users can provide rating-based feedback and critique-based feedback. This is a strategy to optimize the process of interaction more efficiently by minimizing the number of interactions.

When using the system, users will be given an option of functional requirements that will make the system provide recommendations according to user-provided interactions. After doing iteratively interaction, the system provides some laptop recommendations along with an explanation of the specifications. Mainly speaking, the system that being built is divided into three modules that each of it has its own task which is user model module, recommendation module, and user interface module.

![Figure 1. Overview of General System](image)

From figure 1, each module has its own function. First, user model module is a module that relates between the system and ontology. This module is responsible for preparing questions that will be given to users. Second, recommendation module is a module that aims to finding products that fit user's profile and generate an explanation of the products that will be recommended so that users will more easily understand the specifications, advantages and disadvantages of the product that will be recommended.
Third, user interface module is a module that related directly with users. This module is in charge of displaying questions that have been provided by user model module and displaying the recommendations provided by the recommendation module.

Figure 2 shows the system interaction flow from first interaction until finished. When user first interacts with the system, users will be given several choices of functional requirements. Then, user will choose functional requirements according to user’s wishes. After interacting, then the system will determine whether the interaction that has been done is enough to provide the information or not. This information will be used by the system to provide recommendations. If the information provided is not enough, the system will resume the conversation and if the information provided is sufficient, the system will immediately provide recommendations for laptop products along with their explanations. The last stage, user will choose the recommended laptop. If user selects one laptop product, the system has succeeded in giving recommendations. However, if user chooses more than one or does not choose the system will return to provide several functional requirements.

![Figure 2. Scheme of user-system interaction [4].](image)

2.1. Laptop’s ontology
The ontology model is divided into three classes which is: functional requirements, product and product specifications [5]. All classes have subclasses that make up the hierarchy. Laptop specifications are taken from the official laptop website. We and domain experts discuss the data that has been obtained to become knowledge in the recommendation system. Furthermore, the data are entered into ontology.

2.1.1. Hierarchy of functional requirements class. This hierarchy is a class hierarchy that presents the functional requirements of users in the laptop domain. Through this hierarchy questions and explanations will be raised.

2.1.2. Hierarchy of product specification class. This hierarchy is a class hierarchy that presents product specifications in the laptop domain. This hierarchy aims to map functional requirements with appropriate products. To define an instance, grouping product specifications will be based on the quality level [5]. If for example RAM 2GB, 4GB, 8GB, 16GB and others then for instances of RAM presented with low RAM, middle RAM, and high RAM.

2.1.3. Hierarchy of products class. This hierarchy is a class hierarchy that presents the types of products in the laptop domain.
2.2. User preference modelling

User preference modelling plays an important role in a recommender system that aims to prepare questions and create a user profile model based on user feedback [4,10]. The types of cases discussed are as follows [4]:

2.2.1. Empty user profile (Case U1). This case is the first time to interact with the system. In this case, user profile is still empty. The strategy is to determine the initial questions to begin interacting.

2.2.2. Multiple product selection (Case U2). In this case, user is still hesitate in choosing several products. The strategy used is to generate questions that refer to the distinguishing elements between product groups to help users choose from the number of products.
2.2.3. No product selected (Case U3). The strategy carried out in this case is to ask for alternative functional requirements at one level and backtrack to functional requirements at the previous level that have not been asked.

2.2.4. Insufficient definition of requirement to produce recommendation (Case U4). In this case the requirements are still too general. The strategy is to ask for more specific functional requirement (the next level). The question posed by the system in the next interaction is the functional requirement related to user’s previous answer.

2.2.5. No matched product based on user profile model (Case U5). In this case the strategy taken is to mark user profile that is irrelevant and to ask other functional requirements at a higher level of hierarchy.

2.3. System prototype
The application is built into two interaction models. The first application (model A) is the Conversational Recommender System which implements the interaction model asking for functional requirements. The second application (Model B) is a general recommender system used in e-commerce (the interaction model asks for technical features of the product) such as lazada, bhinneka, etc.

Figure 6. Display of user needs selection on model A.
3. Evaluation

We evaluate the applications with two types of evaluations. There are evaluate based on system performance and evaluate based on user satisfaction. In this test involving 39 users. Characteristics of users who take part of this test are people who can use a laptop and can use website-based applications.

