

Movie recommender chatbot based on Dialogflow

Zinke Abdurahman Baizal¹, Nurul Ikhsan¹, Ichwanul Muslim Karo Karo², Reinaldo Kenneth Darmawan³, Roby Dwi Hartanto⁴

¹School of Computing, Telkom University, Bandung, Indonesia

²Department of Mathematics, State University of Medan, Medan, Indonesia

³Ninjavan Pte Ltd, Singapore

⁴Bandung Techno Park, Bandung, Indonesia

Article Info

Article history:

Received Dec 13, 2021

Revised Aug 8, 2022

Accepted Sep 3, 2022

Keywords:

Chatbot

Conversational recommender system

Dialogflow

Movie recommendation

Recommender system

ABSTRACT

Currently, the online movie streaming business is growing rapidly, such as Netflix, Disney+, Amazon Prime Video, HBO, and Apple TV. The recommender system helps customers in getting information about movies that are in accordance with their wishes. Meanwhile, the development of messaging platform technology has made it easier for many people to communicate instantly. Utilizing a messaging platform to build a recommender system for movies, provides special benefits because people often access the messaging platform all the time. In the Indonesian language, there are many slang terms that the system must recognize. In this study, we build a chatbot on a messaging platform which users can interact with the system in natural language (in Indonesian language) and get recommendations. We use rule-based and maximum likelihood as a method in natural language processing (NLP), and content-based filtering for the recommendation process. The recommender system interaction is built through a conversation mechanism that will form a conversational recommender system. The interaction is based on a chatbot which is built using Dialogflow and implemented on the telegram. We use the accuracy of recommendations and user satisfaction to evaluate the system performance. The results obtained from the user study indicate that the NLP approach provides a positive experience for users. In addition, the system also produces an accuracy value of 83%.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Zinke Abdurahman Baizal

School of Computing, Telkom University

Telekomunikasi St., Number 1, Terusan Buah Batu, Bandung, Indonesia

Email: baizal@telkomuniversity.ac.id

1. INTRODUCTION

The online video streaming business is growing rapidly at this time, along with the development of high-speed internet (4G and 5G). Many platforms play in this area, such as Netflix, Disney+, HBO Max, Hulu, Apple TV, and Paramount. Information overload in this movie domain makes users need a recommender system to help them get the movies they want. In addition to the movie domain, recommender systems have also been developed in many other domains, such as tourism, e-commerce, and culinary [1], [2], [3]. Along with the development of artificial intelligence technology in the field of natural language recognition, a form of recommender system has been developed that allows a two-way mechanism between the user and the system (a conversational mechanism). The conversational recommender system (CRS) is a knowledge-based recommender system that allows users to interact with the system through a question-and-answer mechanism. This CRS is built as a solution to create an interactive system so that users can more

freely express their needs. Sometimes in everyday life, the user is not necessarily able to express specific needs at the beginning of the interaction, so the iterative conversation is expected to reflect the user's needs.

In this study, we develop a chatbot-based CRS, in which user and system interact in natural language. However, in the Indonesian language, there are lots of slang terms and lots of ways for people to express something. We must develop a chatbot that is able to recognize these slang terms. Thus, this chatbot utilizes natural language processing (NLP) so that the system is able to understand the sentences of the user. Related research in the area of the conversational recommender system that utilizes NLP has been carried out by Nica *et al.* [4] in the tourism domain. The study used the concept of model-based diagnosis and Shannon's information entropy. This approach can solve the CRS problems, i.e., the inconsistency between chatbot knowledge and user requirements.

Several researchers developed chatbots for recommender systems. Colace *et al.* [5] developed a chatbot system using Petri Net to recommend vehicle tires. The research got the results of giving correct recommendations by 65.32%, giving correct recommendations but not in accordance with user needs by 13.87%, and giving wrong recommendations by 20.81%. Vaira *et al.* [6] also developed a chatbot to provide relevant instructions to pregnant women and families with young children based on the current situation. CRS is built using the NLP approach to process text sent by the user. This CRS uses the Microsoft Bot framework and is implemented on the Telegram platform. Meanwhile, Dalton *et al.* [7] built a chatbot, "Vote Goat" that can receive voice input. This chatbot recommends movies using Dialogflow and is implemented on the Google Assistant platform.

The novel approach proposed by Baizal *et al.* [8], in which the interaction in CRS is based on product functional requirements. This CRS utilizes ontology to build interactions (asking questions, providing recommendations and explanations). The results of this study indicate that the system can increase user confidence on the recommendations. However, in this system, users express their needs by choosing options that are provided by the system, thereby reducing the freedom of users to express their needs. Meanwhile, Dalton *et al.* [7] can provide users with very accurate movie recommendations. However, the system does not have a good conversation flow yet. Based on these two problems, we propose a movie CRS that allows users to provide feedback to the system in the form of natural language. We use the tools in Dialogflow to conduct training in recognizing slang in the Indonesian language. In this study, we used the movie database (TMDB) dataset. Currently, no prior works that develop movie CRS in natural languages, especially for the Indonesian language.

