

# A Study on Reducing Duplication Responses of Chatbot Based on Multiple Tables

Hyuck-Moo Gwon<sup>†</sup> · Yeong-Seok Seo<sup>††</sup>

## ABSTRACT

Various applications are widely developed for smartphones to meet customer's needs. In many companies, messenger's typed interactive systems have been studied for business marketing, advertising and promotion to provide useful services for the customers. Such interactive systems are usually called as "Chatbot". In Chatbot, duplicated responses from Chatbot could occur frequently, and these make one lose interest. In this paper, we define a case that the response of Chatbot is duplicated according to the user's input, and propose a method to reduce duplicated responses of Chatbot. In the proposed method, we try to reduce duplication responses through a new duplication avoidance algorithm by building multiple tables in a database and by making combinations of user's input and its response in each table. In our experiments, the proposed method shows that duplicated responses are reduced by an average of 70%, compared with the existing method.

**Keywords :** Chatbot, Duplicated Responses, Interactive System

# 다중 테이블을 활용한 챗봇의 중복 응답 감소 연구

권혁무<sup>†</sup> · 서영석<sup>††</sup>

## 요약

현재 스마트폰에서는 사용자의 요구에 맞는 다양한 앱(App)들이 활용되고 있는데, 특히 많은 기업들에서 비즈니스 마케팅, 상업적인 홍보 등을 위해 모바일 메신저 형태의 대화형 시스템을 연구 개발하여 고객들에게 다양한 서비스를 제공하고 있다. 이러한 대화형 시스템을 우리는 '챗봇(Chatbot)'이라고 부른다. 이러한 챗봇의 경우 사용자와 대화시 중복 응답이 자주 발생할 수 있는데, 이러한 중복응답의 경우 특정 서비스에 대한 사용자의 흥미와 관심도를 떨어뜨린다. 따라서 본 논문에서는 이러한 챗봇 개발시 사용자로부터 입력된 발화(utterance)에 따른 챗봇의 중복 응답을 정의하고 이를 감소시킬 수 있는 기법을 제안한다. 이를 위해 데이터베이스 내 다중 테이블을 구축하고 테이블별로 사용자 입력에 따른 챗봇의 응답 조합을 구성하여 새로운 중복 회피 알고리즘을 통해 챗봇의 중복 응답을 감소시킬 수 있도록 한다. 이렇게 제안한 기법의 검증을 위해 자동화된 챗봇을 구현하였고, 본 연구에서 제시한 기법과 기존 응답 방식 연구를 분석해본 결과, 본 연구에서 제안한 기법을 통해 평균 70% 정도의 중복 감소 효과를 확인할 수 있었다.

**키워드 :** 챗봇, 중복 응답, 대화형 시스템

## 1. 서론

스마트폰 중심의 모바일 시대가 도래하게 됨에 따라 이제 는 언제 어디서든 인터넷에 접근하여 다양한 정보들을 교환 및 공유하고 있다. 이러한 기술 트렌트 변화에 따라, PC기반

의 시대에서 포털사이트가 플랫폼 역할을 하였듯, 모바일 시대에서는 메신저 앱(App)이 여타의 앱들을 포괄하는 새로운 플랫폼으로 도약을 시작하고 있다.

스마트폰 사용자들에게 가장 익숙한 서비스는 모바일 메신저라고해도 과언이 아니다. 같은 집에서 부모와 자녀들이 방문을 열고 대화하는 대신 메신저를 이용하여 대화를 나누는 모습도 그리 놀랍지 않은 세상이 되었다. 현재 이러한 메신저 서비스가 고객들의 편의를 도모하기 위한 상업 플랫폼들과 연동하여 "챗봇(chatbot)"이라는 형태로 급격히 진화하고 있다. 챗봇이란 우리가 일상적으로 사용하는 메신저 안에서 정해진 응답규칙(rule)에 따라 채팅 인터페이스를 통해 사용자의 질문에 응답할 수 있는 사람이 아닌 가상의 대화상대

※ 이 성과는 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2017R1C1B5018295).

† 비 회 원 : 한양대학교 컴퓨터소프트웨어학부 석사과정

†† 중 심 회 원 : 영남대학교 컴퓨터공학과 교수

Manuscript Received: February 23, 2018

First Revision: May 31, 2018

Accepted: June 29, 2018

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

또는 대화를 하는 로봇이라고 정의할 수 있다. 챗봇은 현대의 일상생활에서 흔히 접할 수 있게 된 대화형 시스템으로 초기엔 심리치료용[1] 목적으로 개발되었지만 현재 많은 기업 및 단체에서 고객에게 서비스를 제공하기 위한 수단으로서 정보 안내, 질의응답, 엔터테인먼트 등의 목적으로 활발히 발전하고 있다[2-4].

