

Towards a Redundant Response Avoidance for Intelligent Chatbot

Hyuck-Moo Gwon* and Yeong-Seok Seo**

Abstract

Smartphones are one of the most widely used mobile devices allowing users to communicate with each other. With the development of mobile apps, many companies now provide various services for their customers by studying interactive systems in the form of mobile messengers for business marketing and commercial promotion. Such interactive systems are called “chatbots.” In this paper, we propose a method of avoiding the redundant responses of chatbots, according to the utterances entered by the user. In addition, the redundant patterns of chatbot responses are classified into three categories for the first time. In order to verify the proposed method, a chatbot is implemented using Telegram, an open source messenger. By comparing the proposed method with an existent method for each pattern, it is confirmed that the proposed method significantly improves the redundancy avoidance rate. Furthermore, response performance and variation analysis of the proposed method are investigated in our experiment.

Keywords

Chatbot, Human-Computer Interaction, Interactive System, Redundancy Avoidance, Telegram

1. Introduction

As the past PC-based era has evolved into the smartphone-based mobile era, it is now possible to exchange and share various forms of information by accessing the Internet anytime and anywhere. According to such changes in technology trends, messenger apps are becoming a new platform from which to launch other apps in the mobile era as portal sites played a platform role in the PC-based era [1-4].

The most familiar services for smartphone users are mobile messengers. It is not surprising now that parents and children in the same home are having a conversation using messengers instead of talking face-to-face. Such messenger services are rapidly evolving into a form of a “chatbot” in conjunction with commercial platforms to promote the convenience of customers [5,6]. A chatbot can be defined as a non-human virtual conversation partner or a conversation robot, which can answer the user’s questions through a chat interface according to the predetermined response rules in frequently used messengers [7]. Chatbots are interactive systems that are common in modern daily life. Although they were initially developed for psychotherapy, they have been actively advanced for purposes such as information

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Manuscript received January 29, 2019; first revision December 26, 2019; accepted February 4, 2020.

Corresponding Author: Yeong-Seok Seo (ysseo@yu.ac.kr)

* Dept. of Computer Science, Hanyang University, Seoul, Korea (howling6@agape.hanyang.ac.kr)

** Dept. of Computer Engineering, Yeungnam University, Gyeongsan, (ysseo@yu.ac.kr)

An early version of this paper was presented at CSA2018. In this paper, we technically extend and analyze the proposed method, investigate existing studies in detail, and perform additional experiments.

guidance, Q&A, and entertainment as a means to provide services to customers for many companies and organizations [8].

In the past PC-based era, there were PC-based messengers (e.g., MSN messenger [9]) to which chatbot technology could be applied. “PC messengers,” however, were login-based services and thus there was a limitation in that they could not be used any longer after logout. In the mobile era, on the other hand, the “mobile messengers” in smartphones are well suited to be used as channels for business marketing and commerce because they remain connected. In other words, these most familiar communication tools in everyday life are conveniently used for a variety of purposes ranging from online shopping to product orders, accommodation reservation, and bank loan consultation. The CEOs of global social networking service (SNS) leaders are calling intelligent chatbots the future of SNS, and many studies are being conducted on intelligent chatbot technologies in the IT industry [10,11].

One of the key issues in the research on intelligent chatbot technologies is whether the conversation partner can provide flexible and versatile responses like a human to specific utterances that the users send to the chatbot service. In other words, it is necessary to provide smart responses to meet the needs of the users even if the utterances sent by them are overlapping. To date, however, when users continuously send the same utterances to chatbot services, most of the services provide the same redundant responses according to the simple intentions of the chatbot service developers. This lowers the trust of the users in chatbot services, reduces their interest, and causes a problem in that only limited information is provided.

In this paper, we propose a redundant response avoidance method that can provide a larger variety of responses when the user continuously inputs the same utterance in a chatbot service. In the proposed method, when the user inputs and sends an utterance to a chatbot service, first, a morpheme analysis is conducted on the utterance. A major “keyword” in the utterance is then selected, and one of the various responses related to the keyword, which are stored in the database (DB), is selected. To address the problem of redundancy, one of the various response examples is selected not through a simple random selection method applied in the existing studies but through an initially proposed detailed mathematical algorithm, thereby significantly reducing the rate of redundant responses. In this study, the redundancy patterns of chatbot responses, according to the user utterances, are classified into three types. For each pattern, the proposed method is applied and verified in detail.

The contributions of this study are as follows:

- A redundant response avoidance method is proposed, to improve the response diversity of an intelligent chatbot;
- The redundancy patterns of chatbot responses are identified and classified into three types;
- The performance of the redundant response avoidance is quantitatively validated for each redundancy pattern;
- The response times of the methods are compared and analyzed, by measuring the operation (execution) time for the duplicate avoidance operations over 1,000,000 iterations;
- In addition to the proposed method, a variation of the proposed method is suggested by considering the characteristics and features of prime numbers.

The remainder of this paper is organized as follows. In Section 2, related work is examined. In Section 3, the background knowledge required in this paper is explained. In Section 4, the classification of redundancy patterns proposed in this study and the method for avoiding redundancy are proposed. In Section 5, the experiment design for verifying the proposed method and the results of comparison with the existing methods are provided. Finally, in Section 6, the conclusion and future studies are described.