The recommender system is able to filter information to overcome information overload [9] by filtering important information and making predictions for the user about the product/information/goods that the user might like. The recommender system is beneficial for service providers and users because it can reduce users' search time for products/goods [10], [11]. In general, there are three approaches to developing recommender systems, such as: collaborative filtering [12], content-based filtering [13], and hybrid filtering [14]. The recommender system has been widely used to provide product recommendations to customers, especially in the e-commerce field.

To ensure the user's flexibility in expressing needs, CRS must be able to interact in natural language. There are several CRS developed, such as chit-chat, informational chat, and task-oriented chat. Chit-chat focuses on human-like interactions or provides attractive responses to users. Informational chat focuses on helping users or answering user questions, while task-oriented chat focuses on helping users complete specific tasks such as making online bookings [15].

A chatbot is a program that can interact with users via text or voice. The chatbot was initially developed using regular expression (regex) to understand word order patterns, which are named by ELIZA [16]. Based on Gartner's prediction [17], the market for chatbots will increase, such that in 2021, more than 50% of companies will focus more on creating chatbots than developing mobile apps. Chatbots allow interaction between humans and computers by using language that humans can understand [4]. Chatbots allow communication between humans and computers, such as communication between humans and each other [6].

2. RESEARCH METHOD

In this study, we use two methods of NLP, i.e., rule-based, and maximum likelihood. Rule-based is a method to categorize a word (give a tag) based on grammar, such as verbs and objects in [18], [19]. The well-known tagger, Brill tagger [20] has been adapted to many languages. In the language model, the maximum likelihood is used to calculate the probability of a word occurring based on the previous word [21]. The language model is represented by N -grams. Some examples of N -grams in language models are [22]:

- Unigram: $P(w1,n)=P(w1) P(w2) \dots P(w_n)$
- Bigram: $P(w1,n)=P(w1) P(w2|w1) \dots P(w_n|w_{n-1})$
- Trigram: $P(w1,n)=P(w1) P(w2|w1) P(w3|w1,2) \dots P(w_n|w_{n-2},n-1)$.

In the unigram model, each word stands alone so that the probability results are based on each word itself. For the bigram model, the probability of words that appear is influenced by the previous word. Meanwhile, in the trigram model, the probability of words that appear is influenced by the two previous words. In this study, we use the bigram method.

2.1. System design

Recently, there have been many tools and platforms for creating chatbots. The development of the chatbot ecosystem allows many features to be released every day. In this study, we used Dialogflow because it currently supports 16 languages, including Indonesian. In addition, the documentation provided is quite complete and supports several messaging platforms such as Line, Telegram, and Slack. We use rule-based, maximum likelihood, and hybrid to train the intent.

The chatbot will attract the user's interest with a few questions as shown in Figure 1. Each interaction between user-messaging apps will be captured as well as processed by the API. After that, the results of the process will be matched with the database available on the webserver. The command results will be forwarded by Dialogflow in the form of a response to the message application in the form of an answer that is passed on to the user. Users interact with the system via the Telegram platform by greeting the chatbot or by giving orders. The text from the user is processed by the Dialogflow API. The text will be matched with the intent created in the Dialogflow through the ngrok webhook, then proceed with the GET Request process on the REST API from TMDB. The results of the *GET* request will be returned to Dialogflow in JSON form and will be displayed to telegram via an intermediary (webhook) ngrok as shown in Figure 2.

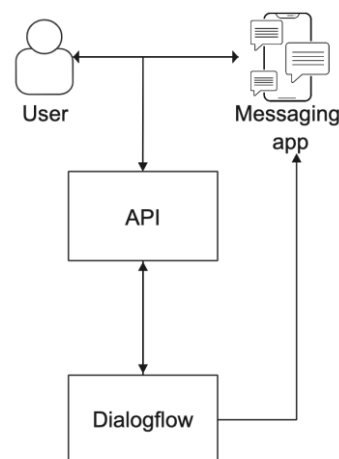


Figure 1. System design

Before the system delivers recommendations, Dialogflow performs natural language processing based on user input. Dialogflow allows us to build a natural language-based conversational user interface. There are two main components in Dialogflow to support this task:

- Agent: acting as a virtual agent for processing conversations with users.
- Intents: categorizing conversations. When the user sends a text, the intent will match the best possible intent in the agent.

The intent matching process is shown in Figure 3. The agent receives user input, then continues with the intent matching process that matches the input from the user with intent, training phrases, as well as action and parameters. The training process is carried out by providing several examples of user input that will be automatically processed by Dialogflow. Figure 4 shows an example of training phrases from users who use slang in the Indonesian language.

To get the preferred genre or movie, Dialogflow provides genre entities as shown in Figure 5. The genre entity aims to match the genre or movie title with the user's text. Dialogflow uses two natural language processing methods, i.e., rule-based, and maximum likelihood. Dialogflow will use rule-based if the intent contains a small number of training phrases, and the maximum likelihood method if the intent has a lot of training phrases.