과거 PC기반의 시대에도 PC기반의 메신저들(e.g., MSN messenger)이 존재하여 챗봇의 기술을 적용할 수 있었지만, “PC 메신저”들은 로그인 기반의 서비스라 로그오프 이후에는 서로간의 대화가 단절되고 더 이상 이용할 수 없다는 한계가 있었다. 그러나 모바일 시대로 넘어오면서 스마트폰의 ”모바일 메신저“들은 사용자들이 항상 접속되어있는 상태로 유지되기 때문에 비즈니스 마케팅, 커머스 통로로 활용하기에 매우 적합한 상황이 되었다. 즉, 일상에서 가장 익숙한 커뮤니케이션 도구가 온라인 쇼핑을 즐기고 물품을 주문하고 숙박을 예약하며 은행에 대출상담에 이르기까지 자연스럽게 채팅만으로 해결할 수 있는 시대가 펼쳐지고 있는 상황이다[5]. 전세계 대표 SNS서비스를 제공하는 기업의 CEO도 SNS의 미래를 지능형 챗봇이라고 할 정도로 IT업계에서 지능적인 챗봇 기술들에 대해 많은 관심과 연구가 이루어지고 있다[6].

이러한 지능형 챗봇에 대한 기술 연구의 핵심 이슈 중 하나는, 사용자들이 챗봇서비스에 전송하는 특정 발화(utterance)들에 대해 대화상대가 실제 사람인것처럼 보다 유연하고 융통성있는 답변을 제공할 수 있는가 하는 것이다. 즉, 사용자들이 전송하는 발화가 중복적이더라도 고객들의 요구에 맞도록 보다 현명한 응답 정보를 제공할 필요가 있다. 그러나 아직까지, 사용자가 연속적으로 같은 발화를 챗봇서비스로 전송하면 챗봇서비스 개발자의 단순 의도에 따라 중복적으로 같은 응답이 제공되는 경우가 대부분이다. 이는 챗봇서비스에 대한 사용자의 신뢰성을 떨어뜨리고 흥미를 저하시키며 제한된 정보만을 제공하게 되는 문제점을 야기한다.

따라서, 본 논문에서는 사용자가 챗봇서비스 이용시 같은 발화를 중복적으로 연속 입력할 경우 보다 다양한 응답 사례를 제공할 수 있는 기법을 제안한다. 본 기법에서는 사용자가 발화를 입력 후 챗봇 프로그램에 전송하면 데이터베이스에 저장된 여러 관련 응답 중 하나를 선별하여 사용자에게 전달하게 된다. 중복 응답 문제를 해결하기 위해 여러 가지 응답 예들 중 단순히 랜덤 방식으로 선택하지 않고 제안한 방법으로 데이터베이스에서 다중 테이블로 분할된 사용자 입력에 대한 챗봇의 응답 조합을 세부 알고리즘에 의해 선별하여 응답 중복 횟수를 크게 줄일 수 있도록 하였다. 이에 따라 본 연구에서는 사용자 발화에 따른 챗봇 응답의 중복을 정의하였고, 실험을 통해 본 연구에서 제안한 기법을 적용하여 구체적으로 검증하였다.

이에 본 논문의 공헌은 다음과 같다.

- 첫째, 기존의 연구들에서 중요 이슈로 고려되지 않았던 챗봇 응답의 중복성에 대해 사용자의 관점에서 환기하고 이를 정의.
- 둘째, 챗봇 응답의 중복 감소와 랜덤성 향상을 위해

데이터베이스에서 응답 테이블 분할 및 각 테이블별 사용자 입력과 챗봇의 응답 조합 체계 구성.

- 셋째, 챗봇 응답 선택시 소수를 활용한 고유의 중복 회피 알고리즘을 제안.
- 넷째, 데이터베이스를 연계할 수 있는 언어(C#, JSON 등)로 개발된 최신 챗봇 플랫폼의 경우 제안된 기법을 쉽게 적용 가능.
- 다섯째, 제안한 방법을 기존의 챗봇과의 성능비교 실험을 통해 우수성을 입증.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 살펴보고 3장에서는 챗봇과의 대화시 중복에 대해 정의하고 중복 횟수 감소를 위한 기법을 구체적으로 제시하였다. 4장에서는 제안한 기법의 검증을 위한 실험 설계와 기존 기법과의 비교분석 결과를 제공한다. 마지막 5장에서는 결론 및 향후 연구를 기술한다.

## 2. 관련 연구

### 2.1 챗봇응답구현방식 1: Combined keyword-base case

챗봇에는 여러 가지 구현 방식이 있다. 입력된 사용자의 발화에서 키워드를 가져와 응답을 생성하는 방식이 그 중 하나이다. 또한 이를 응용한 ELIZA는 채팅 로봇의 시초라고 해도 무방한 챗봇 로봇이다. 1966년 MIT 인공지능 연구실의 Joseph Weizenbaum. 교수에 의해 Rogerian 방식의 심리치료 목적으로 개발되었다. Rogerian 심리치료방식이란 상담자의 대화내용으로부터 질문을 만들어내고 격려하는 방식으로 대화를 유도하는 심리치료 방식으로 ELIZA또한 이 방식을 사용한다. 이는 상담자의 발화에서 ‘Me’나 ‘I’를 ‘You’로 ‘My’를 ‘Your’과 같이 바꾸고 키워드를 되물어보는 트릭으로 상담자로 하여금 대화를 유도해나간다[1].

Table 1. A Conversation with ELIZA

> 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 no?
* Because I want to be happy.
> Why do you want to be happy?
* Because I'm tired of being depressed.