2. Related Work

2.1 Chatbot Response 1: Combined Keyword-Based Type

There are various methods for implementing a chatbot. One of them is to generate responses by finding keywords from the utterance entered by the user. This method was applied by ELIZA, which can be called the first chatting robot [12]. It was developed by Professor Joseph Weizenbaum of the MIT Artificial Intelligence Laboratory for Rogerian psychotherapy. Rogerian psychotherapy is a psychotherapy method that induces a conversation by creating questions from the contents of the conversation with the counselee and encouraging the counselee. ELIZA also used this method. This method induces a conversation from the counselee using the trick of changing “Me” or “I” from the utterance of the counselee into “You,” and “My” into “Your,” as well as asking about the keywords again [12].

ELIZA was a chatbot that performed simple pattern matching such as telling the counselee to refrain from using unpleasant words when the counselee enters such words by using a simple matching method in addition to the above-mentioned trick. When Professor Weizenbaum distributed ELIZA [12] to the college staff, however, their reliance on ELIZA increased and there was a great deal of enthusiasm. Fig. 1 shows an actual conversation of ELIZA. The above-mentioned trick can be easily understood through the contents.

```

> Hello, I am Eliza.
* Hey!

> Don't you ever say Hello?
* Hello!

> How are you today. What would you like to discuss?
* I'm depressed.

> Do you enjoy being depressed?
* No, I don't.

> Why?
* Because I want to be happy.

> Why do you want to be happy?
* Because I'm tired of being depressed.

> That is quite interesting.
* I'm not sure.

```

Fig. 1. Example of a conversation of ELIZA.

It can be seen that when the counselee(*) said “I’m depressed,” ELIZA made a sentence by changing “I” into “you” and using the keyword “depressed.” In addition, as ELIZA used simple pattern matching, it had a limitation of providing the same response whether the counselee entered “Hello” or “Hello Eliza” [13].

2.2 Chatbot Response 2: AIML-Based Type

Artificial intelligence markup language (AIML) is an XML-based scripting language that randomly selects one of various responses that are prepared through the normalization of the user’s utterances when

the chatbot is implemented. It was developed by Dr. Richard Wallace and its origin was an interactive engine called Artificial Linguistic Internet Computer Entity (ALICE). ALICE was a chatbot with excellent performance, winning the Loebner Prize, the most acclaimed annual award in the area of artificial intelligence, twice [14,15].

AIML, the script language, focused on the fact that daily conversations consist of limited sentences. It responds to the user by determining whether the user's utterance matches the <pattern> through the normalization of the utterance and randomly determining a response from the conversation set called <template> as a response to the <pattern>. Fig. 2 is an example of AIML. When “Hello” is entered as <pattern>, one of “How are you?” “Nice to meet you!” and “Hello~” is randomly answered from <template>. In the case of the ALICE method, as a random response is selected from <template> in response to a <pattern>, there is a problem that redundant responses can be delivered to the user if the same <pattern> of the user utterance is entered.

```

<category>
  <pattern>Hello</pattern>
  <template>
    <random>
      <li>How are you?</li>
      <li>Nice to meet you!</li>
      <li>Hello~</li>
    </random>
  </template>
</category>

```

Fig. 2. Example of a normalization of AIML.

2.3 Chatbot response 3: Database-based type

Chatbots do not respond by understanding and thinking about the utterances of the user. They respond by finding a proper response among the responses stored in the database according to rules such as patterns or keywords set by the developer. This process is mainly performed using pattern matching. The easiest pattern matching is the slot-and-frame parser method. As shown in Table 1, the slot-and-frame parser method has a table that consists of input sentences and response sentences [16,17].

Table 1. Example of a slot-and-frame parser table

Phrase	Response
HELLO	Hi, there
What are you doing?	I'm thinking
Weather is nice today	Yes, very nice
Do you like coffee?	I can't drink it
Let's talk	On what?

As shown in Table 1, when the user inputs “HELLO,” the chatbot replies “Hi, there” as a response. For the slot-and-frame parser method, the implementation is easy and the conversation level of the chatbot improves as the data in the table increases. However, a response can be obtained only when the sentence

entered by the user matches that in the table, and more time is required to search the contents of the table as the table is expanded.

Although many chatbot-related studies are currently being conducted, if the responses to the user's overlapping utterances are generated by a simple random method, a 1:1 matching method, or a response generating method, the responses will continue to overlap over time. This causes a problem in the flexible conversations of chatbots. In other words, it is not currently considered how the responses to the user's overlapping utterances are handled. If such a problem is not addressed, there will be limitations in providing sufficient information to the user through a chatbot.

3. Background

3.1 Telegram

The chatbot implemented in this paper used Telegram. Telegram is an open-source messenger [18] and is shown in Fig. 3. It was developed by the brothers Nikolai Durov and Pavel Durov, and is now being developed and managed by the German Telegram Messenger LLP. As of March 2018, the number of active monthly Telegram users exceeded 200 million [19].

Telegram has created and released an application programming interface (API) for every user to implement chatbots. Therefore, if a user wants to create a chatbot, he or she can implement a chatbot on the Telegram app by downloading the API and creating a program that specifies a response according to the utterance of the user using a programming language such as C# and Python.

