



## Conversational Recommender Systems Based on Criticism for Tourist Attractions using TF-IDF

Rayhan M Auliarahman, Z K Abdurahman Baizal\*, Nurul Ikhsan

Fakultas Informatika, Program Studi Informatika, Universitas Telkom, Bandung, Indonesia

Email: <sup>1</sup>rayhanma@student.telkomuniversity.ac.id, <sup>2,\*</sup>Baizal@telkomuniversity.ac.id, <sup>3</sup>ikhsan@telkomuniversity.ac.id

Corresponding Author Email: baizal@telkomuniversity.ac.id

**Abstract**—Tourist attractions are one of the attractions of tourist interest. There are many types of tourist attractions in an area, but this becomes a problem in itself because tourists will find it hard to find or determine a tourist attractions that suit their tastes. Many researches on recommendation systems based on criticism have been carried out with the aim of obtaining user preferences. However, only a few studies have conducted a critique-based recommendation system using the Conversational Recommender System (CRS). With this research, we will discuss a recommendation system based on criticism using natural language or CRS for tourist attractions in Bandung. In this study, we add assistance from the system to help users choose preferences or what can be called System-suggested Critiques (SC), users more easily determine preferences for the system. We use the Term Frequency-Inverse Document Frequency (TF-IDF) to determine critiques submitted to users. Based on the results of an evaluation involving 88 respondents who were asked to fill out a questionnaire after trying the system built, it was found that users were quite satisfied with the system we built. And obtained 62.06% system accuracy which proves that the system performance is quite satisfactory.

**Keywords:** Tourist Attractions; Recommendation System; Chatbot; Critique-Based; User-Initated Critiquing; System-Suggested Critiquing

### 1. INTRODUCTION

In this study, we implement a critical-based chatbot for tourism locations in the city of Bandung, which will be tested for efficiency, level of difficulty, and user satisfaction with the application. we present an online evaluation of the Hybrid-Critiquing system. With system-suggesting critiquing using TF-IDF for criticism recommendations submitted to users. The chatbot will help users by asking for attributes that need criticism. Chatbots will be able to interact with users using the RASA Chatbot Framework and buttons to criticize the recommendations that will be given. The recommendation system uses content-based filtering with the TF-IDF method, because it will ask for input from users in the form of text such as asking for tags such as park, shopping centers, and others, and activities that can be done at these tourist attractions such as watching, shopping, relaxing, studying, and others. Which is where the system is able to receive Natural Language or natural language from users, like communicating from person to person. The system will be able to receive critical messages such as, 'Request a cheap tourist spot to relax' or 'Request a natural tourist spot that I can do outbound'. Because this research was conducted in the city of Bandung, therefore this research was conducted using Indonesian.

With so many tourist attractions, people find it hard to find or determine a tourist attraction they want to visit. According to the ETC, people choose the internet to be their primary source of information to find tourist attractions that suit their tastes. But the internet doesn't have complete information [1].

Recommendation systems using a Critique-based approach have proven successful in recommendation systems because they help the system get feedback from users [2]–[4]. Critique-based is an approach to evaluating the output of system recommendations, allowing the system to filter recommendations that will be given to users to better suit user interests. After the system displays a product recommendation for the user then the user can provide feedback (Critics) for example, "something cheaper", "wider screen", etc. The input is in the form of several aspects of the available product space, the simplest criticism is in the form of product attributes or features [2].

There are 2 ways Conversational Recommender System (CRS) can get user preferences, namely with Navigation By Asking (NBA) or Navigation By Proposing (NBP) [3]. In this study, we used NBP to perform System-suggesting critique (SC). SC has been proven to increase the accuracy of the recommendations generated by the system and reduce the time to use the system [2].

Due to the lack of research that has studied different criticizing techniques by way of Conversational User Interfaces (CUI) which mimic conversations with real humans both text and voice Navigation By Asking (NBA) and Navigation By Proposing (NBP) are navigational strategies used in the CRS to get user preferences [3]. In this study, we used NBP to perform System-suggesting critique (SC). SC has been proven to increase the accuracy of the recommendations generated by the system and reduce the time to use the system [2]. Conversation recommendations allow users to have conversations with the system so that they can clarify preferences according to the user's wishes [5].