또한 ELIZA는 위의 트릭과 같은 방법 말고도 단순 매칭기법을 사용하여 상담자가 불쾌한 말을 입력할 경우 그것을 자제하라는 응답 하는 기능을 추가하는 등 실제로는 단순 패턴

매칭에 그친 챗봇이다. 그러나 Weizenbaum, 교수가 대학내 직원에게 배포한 결과 그들이 점차 ELIZA에 대한 의존도가 높아지고 많은 호응을 얻는 등 엄청난 파장을 몰고 온 챗봇이라고 할 수 있다. Table 1은 실제 ELIZA의 대화 내용으로 대화내용을 보면 위에서 언급한 트릭을 보다 쉽게 확인할 수 있다.

위의 대화내용을 보면 상담자(\*)가 “I’m depressed.”라고 입력할 경우 ‘I’를 ‘you’로 ‘depressed’라는 키워드를 그대로 가져와 문장을 만든 것을 확인할 수 있다. 또한 ELIZA는 단순 패턴 매칭을 사용하기 때문에 상담자가 ‘Hello’를 입력하든지 또는 ‘Hello Eliza’를 입력하든지 똑같은 응답을 제공한다는 한계점을 가지고 있다[7].

### 2.2 챗봇응답구현방식 2: AIML-based case

AIML(Artificial Intelligence Markup Language)이란 XML기반의 스크립트 언어로 챗봇의 구현 시 입력된 사용자의 발화를 정규화를 거쳐 준비된 여러 응답 중 랜덤(random)으로 하나를 응답하는 방식이며 그 시초는 ALICE( Artificial Linguistic Internet Computer Entity)라는 대화형 엔진으로 1995년 Richard Wallace 박사에 의해 개발되었으며 인공지능 학계에서 가장 주목받는 연례행사인 ‘로브너 상(Loebner Prize)’에서 두 차례 우승을 거두는 등 뛰어난 성능을 자랑하는 챗봇이다[8, 9].

이는 AIML이라는 스크립트 언어를 개발해 사용함에 따라 사용자의 확장성이 용이하여 현재에도 많이 이용되며 그 원리는 일상의 대화가 제한된 문장으로 이루어진다는 것에 착안하여 사용자 발화의 정규화를 통해 사용자의 발화가 <patten>과 일치하는 여부를 판단하고 해당 <patten>에 따른 응답으로 <template>이라는 대화 셋을 두어 응답을 랜덤으로 결정하여 사용자에게 응답하는 구조로 Table 2는 AIML의 예시이며 <patten>으로 ‘안녕’이 입력되면 <template>인 ‘안녕하세요’, ‘반가워요!’, ‘안녕~’ 중 랜덤으로 하나를 응답하라는 코드이다. 그러나 ALICE방식의 경우 해당 <patten>에 대한 <template>를 랜덤으로 응답하기 때문에 사용자의 발화의 <patten>이 동일하게 입력될 경우 사용자에게 중복된 응답을 할 수 있다는 문제점이 있다.

Table 2. An example of AIML Structure

```

<category>
  <patten>Hi</patten>
  <template>
    <random>
      <li>Hi.</li>
      <li>Nice to see you!</li>
      <li>Hello~</li>
    </random>
  </template>
</category>
    
```

### 2.3 챗봇응답구현방식 3: Database-based case

챗봇의 경우 사용자의 발화에 대해 이해하고 생각하여 응답하는 것이 아니라 개발자에 의해 정해진 패턴 혹은 키워드 같은 규칙에 따라 적절히 가공, 변형되어 데이터베이스에 저장된 알맞은 응답을 찾아내어 응답한다. 이러한 과정은 주로 패턴 매칭을 사용하여 이루어지는데 가장 쉽게 사용할 수 있는 방법이 Slot-and-Frame Parser 방식이다. Table 3에서 표현되어 있는 것처럼, Slot-and-Frame Parser 방식은 입력 받은 문장과 응답할 문장으로 이루어진 테이블로 구성된다[10, 11].

Table 3. A Table Example of Slot-and-Frame Parser

Sentence	Response
Hello.	Hi, there.
What are you doing?	I was just thinking.
It is a beautiful day?	Yes, the weather is awesome.
Do you like coffee?	I do not drink.
Let’s talk it over.	What do you have on your mind?

위의 Table 3과 같이 만일 사용자가 “Hello”이라고 입력할 경우 이에 해당하는 대답으로 챗봇은 “Hi, there”를 응답하는 방식이다. Slot-and-Frame Parser 방식의 경우 구현이 쉽고 데이터베이스 테이블에 저장된 데이터가 증가할수록 챗봇의 대화 수준도 향상된다는 점이 있다. 그러나 사용자에 의해 입력된 문장이 테이블의 문장과 일치해야만 응답을 얻을 수 있고 테이블이 커질수록 테이블 내용을 탐색하는데 시간이 점차 증가한다.

현재 사용자 발화의 유사도를 측정하여 가장 유사한 발화에 대한 응답을 사용자에게 챗봇의 응답으로서 제공하는 연구들이 진행되었다[12, 13]. 그러나 이러한 연구들 또한 데이터베이스에 저장된 응답을 사용자에게 그대로 제공하기 때문에 챗봇의 중복된 응답을 야기할 수 있다.