By training phrases, the system can mark several words as important pieces of information for data extraction. Figure 6 shows a case where the system wants to find out what movie genre the user likes. In the training phrases, we know that after the word genre, the genre variable will be taken, so when the user enters the word “*Saya suka genre action*” (English: “I like the action genre”), the system can retrieve the “action” as a genre variable. Another example is when a user inputs a word that does not contain a genre variable, the system will ask for the movie genre again.

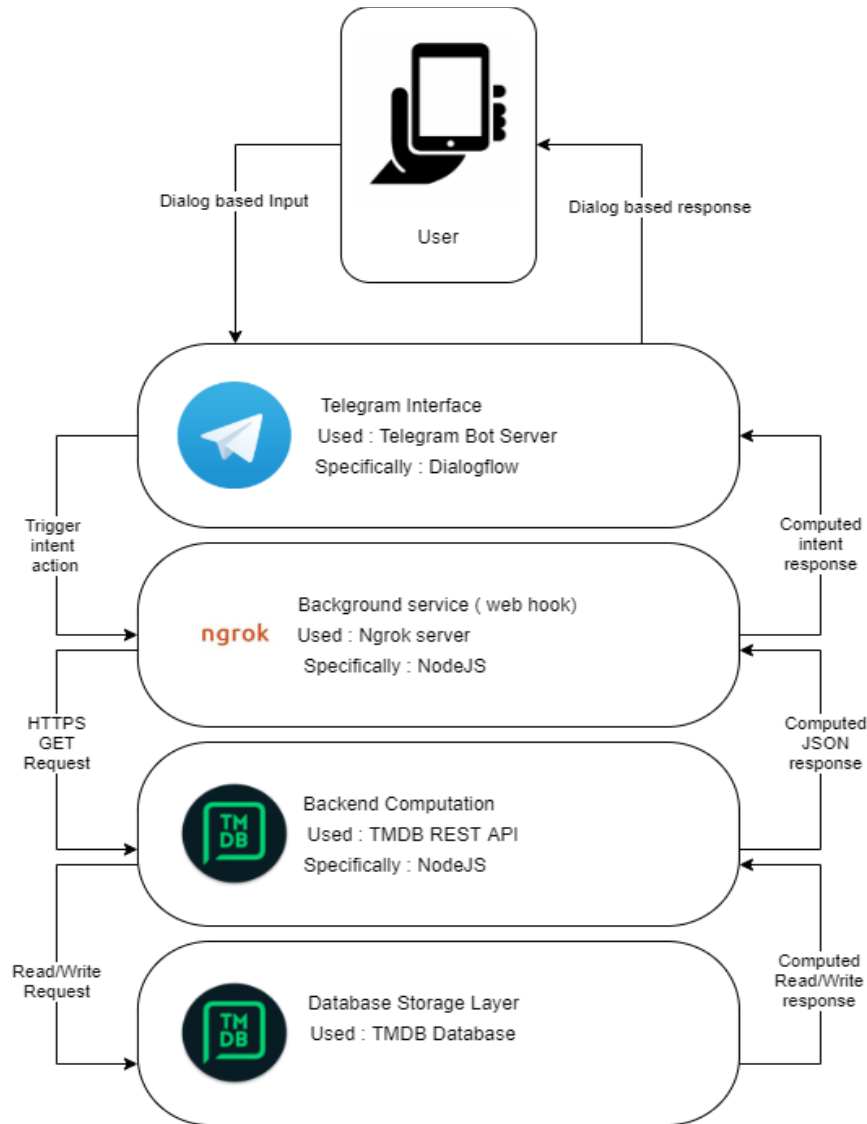


Figure 2. Chatbot architecture

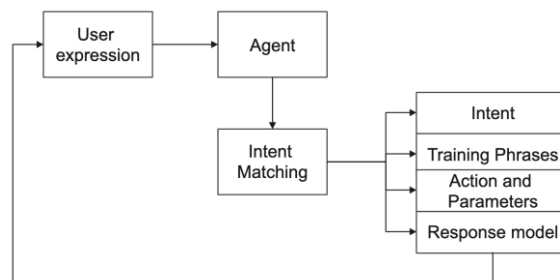


Figure 3. Matching intent process

Training phrases

” Add user expression

” gak tau judul filmnya

” gatau judulnya

” aduh lupa

” aduh lupa judul filmnya

” lupa

” tidak ingat

” aku tidak tau judul filmnya

” aduh saya lupa judul filmnya

” aku tidak ingat

” saya tidak ingat judulnya

Figure 4. The example of training phase from user (in Indonesian language)

genre

SAVE

- Define synonyms Regexp entity Allow automated expansion Fuzzy matching

action	action
horror	horror
drama	drama
scifi	scifi, science fiction, sci fi, sci-fi
romance	romance
documenter	documenter, documentary, dokumenter
adventure	adventure
crime	crime
animation	animation, animasi
comedy	comedy
family	family
fantasy	fantasy
history	history
music	music, musical
mystery	mystery
war	war
western	western