Based on the way in which criticism is made, There are two types of critique-based recommendation systems, namely: System-suggesting critique (SC) and User-initiated critique (UC). And there are 2 types of criticism that can be done by the user, namely Unit Critiques and Compound Critiques. Unit Critiques refers to criticism that limits only one feature at a time. While for multiple criticism refers to criticism that limits several



features at the same time, this multiple criticism can also improve the performance of the recommendation system and its accuracy because it asks for feedback from users [2].

The system will be built on the Telegram platform, making it more flexible for use on mobile and desktop. The data used are domiciled in the city of Bandung, because the testing session will be held in the city of Bandung first. In the future, it may be possible to reproduce the data, not only in the city of Bandung. The system built will use recommendations based on criticism where the system will receive or request criticism from several data attributes. Then the system is also built using the Content-Based Filtering method, and a jump start will be carried out where new users will have to choose 3 tourist attractions from the 3 list of tourist attractions that have been provided [6], therefore the system will not get Cold-Start problems, because it will not ask for search history data from users, but ask for preferences from users directly.

We use Telegram because Telegram proved to be very popular messaging service and has an open source platform, and all the services that Telegram has are completely free. Telegram was built in 2013 by a Russian-born entrepreneur. Telegram has also become a very popular messaging service in forty countries, such as Germany and the United States, beating other messaging services [7], [8].

We used the *RASA* framework for this study, *RASA* is an open source machine learning framework for automated text and voice-based conversations. *RASA* can understand the message the intent of the message the user send and respond. *RASA* also can be connected to various social media that have bot services, and APIs. *RASA* is able to take actions that refer to a context and can reply to unspoken conversations and direct the conversation when the user deviates from the normal course of conversation [9].

### 3. RESEARCH METHODOLOGY

#### 2.1 Chatbot

A chatbot is an application or software capable of providing services using natural language conversations. Which aims to provide services to users efficiently [1]. using the Natural Language Processing (NLP) module, which can analyze keywords from the text sent by the user. with that the system will know the intent of the text sent by the user [10]. Chatbots have become easier to use and can help with everyday things like information retrieval tools, auto-responders, business, and e-commerce [11].

#### 2.2 Critique-based

Critique-based or critique approach is a recommendation system approach where the system can use multiple criticisms from users on various attributes for the recommendation system [11]. Critique-based is divided into 2, namely User-Initiated Critiquing (UC), and System-Suggested Critiquing (SC).

UC, is a flexible criticism approach because the user can determine which features and how they are criticized. UC is very useful for reducing the value of the less important attribute and will get a more optimal value for the important attribute. Some previous studies examining UC found that some users had difficulty initiating criticism without knowing its attributes [12].

SC, is a critical approach that suggests criticism for the feature. The system will ask the user whether it is necessary to criticize the attribute or not. However, in this paper we will not use a Knowledge-based recommender, so this system does not require any knowledge of datas. This approach also makes it easier for users to use the recommendation system, because they get directions from the system, and can reduce the duration of using the system [2], [11].

Many Conversational Recommender Systems (CRS) have been created for research and commercial use. By using the Dialog-based technique by asking questions to collect information about the product that the user is looking for [13], [14].

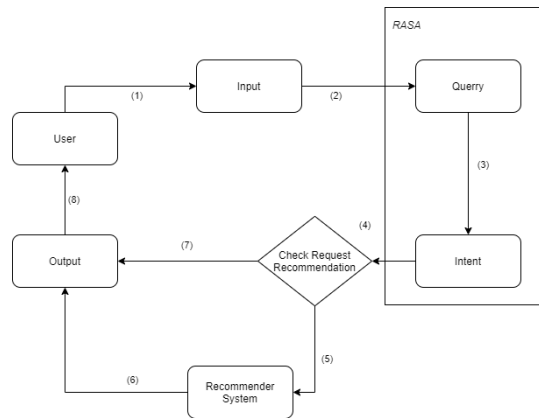
There are several approaches to implementing Critiques-based. Unit Critiques are used on one of the attributes or product features, usually facilitating the type of feedback or criticism such as 'more,' less, or other types of criticism. Then there is Compound critiques, which allows you to critique several product attributes or features at the same time [12]. In this test we will use NBP and navigation with keys to avoid typos for some functions. And use Compound Critiques and Unit Critiques for option critique for users.