현재에도 많은 챗봇 관련 연구 및 플랫폼(Google Assistant, Amazon Alexa, Facebook messenger bot)의 개발이 진행되고 있다[14, 15]. 그러나, 사용자의 중복된 발화에 대한 응답이 단순한 랜덤 방식, 사용자 발화에 대한 응답을 정해진 문장 양식에 따라 직접 생성해주는 방식 또는 일정 수의 준비된 답변이 계속적으로 로테이션 되는 방식(Google Assistant), 그리고 JSON을 통해 개발자가 응답을 지정해주는 매칭 방식(Amazon Alexa, Facebook messenger bot)들은 사용자의 중복된 발화에 대한 응답 처리가 충분히 고려되지 않은 설계 방식이기 때문에 챗봇 응답의 중복이 자주 발생하게 되고 이는 챗봇의 유연한 대화에 문제점을 발생시킨다. 이러한 문제점이 개선되지 않는다면 챗봇을 통해 사용자에게 필요한 정보를 충분히 제공하기에는 한계가 존재할 수 밖에 없다.

### 3. 중복 정의와 감소 방법

챗봇 사용 시 Fig. 1과 같이 사용자의 여러 입력으로 인해 도출된 챗봇의 응답이 이전의 응답과 같은 응답이 나오는 경

우를 챗봇의 중복 응답이라고 정의한다. 이러한 경우 바로 이전의 챗봇의 응답과 현재의 챗봇의 응답이 같거나 몇 번의 다른 응답 이후 챗봇의 응답이 같은 경우를 생각할 수 있다. 이는 단순히 데이터베이스에서 사용자 발화에 대한 챗봇의 응답을 1:1 매칭 혹은 여러 응답 중 랜덤하게 사용자에게 응답한 결과이다.

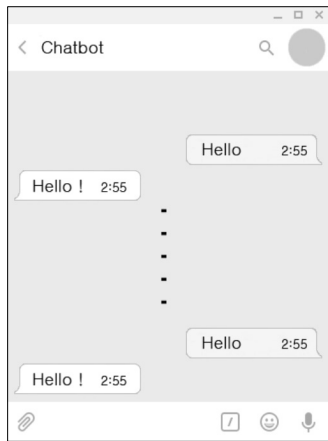


Fig. 1. Duplicated Response Messages

따라서 본 연구에서는 기존 연구들에서 챗봇의 응답을 데이터베이스에서 가져온다는 점과 사용자 발화에 대한 챗봇의 응답이 1:1 매칭이 아니라고 한다면 랜덤하게 응답한 경우 중복이 발생하는 점을 인지하여, 같은 사용자 발화에 대한 챗봇의 응답을 데이터베이스 내에 여러 테이블로 나누어 저장 후 선택된 특정 테이블에서 응답 리스트를 가져올 수 있도록 하고, 랜덤하게 응답하는 방식 대신 추가적인 알고리즘 연산을 통해 사용자에게 중복을 피해 보다 정교한 응답을 제공하는 방식을 제안한다. 다음 Fig. 2는 본 연구의 전체적인 개요도를 나타낸다.

Fig. 2에서 표현한 것처럼, 사용자가 발화를 입력하면 챗봇 프로그램에서는 사용자의 입력에 따라 데이터베이스에서 응답해야 할 메시지를 매칭하여 선택한다. 데이터베이스 내에는 응답 메시지들을 여러 테이블로 구성해두었기 때문에 어느 테이블을 선택하여 응답 메시지를 가져올지를 선택한다. 또한 특정 테이블 선택 후 해당 테이블 내의 응답 목록 중 특정 메시지를 식별하여 사용자에게 보여준다.

데이터베이스 내의 테이블은 다음 Table 4와 Table 5의 형태로서 사용자 발화에 대한 응답이 각각 다른 응답으로만 저장되어있다. 만일 Table 4 형태의 테이블이 선택된다면 “Hello”라는 사용자 입력에 대해서는 “Hello.”, “How are you, today?”, “Hi, there.” 중에서 응답하며 “Good Morning”이라는 사용자 입력을 받으면 “morning!”, “Yeah, How are you?” 중에서 하나를 응답하도록 한다. Table 5 형태의 테이블을 선택하게 된다면 사용자 입력 “Hello”에 대한 응답으로는 “How are you?”, “Hello, my name is Chatbot”, “Hi !” 중에서 선택하고 “Good morning”에 대한 응답으로는 “Yes, Good morning.”을 선정한다.