In this study, a chatbot was implemented using the API of Telegram [20], followed by an experimental design and results analysis.

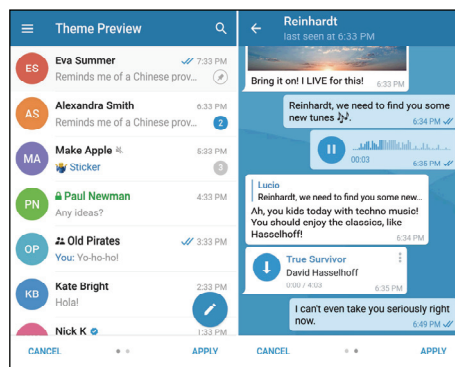


Fig. 3. Telegram messenger. Adapted from [18].

4. Redundancy Pattern and Avoidance Method

4.1 Classification of Redundancy Patterns

In this study, redundancy patterns are identified first according to the response results as shown in Table 2. In Pattern 1, although the chatbot has various responses, it delivers a redundant response when

the user inputs an overlapping utterance. In Pattern 2, the chatbot delivers a different response first when the user inputs an overlapping utterance, but later it repeats the previous response for an additional overlapping utterance. In Pattern 3, even though the user utterances are different, redundant responses are delivered because the matching keywords for the utterances are the same.

All of the patterns in Table 2 can be derived by the existing methods mentioned in Section 2 when there are various matching results according to the user utterance. Therefore, in this paper, we develop a method to avoid redundancy using a mathematical algorithm, instead of the existing selection methods when a chatbot has various responses for the user utterance.

Table 2. Classification of the redundancy patterns of chatbots

		Example
Pattern 1	<Chatbot> Hi, There <Chatbot> Hi, There	<User> Hi <User> Hi
A redundant response occurs when the user inputs an identical utterance.		
Pattern 2	<Chatbot> Hi, There <Chatbot> Hello <Chatbot> Hi, There	<User> Hi <User> Hi <User> Hi
A different response is delivered first for an identical utterance from the user, but later a redundant response occurs for an additional identical utterance.		
Pattern 3	<Chatbot> Hi, There <Chatbot> Morning! <Chatbot> Hi, There	<User> Hi, Nice meet you <User> Hi, Good Morning <User> Hi, Chatbot
A redundant response occurs for a different utterance from the user.		

4.2 Method for Redundancy Avoidance

To avoid redundancy patterns in Table 2, it is necessary to first identify which of the responses in the DB is the chatbot's previous response for the user's overlapping utterance. Therefore, in this paper, a variable is declared in the chatbot program; a reference to the response previously chosen by the chatbot among the responses in the DB is stored in the variable. The variable is named as *temp*. The following Eq. (1) is a method of obtaining the value that determines which of the responses in the DB will be delivered to the user.

$$temp_k = (temp_{k-1} + (R + 1)) \% N \quad (1)$$

where N is the number of responses retrieved from the DB corresponding to the keyword extracted from the user utterance, and R is a random number. In the case of R , the “+1” operation is used to prevent that Eq. (1) is to be in the form of $temp_k = temp_{k-1} \% N$ when R is randomly selected as 0. The equation causes the same response as the previous response of the chatbot. In addition, the result of the $(temp_{k-1} + (R + 1))$ operation is divided by N , and the remaining value is calculated so that a value higher than N could not be obtained. Thus, if the $temp_k$ -th response in the DB is delivered as an actual response through the above equation, redundant responses encountered by the existing studies can be avoided because the N responses are continuously changed according to the R value. In consideration of redundancy patterns, the range of R needs to be set carefully not to deliver the same $temp_k$ -th response consecutively again after $temp_k$ -th response is delivered, to achieve stable redundancy avoidance.

Redundant responses can be avoided, especially in Pattern 1, even if R is a specific number smaller than N instead of a random number. However, this leads that the chatbot will deliver replies in a predetermined order. This will lower the interest of the user and act as a major drawback for the chatbot whose aim is to appear as a form of highly-advanced artificial intelligence. Thus, the random number R with a range is applied.

Fig. 4 shows a case in which the user inputs the same utterance three times in a row. The range of R is assumed to be between 0 and 2 for a simple and clear description. As can be seen in Fig. 4, when “Hello” is entered first by the user, the temp value started from 0. In addition, “Response Set” is retrieved from the DB corresponding to the keyword (“Hello”) extracted from the user utterance. Then, “Nice to meet you” from the DB’s response set is delivered. When “Hello” is entered again, R is randomly determined as 2, and the temp value is calculated as 3 through the equation, and thus “Hi ~”, the third response in the DB’s response set, is delivered. When “Hello” is entered again, R is determined as 1, and the temp value is calculated as 5 through the equation, and thus “Hello!”, the fifth response in the DB’s response set, is delivered. Even though the user enters an overlapping utterance after the chatbot responded, the chatbot provides the response that does not overlap with the previous one through Eq. (1).