#### 2.3 System Flow

Figure 1 illustrates the process flow for requesting tourist recommendations from TourismBot. This chatbot accepts input sent by the user and will display recommendations in the form of text. This system performs several steps: (1) the user enters or sends text to the chatbot. (2) After receiving text from the chatbot user, transfer it to the system to understand the user's intent by using Natural Language Understanding (NLU). (3) After understanding the intent of the text that the user sends, then the text is equated with the intent that has been built previously. (4) After being matched with the appropriate intent then *RASA* sends data for processing. (5) If the user input intend on requesting a tourist spot recommendation, the system will run the recommendation system function. (6) The results of the recommendation function will prepare tourist spot recommendation data for users.



- (7) If there is no intent to request a tourist spot, RASA will send a pre-made message based on the intended intent.
- (8) The user receives an informational text message from a tourist spot recommended by the system and 2 buttons for feedback and 2 buttons for other actions.



**Figure 1.** Block System Flowchart

**2.4 Dataset**

The data used in the form of data on tourist attractions in the city of Bandung. The contents of the data used are the price of admission tickets, and the type of tourist attractions (natural tourism, culture tour, amusement parks, etc.), and sub tags (Lodging, shopping centers, restaurant, and others). The data were obtained from previous studies [15], [16] which examined the recommendation system in the city of Bandung. The data used may not be 100% relevant because there is no new data that has been updated in 2021.

**Table 1.** Sample data used for testing

id_tour	Name_Tourism	Travel_address	Travel_Category	Entry_fee	Facility	Category
1	Tangkuban Perahu	Cikahuripan, lembang, Kab.Bandung Barat	Wisata alam dan Outbound	50000	Hotel dan penginapan, rumah makan, toko oleh-oleh	Wisata alam dan Outbound Hotel dan penginapan, rumah makan, toko oleh-oleh

**2.5 TF-IDF**

With the TF-IDF method we can get keywords automatically using TF-IDF, and calculate the similarity between items. with this method the system can determine the relative frequency of words that appear in the document, through the inverse of the proportion of words throughout the document [17]. TF-IDF is used to display tourist attractions recommendations and critique recommendations for users.

$$W_{i,j} = t_{f_{i,j}} \times \log\left(\frac{N}{df_j}\right) \tag{1}$$

TF-IDF Consists of two calculations:

- a. Term Frequency (TF), is the weight or value of a word whose value is obtained from the frequency of occurrence of the word in the document in question.
- b. Inverse Document Frequency (IDF), are the values calculated how words are widely distributed in the collection of relevant datas.

$$idf(j) = \log\left(\frac{N}{df_j}\right) \tag{2}$$

Where :

- W : is the weight of the document against the search word
- Tf : the number of words searched for in a document
- N : amount of data
- df<sub>j</sub> : number of documents containing the word (j)