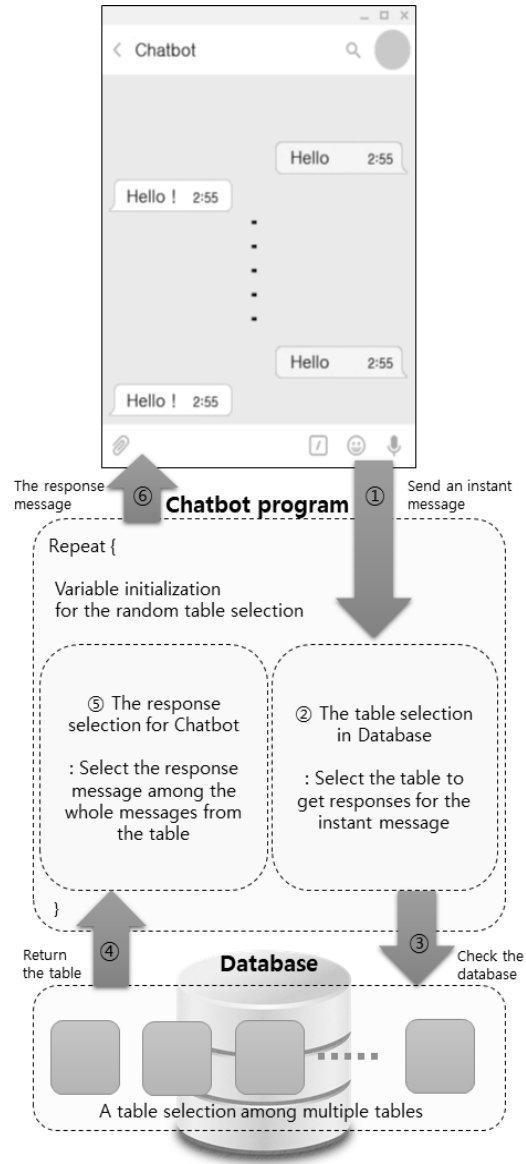


Fig. 2. Overall approach

Table 4. An Example of the Chatbot’s Response Table 1

User’s input message	A response from Chatbot
Hello	How are you, today?
Hello	Hi, there.
Good morning	morning!
Good morning	Yeah, How are you?

Table 5. An Example of the Chatbot’s Response Table 2

User’s input message	A response from Chatbot
Hello	How are you?
Hello	Hello, my name is Chatbot.
Hello	Hi !
Good morning	Yes, Good morning.

Table 6. An Algorithm to Reduce Duplicated Messages of Chatbot

**Algorithm** Chatbot\_Response\_Selection

```

rand ← Random_Value();
primeNumber ← PrimeNumber_Init(); // Initialize a prime number
currentState ← 1; // The number of input messages from an user
// (or the number of response messages from Chatbot)

Repeat
  IF currentState == DB_SIZE // If all of the items(response messages) in tables are actually used as response
  messages
    rand ← Random_Value();
    primeNumber ← PrimeNumber_Change(); // Change the existing prime number into the new one
    currentState ← Init(); // Initialize as 1

  SWITCH rand % N (N=THE_NUMBER_OF_TABLES) // N = the number of tables in the database
  case 0 : // The selection of table 0
    arrayList ← ITEMS_IN_TABLE_0
    k ← rand % THE_NUMBER_OF_ITEMS_IN_TABLE_0
    break;
  case 1 : // The selection of table 1
    arrayList ← ITEMS_IN_TABLE_1
    k ← rand % THE_NUMBER_OF_ITEMS_IN_TABLE_1
    break;
  case 2 :
    :
  case N-1 : // The selection of table N-1
    arrayList ← ITEMS_IN_TABLE_N-1
    k ← rand % THE_NUMBER_OF_ITEMS_IN_TABLE_N-1
    break;

  Chatbot's_Response ← arrayList[k]; // Provide Chatbot's response
  rand ← rand + primeNumber;
  currentState ← currentState + 1;

```

응답 메시지를 선택하고 사용자에게 보여주는 과정에서 챗봇이 하나의 메시지를 사용자에게 제공한 이후 테이블을 중복해서 선택하지 않도록 하고, 이와 더불어 특정 테이블 선택 후 응답 메시지를 중복해서 선택하지 않도록 한다면 사용자 발화에 대한 챗봇의 응답 중복을 크게 줄일 수 있게 된다. 이를 위해 랜덤 변수 및 소수(Prime number)의 조합을 통한 반복 알고리즘을 통해 사용자 발화시 특정 테이블 선택과 그에 따른 응답 메시지를 보다 다양하게 보여줄 수 있도록 하였다. 구체적인 알고리즘은 Table 6에서 상세하게 표현하였다.

Table 6은 챗봇이 응답을 결정하기 위한 전체적인 동작과정을 의사코드로 나타낸 것이다. 사용자 발화에 대한 챗봇의 응답을 제공하기 위해 우선 랜덤값이 할당되는 변수(rand)를

사용하여 어떠한 테이블에서 응답목록을 가져올지 결정한다. 이 변수는 사용자에게 이전에 응답한 테이블과 챗봇의 응답에 대한 값을 가지고 있는 변수로서 사용자가 대화를 입력할 때마다 소수를 더해 값을 변경하고 데이터베이스 내 전체 테이블의 개수(N)로 나머지 연산을 하여 항상 이전과 다른 테이블이 선택될 수 있도록 하는 역할을 수행한다.

특정 테이블이 선택되면 해당 테이블내의 응답 결과들(arrayList)을 챗봇 프로그램으로 가져오지만 가져온 응답 중 어느 결과를 사용자에게 응답해주어야 할지 선택해주어야 한다. 앞서 언급한 rand변수를 활용하여 선택된 테이블내에 저장되어 있는 전체 응답 항목의 개수(NUMBER\_OF\_ITEMS\_IN\_TABLE\_N)로 나머지 연산을 수행하여 그 결과(k)에 해당