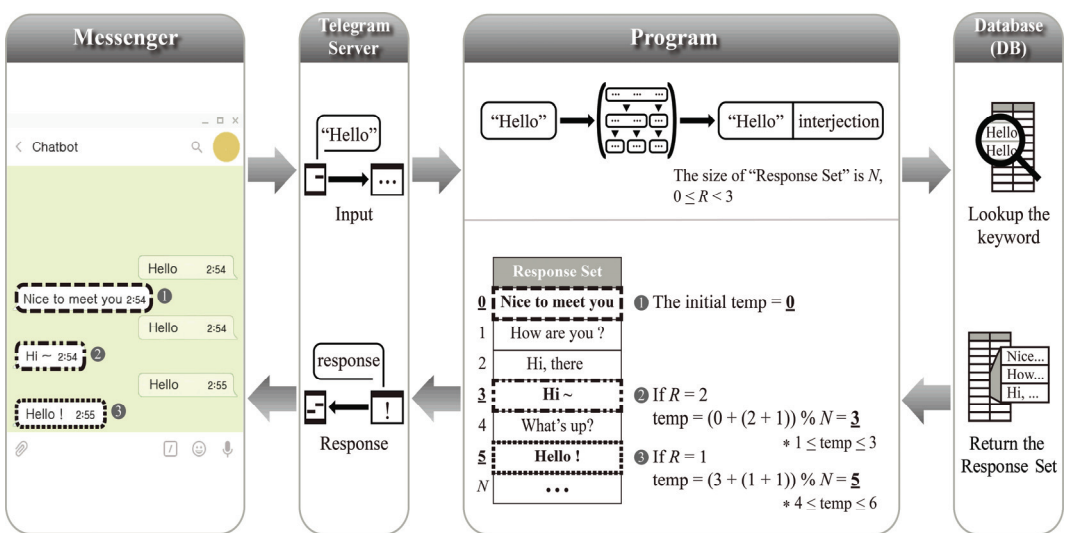


Fig. 4. Overall operation process

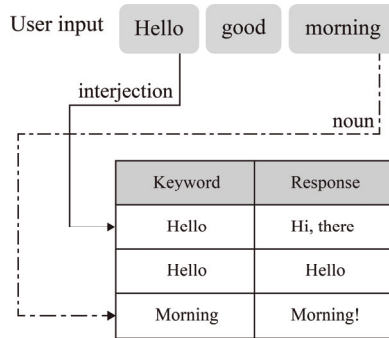


Fig. 5. User input analysis.

Response Set is retrieved by keyword matching as follows: The primary words obtained through the morpheme analysis of the user utterance are used as keywords. As can be seen from Fig. 5, for example, when the user utterance is “Hello good morning,” the interjection “Hello,” adjective “good,” and noun “morning” are identified. After analyzing them and matching the responses related to each keyword in the DB, an actual response is selected using the proposed equation.

5. Experiment

5.1 Experiment Design

In order to verify the proposed method, a chatbot was implemented as shown in Fig. 4. After importing the utterance entered by the user into the chatbot program through Telegram [19], an open-source messenger, a morpheme analysis is conducted. Based on the derived keyword, related responses are retrieved from the DB. A final response is selected using the proposed equation and then delivered to the user.

For a comparison with the proposed method, the existing ALICE method was used. ALICE is an interactive engine based on AIML that randomly selects one of various responses that are prepared through the normalization of the user’s utterances. It is widely known as a chatbot with excellent performance [14,15]. For the experiment of this study, responses for each keyword were implemented using a DB.

The average redundancy probabilities were obtained to compare and analyze the methods. The experiment generated 100 random numbers for the comparison (message conversations 100 times with each chatbot). The average probabilities were obtained by repeating the same experiment 1,000 times.

The experiments were performed with an Intel i5-4690 3.5 GHz, 16 GB RAM, and 128 GB SSD. In the experiment, the C# Random Class was used to generate random numbers for the methods. It returns a non-negative random integer that is less than the specified maximum. If the range to obtain a random number is set as 0 to Z, Z is not included to generate it.

5.2 Comparison for Redundancy Pattern 1

Fig. 6 shows the experiment results for Pattern 1 (P1). The x-axis of the figure represents N and the Y-axis of the figure indicates the redundancy probability.

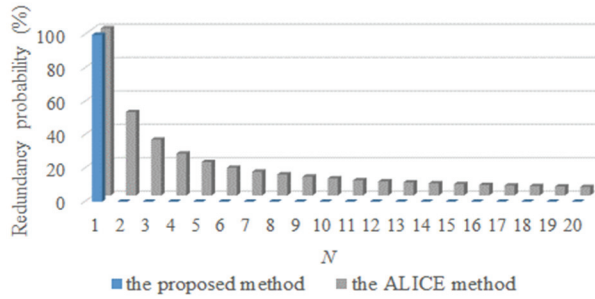


Fig. 6. Comparison of redundancy probabilities for P1 ($0 \leq R < N/2$).

In order to test how many redundant responses occur in the ALICE method, N pairs of “keywords and chatbot responses” are stored in a DB so that a response could be selected according to the “keyword” of the user utterance. In the case of the ALICE method, the k -th chatbot response is randomly provided according to the keyword among N . In the case of the proposed method, the range of R is set as $0 \leq R < N/2$ to avoid P1 for all N except $N=1$.

As shown in Fig. 6, it is confirmed that there is no redundancy if N is 2 or higher when the proposed method is applied. This is because the $temp$ value of Eq. (1) is continuously changed according to R and this leads that the response of the chatbot does not overlap with the previous one. The redundancy probability of the ALICE method is inversely proportional to N .