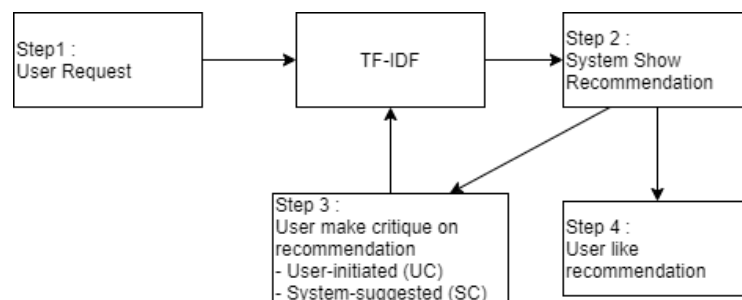


## 2.6 Algorithm

Users must enter text input to request recommendations from the system. The system will display recommendations for tourist attractions one by one based on the favorite tourist attractions as many as 20 tourist attractions ( $K = 20$ ) from the last favorite tourist attractions. For new users, we provide 3 tourist lists of which 1 list is filled with 4 tourist attractions. We ask first first for users to like tourist attractions that have been provided by category. There are 3 lists that have been prepared, the user must like one of the 3 lists that have been provided. With users liking tourist attractions at the beginning, it helps Jump-Start user profiles using the system and avoids Cold-Start [9]. Critique-based algorithm, user can start criticism by sending text message (UC) or ask a bot for help by selecting "Minta Tolong Bot" button. Bot" (SC) on ToursimBot. UC and SC are used dynamically to set user preferences. For example, users can make criticism based on attributes related to tourist attractions (Example: Types of tourist attractions, Activities carried out at tourist sites, etc. by themselves (Example: "Request cheap swimming spots") or request system assistance for provide suggestions (Example: 'We think you will like natural attractions that can do activities such as outbound.'). The data column used to perform TF-IDF is the 'Category' column as shown in table 1. To calculate the similarity of data with user preferences, TF-IDF will be used. The results of SC can produce one or two attributes and therefore produce compound critique. To generate critique suggestions for users using the TF-IDF method through 4 Steps:

- The system will take all the data of tourist attractions that have been liked by the user.
- Then the system will use TF-IDF which looks for word that often appear in the datas
- After getting the TF-IDF values for each word, it is then processed whether there are attribute to make criticism. After that, they are sorted from largest or most frequently occurring value.
- Then it is given to the user whether this criticism will be used or not.

If the user accepts then the system will re-filter the data according to the preferences that have been entered by the user and suggested by the system. if the user rejects the suggestion of criticism. the system will continue recommendations based on criticism before the user selects the "Request Bot Help" button. Then do TF-IDF to choose a recommended tourist spot to be displayed. Figure 2 illustrates the implementation of TF-IDF on the system being built. Before the system provides recommendation, the system will use TF-IDF to calculate the similarity of the data with the tourist attractions that the user last liked.



**Figure 2.** Flowchart Recommendation System

## 3. RESULT AND DISCUSSION

We conduct user research testing ( $N = 88$ ) to measure the level of user satisfaction and accuracy of the systems built by SC and UC.

### 3.1 Participant

We invited 133 participants through personal contact. A total of 33 participants' data was deleted because they did not try the system that had been built. And 12 participants whose data was deleted due to not completing the test. Finally we used 88 participants whose data was used to perform this test. (Age: 18-30 (74),31-40 (3),41-50 (1),51-60 (10); Gender: Male = 20 , Female = 68)

### 3.2 Test procedure

the task of the experiment is to find tourist attractions that are liked by the participants. The task performed by the participants contains the following 3 steps:

- Learning tutorial – Participants first read the user study description and introduction to the main features of the system written on the google form.
- Build a user profile – to build a User Profile, the user must first like 3 tourist attractions from the 3 list of tourist attractions that have been made previously
- Post-study questionnaire – After testing the scenario the participants is asked to fill out a questionnaire in which the questions are based on research [2].



Participants were asked to complete a post-study questionnaire to measure the built system from the user's point of view. Questions were asked of users based on [2]. Which are questions P10-P12 to measure how influential criticism is on user experience. All questions are rated on a 10-point scale which is on a scale of 1 (Strongly disagree) to 10 (Strongly Agree).