하는 응답(선택된 테이블내의 응답 항목들 중 k번째 항목)을 사용자에게 응답 메시지로써 제공한다. 이러한 연산을 통해 사용자에게 전해지는 챗봇의 응답의 중복을 방지하고 실질적인 감소 효과를 확보할 수 있다. 추가적으로, 알고리즘 상에서 rand변수에 반복해서 더해주는 소수의 경우 일정한 값을 계속 더할 수 있기 때문에 이를 주기적으로 변경해주지 않는다면 사용자가 챗봇의 응답에 패턴이 있음을 쉽게 인지하게 될 가능성이 발생한다. 이러한 현상들까지도 미연에 방지하기 위해 데이터베이스 내에 저장되어 있는 테이블들의 응답이 최소 한 번씩 사용자에게 모두 응답되었을 경우, 나머지 연산을 통해 같은 결과가 나오는 것을 대비하여 소수를 테이블 개수와 테이블 내의 응답 항목 개수의 값과 관계없는 소수로 바꾸어 주는 것이 필요하다. 이를 위해 모든 테이블의 응답 항목 개수의 합(DB\_SIZE)마다 기존의 소수와 다른 소수로 변수를 변경(PrimeNumber\_Change) 해주었다(만일 테이블 수가 3이라면 3또한 소수인데 rand변수에 더해주는 소수는 3이 아닌 다른 소수로 더해준다는 의미). 이와 같은 알고리즘을 통해 중복 감소와 더불어 사용자가 인지할 수 있는 일정한 응답 패턴까지도 크게 줄일 수 있도록 하였다.

본 연구에서 제안된 기법에서는 기존 단일 테이블을 이용하여 중복을 회피하고자 하는 방식들에 비해 연산이 추가될 수 있다. 즉, 단일 테이블에서 랜덤하게 챗봇의 응답을 가져오는 경우(테이블 읽기 및 난수 발생 연산)와 비교해 보았을 때, 제안된 기법의 경우 다중 테이블 중 하나를 선택하는 연산 한번과 선택된 테이블 읽기 및 난수의 연산(난수 발생과 연산을 같이하므로  $O(1)$ ) 한번이 발생하므로 최종적으로 단일 테이블 방식에 비해 비교연산이 한 번 더 추가적으로 발생한다. 이는 사용자에게 다양한 정보를 제공해줄 수 있는 장점에 대한 최소의 기회비용이 될 수 있다.

#### 4. 실험 설계 및 결과

##### 4.1 실험 설계

본 논문에서 제안한 기법을 검증하기 위해 실제 챗봇을 구현하여 실험하였다. 사용자가 입력한 발화를 데이터베이스의 내용들과의 매칭을 바탕으로, 연관된 응답(response)들을 데이터베이스에서 조회하고 제안된 기법으로 연산하여 최종 응답을 선택한 후 사용자에게 제공할 수 있도록 한다.

본 연구에서 제안한 기법과의 비교를 위해 기존의 ALICE 기법을 데이터베이스로 구현하여 실험을 진행하였다. ALICE의 경우 Table 2에서 언급한 것과 같이 <patten>에 따라 <template>내의 응답들이 랜덤으로 이루어지는 응답 형태를 가진다.

실험상 객관적인 비교를 위해 본 연구에서 제안한 기법의 경우 데이터베이스 내의 테이블 개수는 3개로 설정하고 테이블 당 같은 사용자 발화의 응답을 5개씩 입력하여 활용하였다. 그리고 ALICE 기법의 경우 데이터베이스에 <사용자 발화, 챗봇의 응답>쌍을 제안한 기법과의 똑같은 개수인 15개를 한 테이블에 저장하여 사용자 발화에 따라 응답하도록 하였다.

구현된 두 기법을 비교 분석하기 위해 실험을 통해 사용자 발화 대비 평균 중복 응답 횟수를 식별하였다. 해당 실험은 C#의 Random 클래스를 사용하여 10만 번의 난수 값을 발생시켜 분석하였고 객관성 확보를 위해 같은 실험을 100번 반복수행하여 평균 중복 응답 횟수를 도출하였다.