5.3 Comparison for Redundancy Pattern 2

Fig. 7 shows the experiment results for Pattern 2 (P2). Fig. 7 has the same format as Fig. 6. As presented in Fig. 7, the ALICE method shows the similar trend compared with the results of P1. However, the proposed method provides the higher redundancy probability than that of P1, with the same range of R . In particular, the redundancy probability of the proposed method is much higher than that of the ALICE method when N is 2. This is because R is determined only as 0 when N is 2 in the proposed method, and thus a response of the chatbot always different from the previous one is selected. This causes the redundancy of P2. Accordingly, when N is 3 or higher, it is confirmed that the redundancy gradually decreases for P2.

Note that when the range of R is $0 \leq R < N/2$, all N are even numbers for the redundancy occurrence in P2. This is because, as described in Fig. 8, when N is 10 and R is 4 (when the random number generated in the range of R is 4), the $temp$ value can be repeated as 0 and 5, and thus the responses mapped to the $temp$ values of 0 and 5 can appear again alternately.

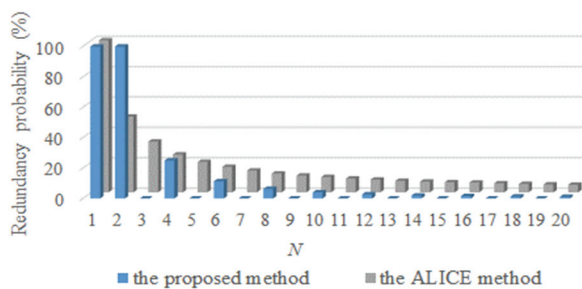


Fig. 7. Comparison of redundancy probabilities for P2 ($0 \leq R < N/2$).

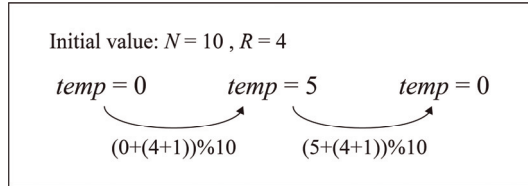


Fig. 8. Example of the redundancy occurrence in P2.

When the range of R is reduced, this issue can be prevented. Unlike Fig. 7, Fig. 9 is a graph showing the results when the range of R is reduced to $N/3$. As provided in Fig. 9, the proposed method confirms that there is no redundancy if N is 3 or higher.

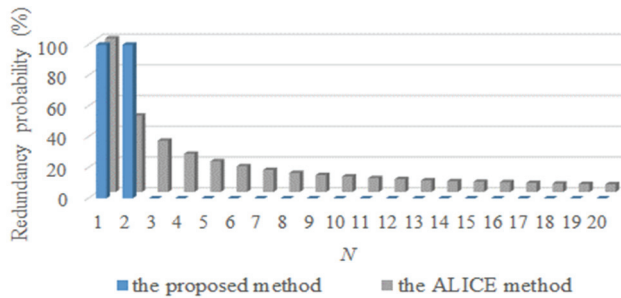


Fig. 9. Comparison of redundancy probabilities for P2 ($0 \leq R < N/3$).

5.4 Comparison for Redundancy Pattern 3

In case of Pattern 3 (P3), a redundant response of the chatbot occurs for a different utterance from the user. Thus, it is necessary to consider the number of keywords extracted from the user utterance by the morpheme analysis (M) and the number of the responses retrieved from the DB related to each keyword (N). In order to verify P3, the experiment results according to the changes of M and N are derived and analyzed. It is inferred that the redundancy probability of the chatbot is inversely proportional to $M * N$ as shown in Eq. (2).

$$Redundancy\ probability \propto \frac{1}{M * N} \quad (2)$$

Fig. 10 shows the experiment results according to the change of M and N when the ALICE method is used for P3. When the number of keywords extracted from the user utterance is M , a keyword can be also selected randomly, like N . As can be seen from Fig. 10, the redundancy probability significantly decreases as M increases. In addition, as N increases, the redundancy probability tends to gradually decrease.

Figs. 11 and 12 are graphs showing the redundancy probabilities for the ALICE method and the proposed method according to the change of N when M is 2 and 3, respectively. Figs. 11 and 12 have the same formats as the previous figures. The maximum probabilities of the y-axis are reduced compared with the results of P1 and P2. As mentioned in the analysis of P2, the redundancy probability of the proposed method tends to be similar to that of the ALICE method when N is an even number owing to the case shown in Fig. 8.

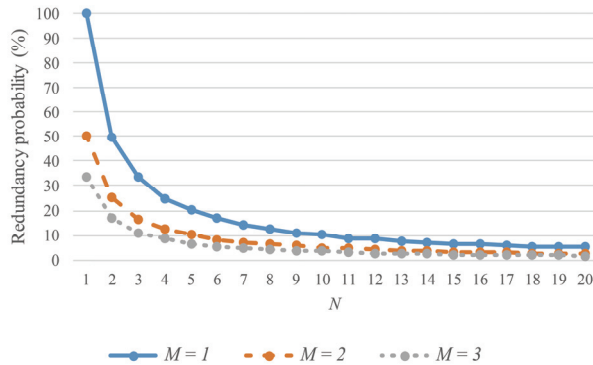


Fig. 10. Redundancy probabilities in the variation of M and N.

