User Adaptive Answers Generation for Conversational Agent Using Genetic Programming

Kyoung-Min Kim, Sung-Soo Lim, and Sung-Bae Cho

Dept. of Computer Science, Yonsei University
134 Shinchon-dong, Seodaemoon-ku, Seoul 120-749, Korea
{kminlim, lss}@cllab.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Abstract. Recently, it seems to be interested in the conversational agent as an effective and familiar information provider. Most of conversational agents reply to user’s queries based on static answers constructed in advance. Therefore, it cannot respond with flexible answers adjusted to the user, and the stiffness shrinks the usability of conversational agents. In this paper, we propose a method using genetic programming to generate answers adaptive to users. In order to construct answers, Korean grammar structures are defined by BNF (Backus Naur Form), and it generates various grammar structures utilizing genetic programming (GP). We have applied the proposed method to the agent introducing a fashion web site, and certified that it responds more flexibly to user’s queries.

1 Introduction

As information intensive society appears, a great deal of information is provided through diverse channels. Users also require an effective information providing service. Accordingly, we have researched the conversational agent that exchanges information between users and agents using natural language dialogue [1]. The belief that humans will be able to interact with computers in conversational speech has long been a favorite subject in science fiction. This reflects the persistent belief that spoken dialogue would be one of the most natural and powerful user interfaces to the computer [2]. Most of conversational agents lack flexibility in diverse situations because of responding repeatedly to users with the fixed answers stored in the reply database in advance. In this paper, we propose the conversational agent responding with various sentences constructed through evolutionary process to improve the user’s adaptability. By the following, it can generate various answers with matching essential keywords, which are in answer scripts, to sentence structures. Accordingly, it leads more adaptive interaction to users.

2 Related Works

2.1 Conversational Agent

A conversational agent provides users with proper information by using natural language [1]. It is a program that understands natural language and is capable of respond-
ing in an intelligent way to a user request [3,4]. The dialogue can be an effective user interface in a more complicated system, since the natural language dialogue is effective in the flexibility, succinctness, and expressive power [5].

ELIZA (http://wwwai.ijs.si.eliza), the first conversational agent, uses the simple pattern matching technique. ALICE (http://alicebot.org) utilizes the simple sequential pattern matching based on keywords. They have the shortcomings of not being able to take into accounts user’s intentions. It also takes considerably much time and effort in constructing response database based on script technique.

2.2 Natural Language Generation (NLG)

The main goal of NLG is to investigate how computer programs can produce high-quality natural language text from computer-internal representations of information [6,7]. The stages of NLG for a given application are disciplined in the order of text planning, sentence generation, and speech synthesis. Generation techniques can be classified into four main types that are canned text systems, template systems, phrase-based systems, and feature-based systems.

The grammar testing system is the first generating program, which is designed by Yngve (1962) and Friedman (1969). The applied fields of NLG range from question-answering system, expert system, database system, CAD system, and CAI system to non-conversation applications that are automatic abstracting system, report generator, and so on.

3 Generating Answers by GP

3.1 Conversational Agent System

Figure 1 shows the overall procedure of the proposed conversational agent. The proposed method uses several preprocessing processes such as morpheme analysis, spacing words, and keyword extraction. In preprocessing process, keywords are extracted from the input query, and then matching is conducted by comparing them with keywords in answer-scripts. Since the proposed method analyzes queries in a specific domain, keywords are defined as words frequently appeared on that domain, and answer-scripts are constructed with those keywords. Words, which are not related with the target domain, are ignored, and only keywords are used for the matching. When the query does not have adequate keywords, the matching is failed and the agent replies with a sentence such as "I don't understand" or "Input another query."

![Fig. 1. The system architecture.](image)
In sentence pattern categorization step, a query is classified into two general categories, question and statement, which are subcategorized into primary and secondary (Table 1). Automata that are constructed on keywords and their sequential information implement the sentence pattern classification module. An initial grammar structure is generated according to the sentence pattern. It constructs the first answer by matching the keywords extracted to the initial grammar structure generated. The following sentence structures for new answers are framed by the evolutionary process between the initial grammar tree generated and other trees. Therefore, it increases the flexibility of a dialogue because it can transfer answers of different types for the same query every time.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Keyword Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Question</td>
<td>Any, Can, Description, Fact, Location, Method, …</td>
</tr>
<tr>
<td>Secondary Question</td>
<td>Compare, Confirm, Cost, Directions, DoHave, …</td>
</tr>
<tr>
<td>Primary Statement</td>
<td>Any, Message, Act, Have, Is, Want, Fact, Other</td>
</tr>
<tr>
<td>Secondary Statement</td>
<td>Time, Conditional, Cause, Feeling</td>
</tr>
<tr>
<td>System Definition</td>
<td>RobotName, WhatUserMeant, …</td>
</tr>
</tbody>
</table>