[Click here to edit entry](#)

Figure 5. Example of entity genre

Training phrases Search training phrases

" Add user expression

" action

" ada dong

" ada kok aku suka genre romance

" oh pasti ada dong

" saya suka genre horror

" ada kok

" ada

Action and parameters ^

Enter action name ⋮

REQUIRED ⊕	PARAMETER NAME ⊕	ENTITY ⊕	VALUE	IS LIST ⊕	PROMPTS ⊕
<input checked="" type="checkbox"/>	genre	@genre	Sgenre	<input type="checkbox"/>	Apa genre film ...
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	-

Figure 6. The illustration of system finds out movie genre the user likes

The recommendation process begins by looking for the details of the movie that the user likes. The system looks for similarities with other movie items based on genre, overview, and keywords by combining them. Then the system calculates the term frequency-inverse document frequency (TF-IDF) and similarities (cosine similarity) to other movies in the database. The process of calculating the cosine similarity is carried out by the TMDB API.

2.2. Pre-processing

The system normalizes TF using (1). Let *N* is total of terms in a document, and *term* is the total of appearing of particular term, then normalized TF obtained by (1).

$$Normalized\ TF = \frac{term}{N} \tag{1}$$

After the normalization process, the system continues to calculate the inverse document frequency (IDF). This process is carried out to find the relevance of the document in accordance with the query. The IDF calculation uses (2).

$$IDF(t) = 1 + \log \frac{Total\ number\ of\ document}{Number\ of\ document\ in\ term\ t\ in\ it} \tag{2}$$

For example, there are movies *P* and *Q*. In this case, the user likes movie *P* and then the system will determine the relevance between movie *P* and movie *Q*. In determining this relevance, 3 categories of movie *P* will be used: *genreP*, *overviewP*, and *keywordP*. In addition, 3 categories of movie *Q* are also used: *genreQ*, *overviewQ*, and *keywordQ*. The category used can have more than 1 word, but in this example, only 1 word is used for each category of the movie. We assume that every word in the category of movies *P* and *Q* has been counted and the TF-IDF results are obtained as shown in Table 1.

Based on Table 1, the cosine similarity is obtained by (3)-(7),

$$Cosine_Similarity(Category_P, Category_Q) = \frac{Category_P \cdot Category_Q}{||Category_P|| * ||Category_Q||} \tag{3}$$

$$Category_P \cdot Category_Q = ((0.14) * (0.14) + (0.70) * (0.70) + (0.145 * 0.70)) \tag{4}$$

$$\|Category_P\| = \sqrt{((0.14)^2 + (0.70)^2 + (0.145)^2)} \tag{5}$$

$$\|Category_Q\| = \sqrt{((0.14)^2 + (0.70)^2 + (0.70)^2)} \tag{6}$$

$$Cosine_Similarity(Category_P, Category_Q) = \frac{0.6111}{0.7284 * 0.9997} = 0.83 \tag{7}$$

Table 1. TF-IDF of movies P and Q

Category	TF * IDF
<i>genreP</i>	0.14
<i>overviewP</i>	0.70
<i>keywordP</i>	0.145
<i>genreQ</i>	0.14
<i>overviewQ</i>	0.70
<i>keywordQ</i>	0.70

2.3. The CRS flow

Figure 7 shows the conversation flow in CRS. First, the system asks the preferred movie genre as shown in Figure 8(a), there are two possible answers, i.e.: i) user does not have an idea about the preferred movie title and ii) user has a preferred movie title. For the first one, the system will ask some movie genres. Since the system gets user’s answer, the system will provide the movie title according to the desired genre.