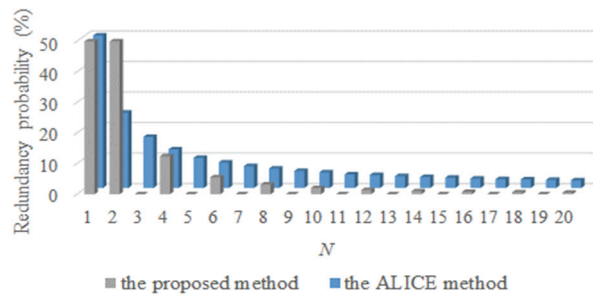


Fig. 11. Comparison of redundancy probabilities for P3 (M = 2, 0 ≤ R < N/2).

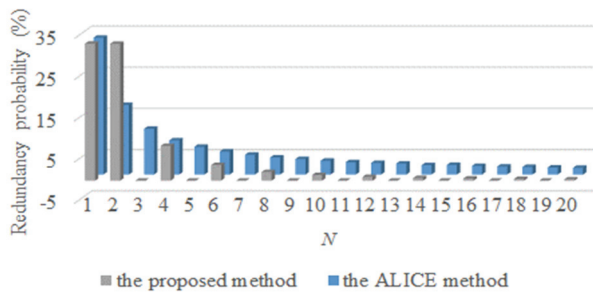


Fig. 12. Comparison of redundancy probabilities for P3 (M = 3, 0 ≤ R < N/2).

As these are the results when the range of R is set to N/2 in Eq. (1), it is judged that a clearer redundancy avoidance tendency can be derived if the range of R is reduced as in P2.

Figs. 13 and 14 show the results when the range of R is changed from N/2 to N/3. As presented in Figs. 11–14, when the range of R is reduced from N/2 to N/3, a clearer redundancy avoidance tendency is confirmed according to the principle mentioned for P2. As provided in Figs. 13 and 14, the proposed method shows that there is no redundancy if N is 3 or higher.

Generally, in the methods to avoid redundant responses, if N that can be retrieved from the DB for each keyword is too small, the application of the methods cannot guarantee the prevention of pattern generation or redundancy in the responses of the chatbot. Therefore, it is necessary to store a sufficient number of chatbot responses to use the methods.

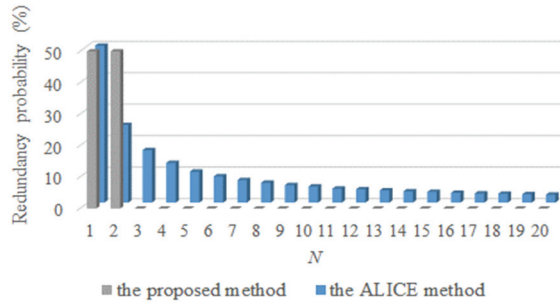


Fig. 13. Comparison of redundancy probabilities for P3 ($M = 2$, $0 \leq R < N/3$).

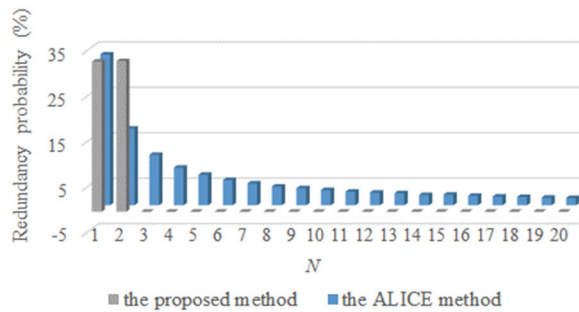


Fig. 14. Comparison of redundancy probabilities for P3 ($M = 3$, $0 \leq R < N/3$).

5.5 Response Performance of Chatbot

Generally, when building chatbots, we can require that the chatbot provide a different response from the previous one if the response corresponding to the user input is duplicated. That is, this method selects a different character string if the character string that was previously responded to is identical to the character string about to be responded. This is a potential solution to redundant responses. However, it is not an optimal solution since the operation takes a relatively long time, depending on the string length.

Alternatively, it is possible to have an index for responses in DB, and provide a different response by making a random selection from the index when a duplicate response occurs. In this method, the operation speed is fast. However, there is a problem in that the re-selected random index is overlapped with the index of the previous response, which could increase the operation time.

In our study, we compared the response performance of the proposed method with the above-mentioned random index method. In this experiment, we measured the operation (execution) time for the duplicate avoidance operations over 1,000,000 iterations, and generated the average operation time by repeating the same experiment 100 times. For the proposed method, we applied both $N/2$ and $N/3$ as the range of random numbers. Moreover, the operation time of both Pattern 1 and 2 is measured. The case of Pattern 3 is excluded because it includes an operation to select which keywords are to be used to respond to the user's input.

Fig. 15 shows the experimental results for response performance of Chatbot. The operation time of the proposed method is converged to 105 ms, while the random index method is from 103 ms to 159 ms. It is confirmed that the random index method requires about 30% more time than the proposed method, in the case of the Pattern 1 and 2. Moreover, the random index method requires about 44% more time than

the proposed method, in the case of both Pattern 1 and 2. In the random index method, as mentioned above, this is because the index corresponding the previous response is re-selected and eventually the method generate the random index again. The experimental results show that the proposed method has better performance than the general random index method irrespective of the number of responses of chatbots.