### 3.2 Korean Grammar Rules

The early natural language generation was nothing but that prepared sentences send out changing little by little without sufficient information about grammar or lexical meaning. It might be short of the flexibility for various situations. A method of settlement is to use the grammar that can define finite rules being able to create infinite natural language [8].

In language processing fields, it generally analyzes a given sentence with a tree form. Each node of a tree represents a sentence component, which is categorized to NP, VP, and so forth according to function. The method, thus, applies grammar rules and generates various replies on a question. Grammars defined in this paper are as follows:

\[
\begin{align*}
S & \rightarrow \text{VP} + e \\
\text{VP} & \rightarrow \text{V} \mid \text{NP} + c \mid \text{Z} + \text{VP} \mid \text{NP} + j + \text{VP} \mid \text{V} + e + \text{VP} \\
\text{NP} & \rightarrow \text{N} \mid \text{N} + j + \text{NP} \mid \text{Z} + \text{NP} \mid \text{VP} + e + \text{NP} \mid \text{N} + \text{NP}
\end{align*}
\]

(S: a sentence; VP: a verb phrase; NP: a noun phrase; V: a verb; N: a noun; Z: an adverb; e: the ending of a word; c: a copula; j: an auxiliary word)

### 3.3 Answers Generation Using GP

It is difficult to generate an appropriate answer structure according to user's question pattern. Therefore, it can generate various grammar trees as evolving grammar structures according to each sentence pattern. The proposed method can create many answers by matching essential answer keywords, which are in answer scripts, to the parse tree generated. Figure 2 shows the procedure of generating answers. The proposed method is as follows:
- It analyzes a user's query pattern.
- It generates a random tree whose depth is $d$ based on defined grammar (3.2).
- It sets up the fitness of a chromosome as 0.5 (Fitness rate: 0–1).
- It selects a random chromosome tree among chromosome trees that have the fitness value more than average.
- It applies keywords extracted from answer scripts to the sentence structure of a selected tree. The script file format is same as Table 2. The <ANSWER> tag only has essential keywords of the answer for a user's query. Other constituents aptly utilize keywords or components of an input query.
- Then, the appropriate initial reply is output.
- After checking the displayed response, a user puts a score with five scales from -2 to +2. The next answer is adjusted to user's feedback.
- New sentence structures are generated continuously through evolving between the initial selected tree and random trees generated according to sentence patterns.

![Diagram](image)

**Fig. 2.** The process of generating answer sentences.

<table>
<thead>
<tr>
<th>Table 2. The example of script database.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SCRIPT</strong>&lt;br&gt;<strong>TOPIC</strong> a topic according to the subject <strong>/TOPIC</strong>&lt;br&gt;<strong>CLASS</strong> a sentence pattern <strong>/CLASS</strong>&lt;br&gt;<strong>KEYWORD</strong> a query based on keywords <strong>/KEYWORD</strong>&lt;br&gt;<strong>ANSWER</strong> a answer based on essential keywords <strong>/ANSWER</strong>&lt;br&gt;<strong>/SCRIPT</strong></td>
</tr>
<tr>
<td><strong>SCRIPT</strong>&lt;br&gt;<strong>TITLE</strong> shop_location <strong>/TITLE</strong>&lt;br&gt;<strong>CLASS</strong> LocationQuestion <strong>/CLASS</strong>&lt;br&gt;<strong>KEYWORD</strong> shop location <strong>/KEYWORD</strong>&lt;br&gt;<strong>ANSWER</strong> Shinchon Hyundai Department B1 <strong>/ANSWER</strong>&lt;br&gt;<strong>/SCRIPT</strong></td>
</tr>
</tbody>
</table>
It selects 80% of high ranks after aligning with descending order based on the fitness of a chromosome. The chromosomes removed in selection process are generated by crossover operation of the two chromosomes survived. The fitness of a chromosome generated takes the average of two chromosomes crossed. It also carries out mutation operation at a fixed rate (5%)..