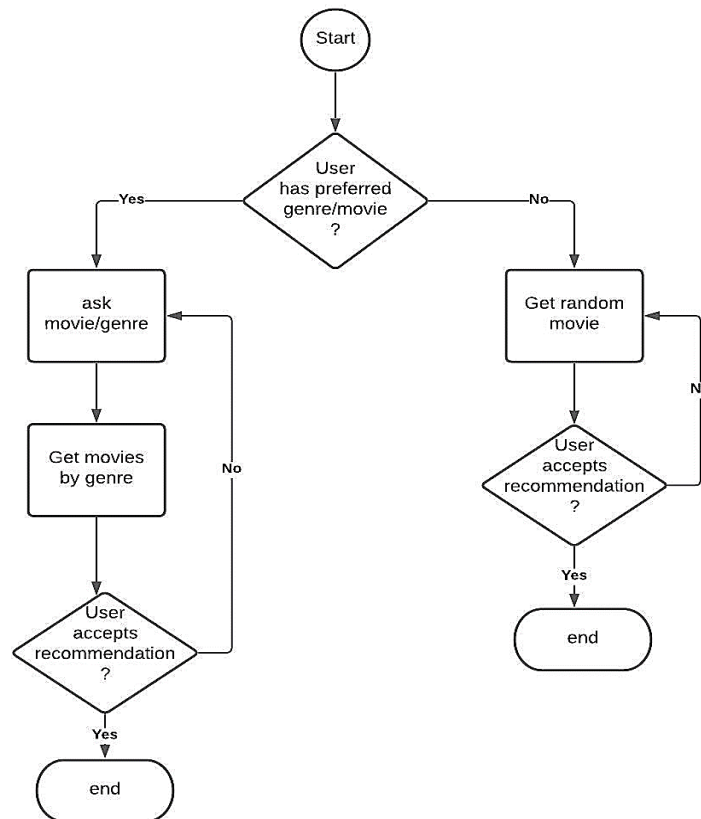


Figure 7. Conversation flow

Since the system provides a movie recommendation and user like this, the system will ask for a maximum of three other movies that are predicted will be preferred by user (the most relevance movies), as shown in Figure 8(b). When user likes the recommended movie and say “*sudah* (enough)”, the system will stop the conversation. Otherwise, the system will ask questions about other movies.

If the user does not have a movie preference, then the system randomly provide movie recommendations as shown in Figure 9. The system will ask again whether the user accepts the recommendation. If user accepts the recommendation, the system will terminate the conversation. Otherwise, the system will recommend another movie (randomly).

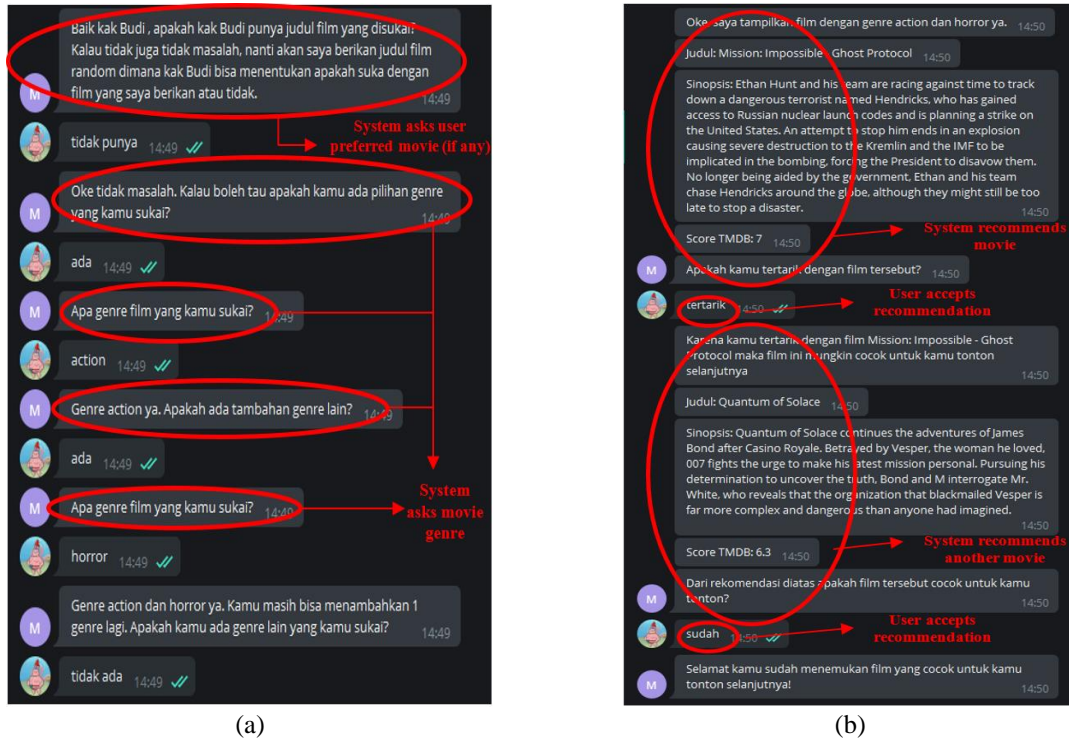


Figure 8. CRS work in (a) asks movie genre from user and (b) recommends movies based on previous interaction as shown in Figure 8(a)