3.1. Evaluate based on system performance

First, users are asked to try model A and model B. Afterwards, every user is being asked to answer the questionnaire. Every interaction between user and the system will be saved by the system. To measure the successful rate in recommending a product in the recommender system, it is necessary to calculate the accuracy of recommendations [3]. If the recommended recommendation product gets a rate of 5 or 4 indicates that the interaction was successful. If the recommended product is choosen and get rated 1 until 3, interaction is considered unsuccessful. The following is the formula for accuracy of recommendations:

\[
Recommendation \ accuracy = \frac{\text{number of successful recommendation}}{\text{number of recommendation}}
\]  

(1)

3.2. Evaluate based on user satisfaction

In addition to evaluating based on system performance, we use a questionnaire to evaluate user satisfaction [3]. The questionnaire is using 10 questions that contained in table 1. To help analyze, the statements are classified into six factors. There are: 1) easy to use / usability (ETU), 2) informative (INF), 3) perceived efficiency (PE), 4) ease of understanding (EOU), 5) perceived quality of recommendation (PRQ), 6) trust (TR).

---

**Figure 7.** Display of user needs selection on model B.
### Table 1. Statements of questionnaire [8].

| ID | Factor | Information |
|----|--------|-------------|
| SoQ1 | INF | I easily get information on product |
| SoQ2 | PE | I get the product that I like |
| SoQ3 | TR | Someday, I would buy the product that I chose on the system |
| SoQ4 | TR | I will use this system again if I want to buy a laptop |
| SoQ5 | ETU | I find it difficult to find a product that suits my requirements |
| SoQ6 | ETU | When using this system, I found no difficulty |
| SoQ7 | EOU | The given questions are easily to understand |
| SoQ8 | EOU | I easily understand the given questions |
| SoQ9 | PRQ | I like the product that I choose |
| SoQ10 | PRQ | I don't like this system interaction |

#### 3.3. The evaluation results

The way to find out the success of the A and B model applications is to compare the evaluation results based on system performance and user satisfaction. System performance is included in user satisfaction because if the system's performance is better, the user is more satisfied. Based on figure 6 and figure 7, recommender system with functional requirements questions (model A) have higher percentage than the general recommender system used in e-commerce (model B). It means model A is successful in recommending products match with user’s needs.

![Figure 8. Evaluation based on system performance.](image)

![Figure 9. Evaluation based on user satisfaction.](image)
4. Conclusion
Previously recommender system has been tested on system performance and user satisfaction. Recommendation accuracy obtained from testing based on system performance for the recommender system with the question of functional requirements is 84.6% while the accuracy of recommendations for general recommender systems used in e-commerce is 61.5%. Based on user satisfaction, recommender system with functional requirements questions have higher percentage than recommender system that contained in e-commerce on six factors (trust, easy to use / usability, ease of understanding, perceived recommendation quality, informative, perceived efficiency).

In general, the results of testing based on system performance and user satisfaction indicate that recommender system with the question of functional requirements is better and more useful than recommender system that contained in e-commerce. From the obtained test results, this is according to the hypothesis that a recommender system that uses functional requirements is more helpful in choosing the desired product.

References
[1] Jannach Z, Felfernig and Friedrich 2011 Recommender systems an introduction 2011 IEEE Intelligent Systems 14 pp 44-54
[2] Shimazu H 2002 Expert clerk: a conversational case-based reasoning tool developing salesclerk agents in e-commerce webshops 2002 Artificial Intelligence 233 – 244
[3] Baizal Z A and Murti Y R 2017 Evaluating functional requirements-based compound critiquing on conversational recommender system 2017 In Information and Communication Technology (ICoIC7) 2017 5th International Conference IEEE pp 1-6
[4] Baizal Z A, Widyanotoro D H and Maulidevi N U 2016 Factors influencing user’s adoption of conversational recommender system based on product functional requirements TELKOMNIKA (Telecommunication Computing Electronics and Control) 14(4) 1575-1585
[5] Baizal Z A, Widyanotoro D H and Maulidevi N U 2016 Design of knowledge for conversational recommender system based on product functional requirements 2016 Data and Software Engineering (ICoDSE) 2016 International Conference pp 1-6 IEEE
[6] Baizal Z K A, Iskandar A and Nasution E 2016 Ontology-based recommendation involving consumer product reviews 2016 Information and Communication Technology (ICoICT), 2016 4th Int. Conf. pp 1-6 IEEE
[7] Lorenzi F and Ricci F 2003 Case-based recommender systems: a unifying view 2003 Intelligent Techniques for Web Personalization (Berlin: Springer) pp 89-113
[8] Weng S S and Chang H L 2008 Using ontology network analysis for research document recommendation 2008 Expert Systems with Applications 1857-1869
[9] Baizal Z A, Widyanotoro D H and Maulidevi N U 2016 Query refinement in recommender system based on product functional requirements Advanced Computer Science and Information Systems (ICACIS), 2016 International Conference IEEE pp 309-314
[10] Hu B and Aufaure M A 2013 A query refinement mechanism for mobile conversational search in smart environment Proc. of the IUI 2013 Second Workshop on Interacting with Smart Objects pp 1-6