##### 4.2 실험 결과 분석

Fig. 3과 Table 7은 본 논문에서 제안한 기법과 ALICE 기법과의 비교 분석 결과를 나타낸다. Fig. 3의 경우 사용자와

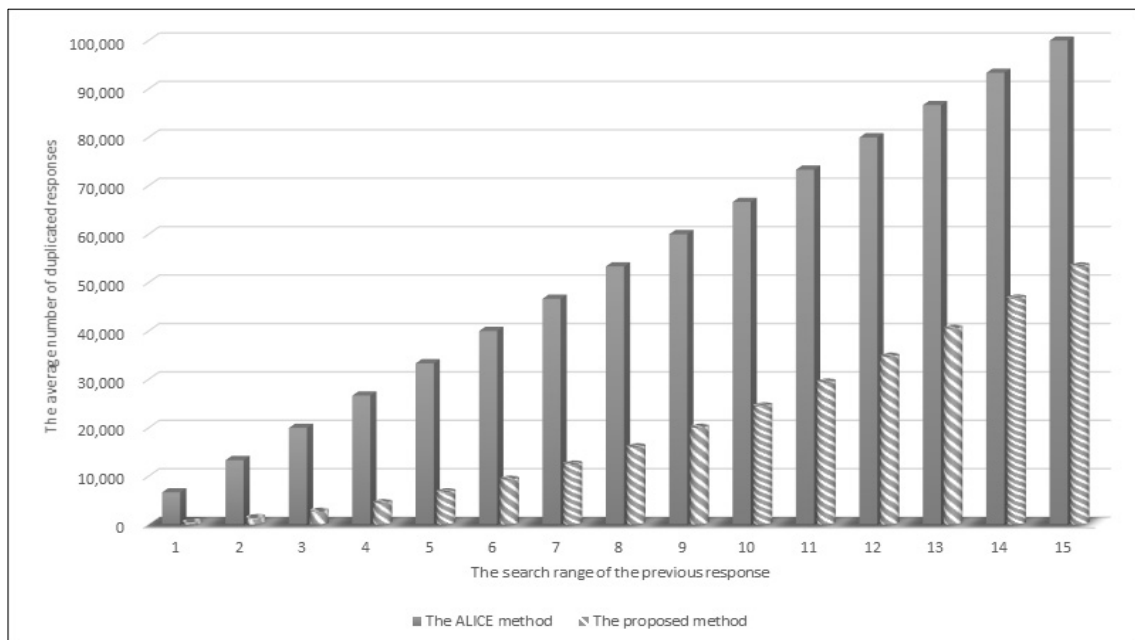


Fig. 3. A frequency of duplicated responses in each search range

챗봇의 대화시 챗봇의 이전 응답 대비 평균 중복 횟수를 보여주는 그래프이다. 해당 그래프의 Y축은 평균 중복 응답 발생 횟수이고 X축은 사용자의 입력에 따른 챗봇의 응답이 이전의 응답과 같은지 탐색하는 범위를 의미한다. X축은 사용자의 입력에 대한 챗봇의 중복 응답 판단 범위이며 챗봇이 기제공한 이전 응답 탐색 범위로서, 이전 응답 탐색 범위인 X가 1일 경우 막대 그래프 상의 수치는 챗봇이 제공한 바로 이전의 응답과의 평균 중복 횟수를 나타낸다. 또한 이전 응답 탐색 범위가 2라면 챗봇이 제공한 2번째 앞의 응답과 바로 이전 응답과의 평균 중복 횟수를 나타낸다.

비교 분석 대상인 2가지 기법들 모두 X축 값이 증가할수록 Y축의 중복 응답 횟수도 증가하는 결과를 보여주었다. ALICE 기법의 경우 난수가 일정한 확률로 발생하기 때문에 챗봇의 중복 응답 횟수가 일정하게 증가하는 것을 확인할 수 있었고, 본 연구에서 제안한 기법의 경우 챗봇의 모든 응답이 한 번씩 제공된 이후 챗봇의 응답에 추가적인 중복 패턴이 발생하는 것을 방지하기 위한 알고리즘 적용으로 인해 평균 중복 응답 횟수가 급감한 것을 확인할 수 있었다. 이 같은 결과는 Table 6에서 언급한 것처럼 소수값 변경을 통해 rand 변수값을 다시 변경해주는 원리가 가장 큰 영향을 미치는 것으로 분석되었다.

Table 7은 Fig. 3에서 소개한 챗봇의 중복 응답 탐색 범위에 따른 평균 중복 응답 발생 횟수에 대해 기존 기법 대비 감소율을 나타낸 것이다. 표의 1번째 열은 Fig. 3의 X축인 중복 응답 탐색 범위를 나타내며, 표의 2번째 열과 3번째 열은 실험에 활용한 기법들의 평균 중복 응답 발생 횟수를 의미한다. 마지막으로 표의 4번째 열은 평균 중복 응답 발생 감소율을 보여준다. 감소율의 경우 소수점 셋째자리에서 반올림하여

나타내었다. 실험결과 기존 기법 대비 최대 93.33%, 최소 46.65%의 중복 응답 발생 감소율을 확인할 수 있었으며 평균적으로 70.01% 정도의 중복 응답 감소율을 보여주었다.

특히, X가 1일 경우, 본 연구에서 제안한 기법 적용시 챗봇이 응답할 때 이전 응답에 사용했었던 테이블과 다른 테이블에서 챗봇의 응답을 가져와 사용자에게 보여주기 때문에 이론적으로 중복이 발생하지 않는다. 본 연구의 실험에서 약간의 중복이 발생하여 93% 정도의 수치를 보이는 이유는 10만번의 반복횟수(10만번의 난수값을 발생시켜 분석)보다 작은 수의 15개의 챗봇 응답들(테이블 개수는 3개로 설정하고 테이블 당 같은 사용자 발화의 응답을 5개씩 입력)로 실험을 수행하였기 때문에, 15개의 모든 응답들을 소비하고 난수를 초기화 해주는 과정의 특성상 이전과 같은 난수가 발생하여 이전 응답과의 중복이 일어나는 경우이다. X의 크기가 커질수록 자연스럽게 이러한 경향이 더욱 현저하였다.

### 5. 결론 및 향후 과제

본 연구에서는 사용자 발화에 대응하는 챗봇의 중복 응답을 감소시키고 챗봇이 보다 유연하고 융통성 있게 응답을 제공할 수 있도록 하기 위한 기법을 제안하였다. 사용자 발화에 따른 챗봇 응답의 중복을 정의하였고 이를 완화하기 위해 보다 정교한 데이터베이스 분할 기반의 수학적 연산 알고리즘을 적용하여 사용자에게 보다 현실감 있는 챗봇 응답을 제공할 수 있도록 하였다. 특히 본 연구에서 제안한 기법의 경우, 현재 많은 연구 개발이 이루어지고 있는 여러 플랫폼(Telegram, Amazon Alexa, Facebook messenger bot 등)에서 데이터베이스를 연계할 수 있는 언어(C#, JSON 등)로 개발된 챗봇 플랫폼의 경우 제안된 기법을 쉽고 빠르게 적용할 수 있다.