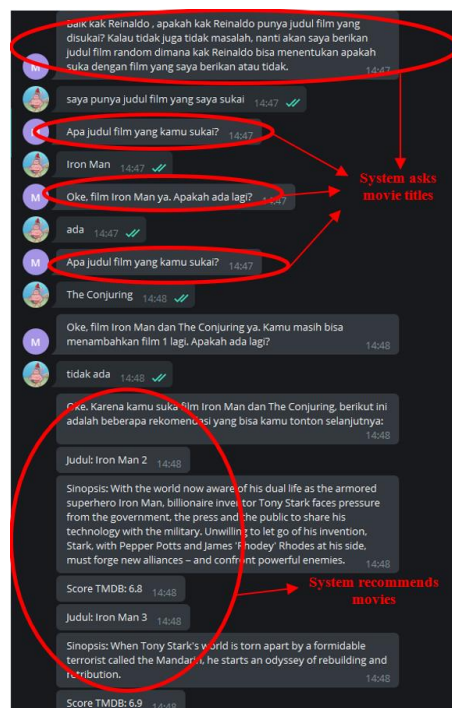


Figure 9. System asks movie titles and recommends movies

3. RESULTS AND DISCUSSION

3.1. Evaluation tools

The focus of CRS is to guide the user to express his needs, and also provide fluid interaction like a natural conversation between the user and a domain expert (or experienced sales assistant) [23]. Therefore, we evaluate this system by considering a user satisfaction survey. User experience is one of the aspects considered in evaluating a CRS [24]–[26]. The online questionnaire was given to 60 respondents who had used CRS. The respondent's ages start from 17 to 22 years, and most of them are students. We choose this range because the users at these ages are considered active users, which still enthusiastic to watch movies, so they are familiar with the digital environment such as chatbot. There are 8 questions in the questionnaire, which observe user experience from some factors. Each question refers to 6 factors based on the evaluation method used by Baizal *et al.* [25], as: i) ease of understanding (EOU), ii) perceived recommendation quality (PRQ), iii) perceived efficiency (PE), iv) informative (INF), v) trust (TR), and vi) easy to use (ETU).

Both P5 and P6 are negative questions as shown in Table 2, the negative questions used to validate the consistency of user respond. User must be able to differentiate between positive sentence and negative sentence during the filling questionnaire, therefore we get a valid answer. The evaluation process actually started when we have hosted the application at our server. In the next step, user will be asked to interact with the chatbot. Finally, user fill the questionnaire based on his experience. Referring to system architecture shown in Figure 2, we build the testing environment divided into two parts, i.e., private service and public service. The private service was provided in on-premises server at the local campus network, including database storage layer element and backend computation. The private service is exposed to the public by using the Ngrok proxy, so the telegram bot-father is able to interact with the backend computation.

3.2. Analysis of the testing result

The user's feedback is shown in Figure 10. Evaluation conducted on 60 respondents showed that the system was able to produce an accuracy of 83% for P8.

$$Score = \frac{\sum Answer(Yes|No)}{\sum Respondent} \quad (8)$$

In (8) shows the score value of respondents' answers, with $\sum Answer(Yes|No)$ being the number of answers Yes or No, and $\sum Respondent$ is the total number of respondents.

Table 2. Questionnaire

ID	Factor	Question
P1	EOU	Can you understand the conversation with the chatbot?
P2	PRQ	Are you satisfied with the results of the proposed movie recommended by the chatbot?
P3	PE	Can you get detailed movie information quickly?
P4	INF	Can you easily get movie information?
P5	ETU	Do you find it difficult to get movie information according to your wishes?
P6	ETU	Do you find it difficult to use the chatbot?
P7	TR	Are you interested in using this chatbot in the future?
P8	TR	Did you manage to find a movie that suits your preferences?

The result of the questionnaire as shown in Figure 10 shows that the proposed system gets a positive response for each question. We notice that the user is quite satisfied when using the system. The results of the easy-to-use factor in questions P5 and P6 show that users feel it is easy to use. This result was supported by the ease of understanding factor in the P1 question with the results of 97% respondents giving a positive response. The results of the P8 indicate that the system can provide accurate recommendations with an accuracy of 83%. Hence, we notice that the results of the system recommendations are quite accurate. However, if there is a movie title that consists of more than one word, the system is sometimes unable to retrieve the complete movie title from user input. This will affect the results of the recommendations that will be given to users. In addition, when user does not know the title of the movie he likes, it would be better if the system also asks for the preferred actor so that the search results for the movie can be more specific.

However, as overall as shown in Figure 10, the questionnaire results of 6 testing factors, i.e., ease of understanding (EOU), perceived recommendation quality (PRQ), perceived efficiency (PE), informative (INF), trust (TR), and easy to use (ETU) show that the average yield obtained is above 80%. These results indicate that overall, the system built is able to provide a positive experience for users.