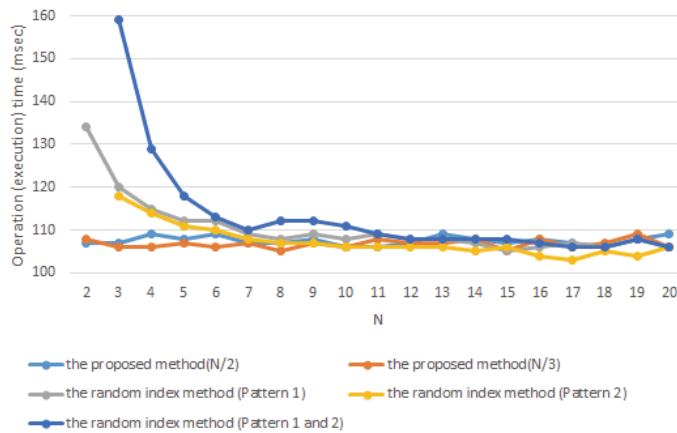


Fig. 15. Comparison of response performance.

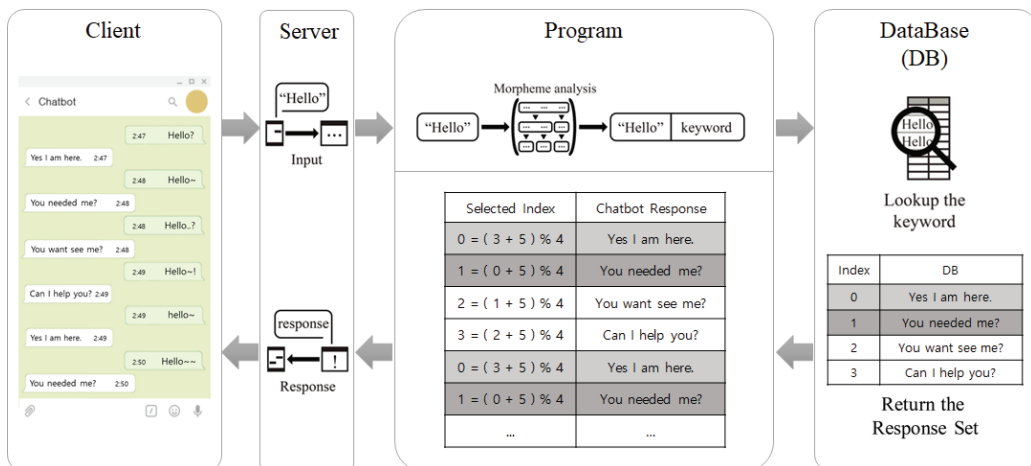


Fig. 16. Operation process of the variated equation.

5.6 Approach Variation Analysis

In addition to the method proposed in this paper, the problem of redundant response of chatbots could be solved by other equations. The following equation is one example:

$$temp_k = (temp_{k-1} + PN) \% N \tag{3}$$

where N is the number of responses retrieved from the DB corresponding to the keyword extracted from the user utterance, and PN is a prime number. In this equation, the prime number is added to the current

index ($temp$) in DB, and then it is divided by the number of responses, N . Finally, the response corresponding to $temp_k$ is provided to the user of chatbots.

Although the equation could be helpful to avoid redundant responses of Chatbots, there is an issue that a certain pattern occurs in responses of chatbot. Fig. 16 shows this issue when N is 4 and PN is 5.

It would be possible to change PN every fixed period to solve the issue so that a certain pattern does not occur. However, this could be a complicated method with high execution complexity. The proposed method covers the both the redundant pattern and the redundant response issues.

6. Conclusion

In this paper, we proposed a method to improve the response diversity of a chatbot and to avoid redundant responses when there are multiple chatbot responses for the user utterance. In the proposed method, for the user utterance, major keywords were extracted by the morpheme analysis, and the responses related to each keyword were delivered to the user through the mathematical algorithm. Furthermore, the redundancy patterns of chatbot responses, according to the utterances entered by the user, were defined and classified into three types.

In order to verify the proposed method, a comparative analysis was conducted through an experiment for each pattern. The results showed that the proposed method significantly reduced the redundancy probability, compared with the ALICE method under the same conditions.

Although there were considerable encouraging results in this study, we will conduct further studies on the following issues. We will research methods for avoiding complex redundancy patterns that occur when the user utterance is very long. Moreover, we will investigate the methods to provide additional related information with correct answers for users, through various data analysis [6,21-25].

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2017R1C1B5018295).