**3.3 Interaction Log**

**Table 2.** Questions for the user experience questionnaire with TourismBot

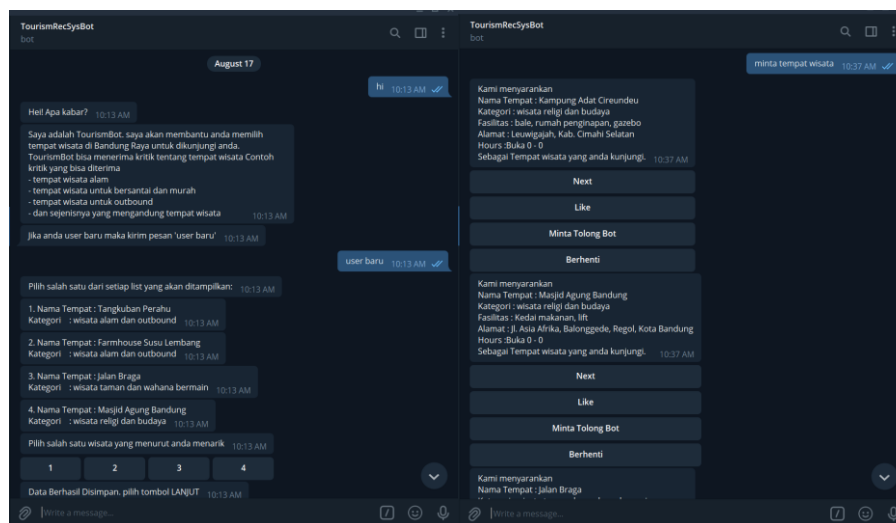
Question
P1 : The recommendations are of interest to me.
P2 : Easily find the tourist place i want to visit .
P3 : I use this application easily.
P4 : The tourist attractions that I receive are various.
P5 : I can easily tell the system that I like the recommended tourist spots.
P6 : I feel I can change my taste using TourismBot.
P7 : I'm sure I would have liked tourist places that TourismBot give to me.
P8 : After receiving recommendations for tourist attractions I like to give feedback on tourist attractions recommended to me . (Like on recommended tourist attractions)
P9 : This TourismBot trustworthy.
P10: TourismBot is easy to understand in conversation by me .
P11: When I have a conversation with TourismBot, I know what I can say or do in each dialogue .
P12: TourismBot works as I would expect in this conversation.
P13: I will use again TourismBot
P14: I am satisfied with using TourismBot to search for tourist attractions I want to visit .

We store user interactions with TourismBot. Interactions are stored in the form of whether the user uses bots for criticism or not.

**3.4 Interface Design**

The interface built via telegram which can fulfill several requirements 1) can display data on tourist attractions that will be recommended and 2) the user can make a critique of the system (UC) or the system can suggest criticism (SC). Since not much can be changed from its Telegram we prepared some help for user easy to use bot. We provide a help button to navigate to the system being built. As in figure 3.

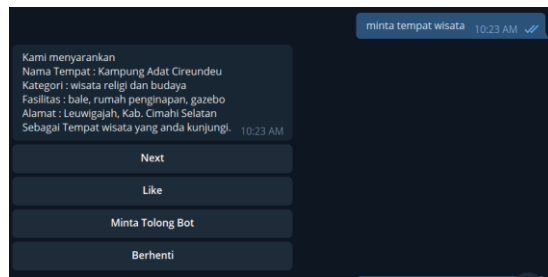
To run the program the user is given the buttons that have been prepared in advance. To make users avoid mistyped words, so that the system can run according to the flow. With the button the user can reply to messages quickly and precisely as in the study [18].



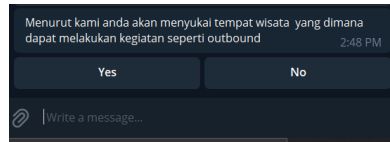
**Figure 3.** Views for TourismBot

For each tourist spot recommendation provided by TourismBot, four buttons are provided which are used to provide feedback to the system (Figure 4). The 'Next' button will allow users to see recommendations for the next tourist spot on the list. Then there is a 'Like' button, the recommended tourist attractions displayed will be included in the user's favorite list of tourist attractions. The 'Minta Tolong Bot' button triggers the System-suggesting critiquing function to display critique suggestions for users from a list of favorite tourist attractions (Figure 5). The 'Berhenti' button will stop the system from displaying recommended tourist attractions.