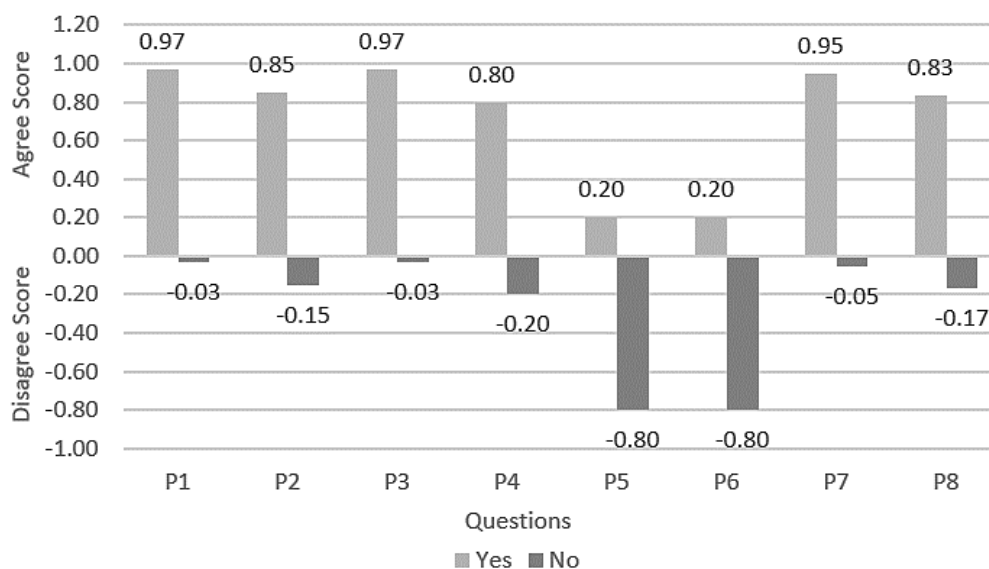


Figure 10. The questionnaire results

4. CONCLUSION

In this study, the proposed NLP-based CRS has an accuracy of 83% (high enough) in recommending movies. This accuracy value can change depending on the movie information contained in the system. In addition, the results of the questionnaire showed good results for the 6 factors asked: EOU, PRQ, PE, INF, TR, and ETU. The evaluation result shows that the proposed CRS can provide a good user experience in recommending movies because users feel flexible in expressing their needs.




REFERENCES

- [1] N. Koubai and F. M. Bouyakoub, "MyRestaurant: a smart restaurant with a recommendation system," *International Journal of Computing and Digital Systems*, vol. 8, no. 2, pp. 143–156, Jul. 2019, doi: 10.12785/ijcds/080206.
- [2] B. T. Choudhary and B. Tulasi, "Recommender system for personalised travel itinerary," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 5, pp. 4460–4465, Oct. 2019, doi: 10.11591/ijece.v9i5.pp4460-4465.
- [3] Z. K. A. Baizal, K. M. Lhaksmana, A. A. Rahmawati, M. Kirom, and Z. Mubarok, "Travel route scheduling based on user's preferences using simulated annealing," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 2, pp. 1275–1287, Apr. 2019, doi: 10.11591/ijece.v9i2.pp1275-1287.
- [4] I. Nica, O. A. Tazl, and F. Wotawa, "Chatbot-based tourist recommendations using model-based reasoning," in *CEUR Workshop Proceedings*, 2018, vol. 2220, pp. 25–30.
- [5] F. Colace, M. De Santo, F. Pascale, S. Lemma, and M. Lombardi, "BotWheels: a Petri Net based Chatbot for recommending tires," in *Proceedings of the 6th International Conference on Data Science, Technology and Applications*, 2017, pp. 350–358, doi: 10.5220/0006491903500358.
- [6] L. Vaira, M. A. Bochicchio, M. Conte, F. M. Casaluci, and A. Melpignano, "MamaBot: a system based on ML and NLP for supporting women and families during pregnancy," in *22nd International Database Engineering and Applications Symposium on*, 2018, pp. 273–277, doi: 10.1145/3216122.3216173.
- [7] J. Dalton, V. Ajayi, and R. Main, "Vote goat: conversational movie recommendation," in *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jun. 2018, pp. 1285–1288, doi: 10.1145/3209978.3210168.
- [8] Z. K. A. Baizal, D. H. Widyanoro, and N. U. Maulidevi, "Computational model for generating interactions in conversational recommender system based on product functional requirements," *Data and Knowledge Engineering*, vol. 128, Jul. 2020, doi: 10.1016/j.datak.2020.101813.
- [9] L. Ravi, V. Subramaniaswamy, V. Vijayakumar, S. Chen, A. Karmel, and M. Devarajan, "Hybrid location-based recommender system for mobility and travel planning," *Mobile Networks and Applications*, vol. 24, no. 4, pp. 1226–1239, Apr. 2019, doi: 10.1007/s11036-019-01260-4.
- [10] M. Li, Y. Li, W. Lou, and L. Chen, "A hybrid recommendation system for Q&A documents," *Expert Systems with Applications*, vol. 144, Apr. 2020, doi: 10.1016/j.eswa.2019.113088.
- [11] E. Tang, "A quantum-inspired classical algorithm for recommendation systems," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, Jun. 2019, pp. 217–228, doi: 10.1145/3313276.3316310.
- [12] H. Gao, Y. Xu, Y. Yin, W. Zhang, R. Li, and X. Wang, "Context-aware QoS prediction with neural collaborative filtering for internet-of-things services," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4532–4542, May 2020, doi: 10.1109/JIOT.2019.2956827.
- [13] S. Reddy, S. Nalluri, S. Kuniseti, S. Ashok, and B. Venkatesh, "Content-based movie recommendation system using genre correlation," in *Smart Innovation, Systems and Technologies*, vol. 105, Springer Singapore, 2019, pp. 391–397, doi: 10.1007/978-981-13-1927-3_42.
- [14] J. Bobadilla, S. Alonso, and A. Hernando, "Deep learning architecture for collaborative filtering recommender systems,"