References

- [1] P. K. Sharma, J. H. Ryu, K. Y. Park, J. H. Park, and J. H. Park, "Li-Fi based on security cloud framework for future IT environment," *Human-centric Computing and Information Sciences*, vol. 8, article no. 23, 2018. <https://doi.org/10.1186/s13673-018-0146-5>
- [2] S. Chu and K. Zhu, "Designing a vibrotactile reading system for mobile phones," *Journal of Information Processing Systems*, vol. 14, no. 5, pp. 1102-1113, 2018.
- [3] J. Kang, "Mobile payment in Fintech environment: trends, security challenges, and services," *Human-centric Computing and Information Sciences*, vol. 8, article no. 32, 2018. <https://doi.org/10.1186/s13673-018-0155-4>
- [4] S. H. Lee, J. H. Roh, S. H. Kim, and S. H. Jin, "Feature subset for improving accuracy of keystroke dynamics on mobile environment," *Journal of Information Processing Systems*, vol. 14, no. 2, pp. 523-538, 2018.

- [5] A. Shevat, *Designing Bots: Creating Conversational Experiences*. Sebastopol, CA: O'Reilly Media Inc., 2017.
- [6] S. S. Jeong and Y. S. Seo, "Improving response capability of chatbot using twitter," *Journal of Ambient Intelligence and Humanized Computing*, 2019. <https://doi.org/10.1007/s12652-019-01347-6>
- [7] A. Zhou, M. Jia, and M. Yao, *Business of Bots: How to Grow Your Company Through Conversation*. Ithaca, NY: Topbots Inc., 2017.
- [8] M. McTear, Z. Callejas, and D. Griol, *The Conversational Interface: Talking to Smart Devices*. Cham, Switzerland: Springer, 2016.
- [9] Wikipedia, "Windows Live Messenger," 2012 [Online]. Available: https://en.wikipedia.org/wiki/Windows_Live_Messenger
- [10] J. Guynn, "Zuckerberg's Facebook messenger launches 'chat bots' platform," 2016 [Online]. Available: <https://www.usatoday.com/story/tech/news/2016/04/12/facebook-messenger-f8-chat-bots/82919056/>
- [11] S. Mirri, M. Rocchetti, and P. Salomoni, "Collaborative design of software applications: the role of users," *Human-centric Computing and Information Sciences*, vol. 8, article no. 6, 2018. <https://doi.org/10.1186/s13673-018-0129-6>
- [12] J. Weizenbaum, "ELIZA: a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, no. 1, pp. 36-45, 1966.
- [13] Eliza, "Computer therapist," [Online]. Available: <http://psych.fullerton.edu/mbirnbaum/psych101/eliza.htm>
- [14] L. S. Levin, D. A. Evans, and D. M. Gates, "The ALICE system: a workbench for learning and using language," *CALICO Journal*, vol. 9, no. 1, pp. 27-56, 1991.
- [15] R. S. Wallace, "The anatomy of ALICE," in *Parsing the Turing Test*. Dordrecht, The Netherlands: Springer, 2009, pp. 181-210.
- [16] M. Probert, "The Mechanics of Human Conversation," [Online]. Available: <https://goo.gl/EMkR3H>
- [17] M. Probert, "Conversations with My Computer," [Online]. Available: <https://goo.gl/o4LyuG>
- [18] Telegram Custom Themes [Online]. Available: <https://www.telegram.org/blog/android-themes>
- [19] Telegram [Online]. Available: <https://www.telegram.org>
- [20] Telegram Bot APIs [Online]. Available: <https://core.telegram.org/api>
- [21] Y. S. Seo and D. H. Bae, "On the value of outlier elimination on software effort estimation research," *Empirical Software Engineering*, vol. 18, no. 4, pp. 659-698, 2013.
- [22] Y. Jang, C. H. Park, and Y. S. Seo, "Fake news analysis modeling using quote retweet," *Electronics*, vol. 8, no. 12, article no. 1377, 2019. <https://doi.org/10.3390/electronics8121377>
- [23] Y. S. Seo and J. H. Huh, "GUI-based software modularization through module clustering in edge computing based IoT environments," *Journal of Ambient Intelligence and Humanized Computing*, 2019. <https://doi.org/10.1007/s12652-019-01455-3>
- [24] B. Yin, X. Wei, J. Wang, N. Xiong, and K. Gu, "An industrial dynamic skyline based similarity joins for multi-dimensional big data applications," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2520-2532, 2019.
- [25] S. Ahmed, M. Z. Asghar, F. M. Alotaibi, and I. Awan, "Detection and classification of social media-based extremist affiliations using sentiment analysis techniques," *Human-centric Computing and Information Sciences*, vol. 9, article no. 24, 2019. <https://doi.org/10.1186/s13673-019-0185-6>



Hyuck-Moo Gwon <https://orcid.org/0000-0002-8765-1863>

He received the B.S. degree in the Department of Computer Engineering from Yeungnam University in 2018. Since March 2018, he is currently with the Department of Computer Science from Hanyang University as a M.S. student. His research interests include artificial intelligence, social network service, and data mining.

**Yeong-Seok Seo** <https://orcid.org/0000-0002-5319-7674>

He received the B.S. degree in computer science from Soongsil University, Korea, in 2006, and the M.S. and Ph.D. degrees in computer science from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 2008 and 2012, respectively. From September 2012 to December 2013, he was a postdoctoral researcher in KAIST institute for Information and Electronics. From January 2014 to August 2016, he was a senior researcher in Korea Testing Laboratory (KTL), Korea. He is currently an assistant professor in the Department of Computer Engineering, Yeungnam University, Korea. His research interests include software engineering, artificial intelligence, Internet of Things (IoT), and data mining.