Genetic algorithm that uses humans to provide fitness, rather than a programmed function to compute fitness, is called an interactive genetic algorithm [9]. In this paper, we can apply GP for generating answers by initializing the population of individuals encoded from sentence structure features, setting and evolving the fitness as 'whether the answer sentence generated is natural'. It is hard to set a clear standard because user's feedback is intervened individual subjectivity in a sizable portion. However, it is almost impossible to make suitable fitness function.

4 Experimental Results

We have conducted in two sides for verification of various replies. It sets the initial population size as 100 and tested 5 types (LocationQuestion, WhoQuestion, MethodQuestion, TimeQuestion, and DescriptionQuestion) among sentence patterns more than 30 types (Table 1) presented before.

4.1 Usability Test for Answer Sentences Generated

It inputs the questions according to sentence patterns of 5 types at random. While repeating dialogues on the same question by each 10 times, we get user's feedback scores. We confirm whether it is actually natural sentences generated or not. Then, it observes how much user's feedback affects the replies later.

![Graph](image)

**Fig. 3.** The user evaluation for reply sentences.

Figure 3 shows the variation of average scores on dialogue iteration. Seen in the result, answers are improved gradually in the process of repeating dialogue, and more adaptive answers to the user are generated with the feedback.

4.2 Diversity Evaluation for Answers Generated

We have confirmed the various replies generated with new grammar structures against the same question through the examples of actual dialogue (Table 3 and Figure 4). In conversation 1, an incomplete sentence displayed with the first reply. The
Table 3. An example of conversation.

<table>
<thead>
<tr>
<th>Conversation 1</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>YOU:</strong> Where is the location of the shop?</td>
<td><strong>BOT:</strong> Shinchon Hyundai Department B1 ($N+N+N$)</td>
<td><strong>Feedback:</strong>(-2-2): $&gt;&gt; -1$</td>
<td></td>
</tr>
<tr>
<td><strong>BOT:</strong> The shop is on the 1st floor Hyundai Department Store. ($N+N+j+j+N+j+V+e$)</td>
<td></td>
<td><strong>Feedback:</strong>(-2-2): $&gt;&gt; 1$</td>
<td></td>
</tr>
<tr>
<td><strong>BOT:</strong> It is located on the 1st floor Hyundai Department Store in Shinchon. ($N+j+N+N+j+N+V+e$)</td>
<td></td>
<td><strong>Feedback:</strong>(-2-2): $&gt;&gt; 2$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conversation 2</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>YOU:</strong> What kind of style is good?</td>
<td><strong>BOT:</strong> Casual pants and the shirt are good. ($N+N+j+j+N+j+V+e$)</td>
<td><strong>Feedback:</strong>(-2-2): $&gt;&gt; 1$</td>
<td></td>
</tr>
<tr>
<td><strong>BOT:</strong> The Striped skirt and the shirt matched. ($N+N+j+j+N+j+V+e$)</td>
<td></td>
<td><strong>Feedback:</strong>(-2-2): $&gt;&gt; 2$</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4. An example of actual conversation with the system interface.

The user gave ‘-1’ as feedback value, and then a new reply of almost perfect grammar structure was displayed with the following reply. It can be also known that grammar structures were changed every time through free structure representations in a parenthesized passage. The system saves it with optimal solution if a user gives to the ‘+2’ point that is a peak on a generation reply. Afterward, the same reply is displayed on the same question according to priority. In addition, seen in conversation 2, it has an effect on various answers generation by replies generation of new contents because answer script part deals with several lines.

5 Conclusion and Future Works

As the conversational agents respond with fixed answers, they tend to lack of the flexibility and adaptability. In this paper, we have exploited such possibility on gener-
ating answers of various types with the same stimuli, instead of just generating different answers for each query. While conventional conversational agents have static answers, the proposed agent has dynamic answers adapted to query’ sentence structures through the evolution. We have confirmed that the method increases the adaptability to users. However, it is difficult to define perfect grammars for Korean. It thus needs more studies on Korean grammar to analyze the structure of Korean sentences.

Acknowledgements

The work was supported by Korea Brain Science and Engineering Research Program sponsored by Korean Ministry of Science and Technology.

References