- Applied Sciences*, vol. 10, no. 7, Apr. 2020, doi: 10.3390/app10072441.
- [15] Y. Sun and Y. Zhang, "Conversational recommender system," in *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jun. 2018, pp. 235–244, doi: 10.1145/3209978.3210002.
- [16] J. Weizenbaum, "ELIZA — a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 26, no. 1, pp. 23–28, Jan. 1983, doi: 10.1145/357980.357991.
- [17] K. Panetta, "Garnet top strategic predictions 2018 and beyond," *Gartner*, 2018.
- [18] K. K. Purnamasari and I. S. Suwardi, "Rule-based part of speech tagger for Indonesian language," *IOP Conference Series: Materials Science and Engineering*, vol. 407, no. 1, Sep. 2018, doi: 10.1088/1757-899X/407/1/012151.
- [19] F. Zhang, H. Fleyeh, X. Wang, and M. Lu, "Construction site accident analysis using text mining and natural language processing techniques," *Automation in Construction*, vol. 99, pp. 238–248, Mar. 2019, doi: 10.1016/j.autcon.2018.12.016.
- [20] M. Al-Smadi, S. Al-Zboon, Y. Jararweh, and P. Juola, "Transfer learning for arabic named entity recognition with deep neural networks," *IEEE Access*, vol. 8, pp. 37736–37745, 2020, doi: 10.1109/ACCESS.2020.2973319.
- [21] R. J. Rossi, *Mathematical statistics: an introduction to likelihood based inference*. John Wiley & Sons, 2018, doi: 10.1002/9781118771075.
- [22] V. Lavrenko and W. B. Croft, "Relevance-based language models," *ACM SIGIR Forum*, vol. 51, no. 2, pp. 260–267, Aug. 2017, doi: 10.1145/3130348.3130376.
- [23] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender systems: an introduction*. Cambridge University Press, 2010, doi: 10.1017/CBO9780511763113.
- [24] A. H. Afridi, "Transparency for beyond-accuracy experiences: a novel user interface for recommender systems," *Procedia Computer Science*, vol. 151, pp. 335–344, 2019, doi: 10.1016/j.procs.2019.04.047.
- [25] Z. K. Abdurahman Baizal, Y. R. Murti, and Adiwijaya, "Evaluating functional requirements-based compound critiquing on conversational recommender system," *5th International Conference on Information and Communication Technology*, May 2017, doi: 10.1109/ICoICT.2017.8074656.
- [26] D. Biduski, E. A. Bellei, J. P. M. Rodriguez, L. A. M. Zaina, and A. C. B. De Marchi, "Assessing long-term user experience on a mobile health application through an in-app embedded conversation-based questionnaire," *Computers in Human Behavior*, vol. 104, Mar. 2020, doi: 10.1016/j.chb.2019.106169.

BIOGRAPHIES OF AUTHORS






Zinke Abdurahman Baizal    received the B.S. degree from the Gadjah Mada University, Yogyakarta, Indonesia, in 1998. He joined the School of Computing at Telkom University in 1999. He received an M.S. degree in computer science from Gadjah Mada University in 2003. He received a Ph.D. degree in electrical engineering and informatics at Institut Teknologi Bandung (ITB), Indonesia. He is currently a dean in the School of Computing at Telkom University. He is an author of a recommender system book, more than 26 articles, and 6 inventions. His research interests include recommender systems, data mining, artificial intelligence, semantic web as well as information extraction. He can be contacted at baizal@telkomuniversity.ac.id.






Nurul Ikhsan    received the B.S. degree in Physics from Institut Teknologi Bandung, Indonesia, in 2010. He received an M.S. degree in Computational Science from Institut Teknologi Bandung in 2013. He joined the School of Computing at Telkom University in 2014. He received a Ph.D. degree in Physical and Mathematical Science at Kanazawa University, Japan. He is currently a lecturer and researcher in the School of Computing at Telkom University. His research interests include modelling and simulation, quantum computing, and artificial intelligence. He can be contacted at email: ikhsan@telkomuniversity.ac.id.






Ichwanul Muslim Karo Karo    received B.Sc. from Computational Science, Telkom University, Indonesia, in 2015. He achieved the M.Kom. degree in School of Computing, Telkom University 2017. He is a Computer Science lecturer in State University of Medan. In addition, He is an entrepreneur as owner @konveksita. His research interests include data mining, spatial mining, and business intelligence. He can be contacted at email: ichwanul@unimed.ac.id.



Reinaldo Kenneth Darmawan    received B.Sc. from Computational Science, Telkom University, Indonesia, in 2020. Since 2020, he works as an engineer at Ninjavan, Pte Ltd, Singapore. His research interests are recommender system, natural language processing, and conversational recommender system. He can be contacted at email: reinaldo.kenneth@gmail.com.



Roby Dwi Hartanto    received B.Sc. from information system, Telkom University, Indonesia, in 2019. Since 2019, he has worked as an engineer at Bandung Techno Park Indonesia, His research interests are Recommender systems and conversational recommender systems. He can be contacted at robby.dh596@gmail.com.