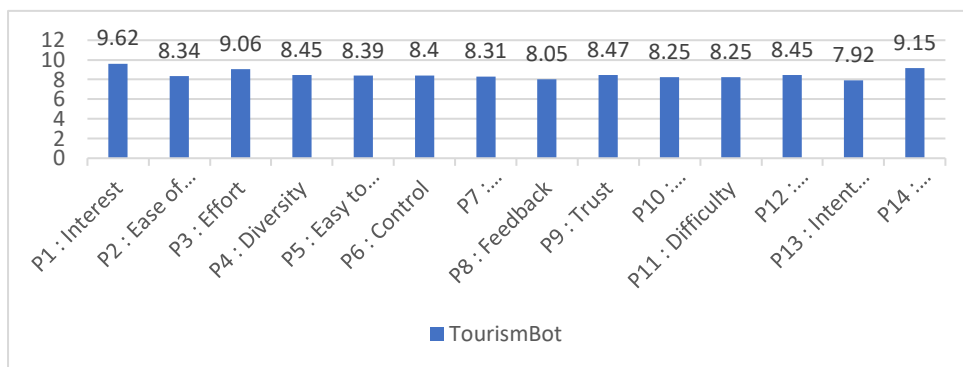
**Figure 4.** Conversation flow view for user-initiated critiquing



**Figure 5.** Conversation flow view for System-suggested critiquing

**3.5 Subjective Experience**

Figure 6 displays the results of the user's response to the questions in the table 2.



**Figure 6.** User satisfaction and usability assessment results

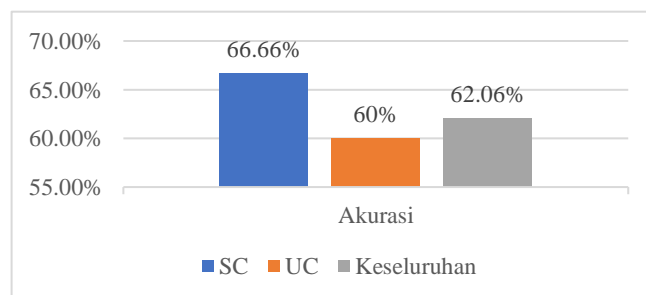
The system gets a very high average score from the user's side, as well as from the recommendations or the User-Interface. Overall, users really like the recommendations submitted by the system (P1 & P14). According to users, the system is easy to use to get recommendations for tourist attractions (P2, P3, P5, P6, P11). With the criticism feature, users trust the recommendations generated by TourismBot (P9) more. And with the critique feature as well as UC or SC greatly affects user satisfaction with using TourismBot (P10, P11, P12).

**3.6 Recommendation Accuracy**

In this section, we will discuss the accuracy of the system recommendations that have been made. Here we will measure accuracy with the help of the system (SC) and without the help of the system (UC). To determine the accuracy of the system we have built, we use the formula:

$$Accuracy = \frac{Number\ of\ successful\ recommendations}{Number\ of\ recommendations\ given} \times 100\% \tag{3}$$

By using the formula obtained from [3]. And taking data from the user's favorite list generated by the system for tourist spot recommendations. Then the results are shown in figure 6.



**Figure 7.** System Accuracy



From the results of user data shown in figure 7, an accuracy of 60% is obtained without asking for help from bots. and 66.66% with the help of bots. Therefore it can be concluded that with the help or suggestions of criticism for users can improve the accuracy of recommendations to users. The accuracy obtained from the system is different from that obtained from user reviews, where the accuracy value can be said to be very high, while measured from the database, the accuracy is 62.06%.

#### 4. CONCLUSION

The system built in this study received very good feedback from the user's side, as well as from the way it works or how easy it is to use the system built, has an average value of 8,533 which is shown in figure 1. by using the RASA framework and using telegram API, the feedback obtained from users is quite good and makes it easier for users to access the system. The accuracy obtained from the results of trials by users is 62.06% overall. And there is a difference in accuracy for users who use SC and those who don't (UC), SC gets a higher accuracy of 66.66% compared to those who don't, which is 60%. With this it can be concluded that the SC can help users to choose criticism so that the recommendations submitted by the system are more suitable for users. This proves that SC can increase the accuracy of the recommendations given to users. In the future, it may be possible to research again for the features of other tourist attractions or try for other domains. And using more complete and up-to-date data because the data used by tourist attractions in Bandung 2016, so some data may be irrelevant. And due to the limited performance of the computer used during testing, there are frequent crashes on the system when undergoing the testing process to users.