제안한 기법의 검증을 위해 본 연구에서 제안한 기법과 기존 연구 기법이 적용된 챗봇을 실제 구현하였고, 실험을 통해 사용자 발화에 따른 중복 응답을 서로 비교 분석한 결과 동일한 조건에서 평균 중복 응답 발생율이 평균 70.01% 감소한다는 것을 확인할 수 있었다.

본 연구를 통하여 사용자의 동일한 발화에 따른 챗봇의 중복 응답 감소에 대해 상당히 고무적인 결과를 도출할 수 있었다. 이러한 결과를 바탕으로 향후에는 데이터베이스 내에서 최적의 테이블 분할 개수 이슈와 함께 기존보다 재치있고 정확한 답변을 사용자에게 제공하기 위한 응답 루틴에 대한 추가 연구를 수행하고자 한다.

### References

[1] Joseph 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.

[2] Adelyn Zhou, Marlene Jia, and Mariya Yao, "Business of Bots: How To Grow Your Company Through Conversation," Topbots Inc, 2017.

Table 7. A Reduction Ratio for the Average Duplicated Responses

X	The Alice method	The proposed method	Reduction ratio (%)
1	6,667	445	93.33
2	13,321	1,335	89.98
3	19,988	2,663	86.68
4	26,660	4,443	83.33
5	33,324	6,666	80.00
6	39,994	9,294	76.76
7	46,658	12,406	73.41
8	53,334	15,983	70.03
9	59,993	20,012	66.64
10	66,655	24,416	63.37
11	73,315	29,336	59.99
12	79,985	34,687	56.63
13	86,646	40,461	53.30
14	93,323	46,683	49.98
15	99,979	53,336	46.65
Average reduction ratio (%)			70.01

[3] Michael McTear, Zoraida Callejas, and David Griol, "The Conversational Interface: Talking to Smart Devices," Springer International Publishing, 2016.

[4] Dong-ah Park, "A Study on Conversational Public Administration Service of the Chatbot Based on Artificial Intelligence," *Journal of Korea Multimedia Society*, Vol.20, No.8, pp.1347-1356, 2017.

[5] Sumin Choi and Yongsoon Choi, "Analysis on the Conversational Commerce Service Interface of the AI Chat-Bot Based on Mobile Messenger Apps," in *Proceedings of HCI KOREA 2017*, pp.237-240, 2017.

[6] Jessica Guynn, Zuckerberg's Facebook Messenger launches 'chat bots' platform [Internet], <https://www.usatoday.com/story/tech/news/2016/04/12/facebook-messenger-f8-chat-bots/82919056/>

[7] Eliza, computer therapist [Internet], <http://www.manifestation.com/neurotoys/eliza.php3>.

[8] Lori S. Levin, David A. Evans, and Donna M. Gates, "The Alice System: A Workbench for Learning and Using Language," *CALICO Journal*, Vol.9, No.1, pp.27-56, 1991.

[9] Alicebot, Free AIML chat bot content [Internet], <http://alicebot.wikidot.com/start>.

[10] Matthew Probert, The Mechanics of Human Conversation [Internet], [http://cnqzu.com/library/Anarchy%20Folder/Sciences%20and%20Mathematics/Psychology/Probert,\\_Matthew\\_-\\_Mechanics\\_of\\_Human\\_Conversation,\\_The.txt](http://cnqzu.com/library/Anarchy%20Folder/Sciences%20and%20Mathematics/Psychology/Probert,_Matthew_-_Mechanics_of_Human_Conversation,_The.txt).

[11] Matthew Probert, Conversations with My Computer [Internet], <https://de.scribd.com/document/174370699/Conversations-With-My-Computer-Probert>.

[12] Min-Chul Yang, Yeon-Su Lee, and Hae-Chang Rim, "A Machine Learning based Method for Measuring Inter-utterance Similarity for Example-based Chatbot," *Journal of Korea Academia-Industrial Cooperation Society*, Vol.11, No.8, pp.3021-3027, 2010.

[13] Gum-Won Hong, Yeon-Soo Lee, Min-Jeoung Kim, Seung-Wook Lee, Joo-Young Lee, and Hae-Chang Rim, "A Korean Mobile Conversational Agent System," *Journal of the Korea Society of Computer and Information*, Vol.13, No.6, pp.263-271, 2008.

[14] Amazon ALEXA [Internet], <https://developer.amazon.com/alexa>

[15] Facebook for developers [Internet], <https://developers.facebook.com/docs/messenger-platform>



### 권혁무

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

e-mail : howling6@naver.com

2018년 영남대학교 컴퓨터공학과(학사)

2018년~현 재 한양대학교

컴퓨터소프트웨어학부 석사과정

관심분야 : Big Data, Data Mining



### 서영석

<https://orcid.org/0000-0002-5319-7674>

e-mail : ysseo@yu.ac.kr

2006년 숭실대학교 컴퓨터학부(학사)

2008년 KAIST 전산학과(석사)

2012년 KAIST 전산학과(박사)

2014년~2016년 한국산업기술시험원(KTL)

선임연구원

2016년~현 재 영남대학교 컴퓨터공학과 교수

관심분야 : Data Mining, Software Modularization, Software Cost Estimation, Software Measurement and Analysis, Mining Software Repositories, and Software Process Improvement