#### REFERENCES

- [1] M. Anranur Uwaisy, Z. K. A. Baizal, and M. Yusza Reditya, "Recommendation of scheduling tourism routes using tabu search method (case study bandung)," *Procedia Comput. Sci.*, vol. 157, pp. 150–159, 2019.
- [2] Y. Jin, W. Cai, L. Chen, N. N. Htun, and K. Verbert, "MusicBot: Evaluating critiquing-based music recommenders with conversational interaction," *Int. Conf. Inf. Knowl. Manag. Proc.*, pp. 951–960, 2019.
- [3] Z. K. Abdurahman Baizal, Y. R. Murti, and Adiwijaya, "Evaluating functional requirements-based compound critiquing on conversational recommender system," *2017 5th Int. Conf. Inf. Commun. Technol. ICoICT 2017*, vol. 0, no. c, 2017.
- [4] H. Xie *et al.*, "Incorporating user experience into critiquing-based recommender systems: a collaborative approach based on compound critiquing," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 5, pp. 837–852, 2018.
- [5] B. Priyogi, "Preference elicitation strategy for conversational recommender system," *WSDM 2019 - Proc. 12th ACM Int. Conf. Web Search Data Min.*, pp. 824–825, 2019.
- [6] C. Rosenfeld and N. Strand, "Getting To Know You," *Diabetes Self. Manag.*, vol. 33, no. 2, pp. 110–111, 2016.
- [7] T. Sutikno, L. Handayani, D. Stiawan, M. A. Riyadi, and I. M. I. Subroto, "WhatsApp, viber and telegram: Which is the best for instant messaging?," *Int. J. Electr. Comput. Eng.*, vol. 6, no. 3, pp. 909–914, 2016.
- [8] A. Akbari and R. Gabdulhakov, "Platform surveillance and resistance in Iran and Russia: The case of Telegram," *Surveill. Soc.*, vol. 17, no. 1–2, pp. 223–231, 2019.
- [9] Rakesh Kumar Sharma, "An Analytical Study and Review of open source Chatbot framework, Rasa," *Int. J. Eng. Res.*, vol. V9, no. 06, 2020.
- [10] A. M. Rahman, A. Al Mamun, and A. Islam, "Programming challenges of chatbot: Current and future prospective," *5th IEEE Reg. 10 Humanit. Technol. Conf. 2017, R10-HTC 2017*, vol. 2018–Janua, pp. 75–78, 2018.
- [11] M. Güell, M. Salamó, D. Contreras, and L. Boratto, "Integrating a cognitive assistant within a critique-based recommender system," *Cogn. Syst. Res.*, vol. 64, pp. 1–14, 2020.
- [12] L. Chen and P. Pu, "Critiquing-based recommenders: Survey and emerging trends," *User Model. User-adapt. Interact.*, vol. 22, no. 1–2, pp. 125–150, 2012.
- [13] S. Ghose and J. J. Barua, "Toward the implementation of a topic specific dialogue based natural language chatbot as an undergraduate advisor," *2013 Int. Conf. Informatics, Electron. Vision, ICIEV 2013*, pp. 0–4, 2013.
- [14] P. Cordero, M. Enciso, D. López, and A. Mora, "A conversational recommender system for diagnosis using fuzzy rules," *Expert Syst. Appl.*, vol. 154, 2020.
- [15] A. Maalel and M. Lara, "Travel route scheduling based on user 's preferences using simulated annealing-agent system intelligent travellin ... Related papers."
- [16] F. H. Prabowo, K. M. Lhaksmana, and Z. K. A. Baizal, "A multi-level genetic algorithm approach for generating efficient travel plans," *2018 6th Int. Conf. Inf. Commun. Technol. ICoICT 2018*, vol. 0, no. c, pp. 86–91, 2018.
- [17] B. Trstenjak, S. Mikac, and D. Donko, "KNN with TF-IDF based framework for text categorization," *Procedia Eng.*, vol. 69, pp. 1356–1364, 2014.
- [18] C. Pricilla, D. P. Lestari, and D. Dharma, "Designing Interaction for Chatbot-Based Conversational Commerce with User-Centered Design," *ICAICTA 2018 - 5th Int. Conf. Adv. Informatics Concepts Theory Appl.*, pp. 244–249, 2018.