Learning Classifier System for Generating Various Types of Dialogues in Conversational Agent

Eun-Kyung Yun and Sung-Bae Cho

Department of Computer Science, Yonsei University,
134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749, Korea
ekfree@candy.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Abstract. Most of the conversational agents respond to the users in an unsatisfactory way because of using the simple sequential pattern matching. In this paper, we propose a conversational agent that can respond with various sentences for improving the user's familiarity. The agent identifies the user's intention using DA (Dialogue Acts) and increases the intelligence and the variety of the conversation using LCS (Learning Classifier System). We apply this agent to the introduction of a web site. The results show that the conversational agent has the ability to present more adequate and friendly response to user's query.

1 Introduction

One of the first conversational agents, ELIZA, was contrived for the research on natural language processing. This agent uses simple pattern matching technique [1]. ALICE is written in a language called AIML that is based on XML (http://alicebot.org). However, it has shortcomings of not being able to respond to users reflecting their intentions because of simple sequential pattern matching based on keywords. Tackling this problem requires much time and effort in constructing the response database. Most of the conversational agents respond with the fixed answer sentences that are stored in the response database in advance. Therefore, the user may be easily bored as time passes and feel less familiarity to the site. This paper proposes a conversational agent that is able to communicate with the user in various sentences through learning with Learning Classifier Systems (LCS) [2].

2 Learning Classifier Systems

LCS is a kind of methods of genetic-based machine learning [3, 4], and it uses genetic algorithm with a classifier system. Classifier systems, associated with rule-based systems, spend much cost of constructing rules, and it is impossible to work in a changing environment [5]. LCS is appropriate to that case. Classifier system consists of rules (classifiers): one classifier has one or more condition parts that consist of ternary elements \{0, 1, #\} and one action part that consists of \{0, 1\}. The character ‘#’ can take either ‘0’ or ‘1.’ LCS is composed of three modules as shown in Fig. 1.

- Classifier system: The system compares input messages with the condition part of all classifiers and performs matching. The matching classifiers enter
competition for posting their output messages, and only the winner of the
c ompetition actually posts messages. The measure of each classifier is the value
of bid as follows:

\[ bid = c \times \text{specificity} \times \text{strength} \]

where \( c \): constant less than 1, \text{specificity}: condition's length minus number of '#'
symbols; and \text{strength}: the measure of confidence in a classifier.

![Fig. 1. Structure of LCS](image)

Credit assignment system [6]: When the rewards from the environment transmit,
it causes the strength of the classifiers to change to reflect their relevance to the
system performance. This is to classify rules according to their usefulness. The
conventional algorithm used for this purpose is the bucket brigade algorithm.

Rule discovery system: The process of rule discovery in the classifier systems
utilizes genetic algorithm. The GA selects the classifiers with greater strength to
reproduce, generating new individuals by their recombination and mutation. The
new classifiers generated take the places of the weaker ones, modifying the
classifier set of the system.

3 A Conversational Agent Based on LCS

Fig. 2 shows the structure of conversational agent in this paper. The key techniques in
this system are as follows. First, it classifies input sentences through a dialogue act
categorization [7]. It can figure out user's intention roughly and find a response close
to user's intention by dialogue act categorization [8]. Then it matches the most
appropriate response in the script database for user to get the appropriate response to
that query. In this point, we introduce LCS for giving not static answer sentences but
dynamic ones.

In dialogue act categorization, queries are classified into two general categories,
question and statement, which are subcategorized into primary or secondary.
Automata that are constructed on keywords and their sequential information
implement the dialogue act categorization module.

In order to respond, the agent accesses the script database. The script consists of a
list of classifiers, and topic is the primary component that represents a classifier list.
For all the topics, the conditional part of a topic is compared with the query, dialogue acts, and the keywords extracted during dialogue act categorization. This returns the scores of all the topics as a result. The topic with the highest score is selected and then the agent begins to read the classifier list file of that topic.

![System overview](image)

**Fig. 2. System overview**

Fig. 3 shows the structure of one classifier. The condition part consists of the bits to specify dialogue acts and two bits to specify the structure of sentences, which can determine the structure of answer sentences. In other words, the last two bits represent a complete sentence, a sentence only with keywords, and a sentence with modifiers. The current implementation does not have the two bits that will be very useful information to improve the agent for future system based on user modeling. The action part consists of subject, keywords, and predicates that answer sentence contains.

<table>
<thead>
<tr>
<th>Condition Part (n-bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dialogue Act 1</td>
</tr>
<tr>
<td>(a) Condition part</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action Part (m-bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject Part</td>
</tr>
<tr>
<td>(b) Action part</td>
</tr>
</tbody>
</table>

**Fig. 3. Condition and action parts**

After selecting the appropriate topic and reading classifier list, the agent finds matching classifiers. If there is any matching one, corresponding answer sentence posts; otherwise new sentence is created randomly and is added to the classifier list (See Fig. 4). At the same time, new classifiers are created by GA operators, crossover and mutation, and added to the classifier list. Fitness evaluation is performed using strength value that is controlled by user's feedback (evaluation score). In other words, user scores the agent’s answer sentence from -2 to +2 and the strength is recomputed.

### 4 Experimental Results

To show the conversational capabilities of the agent, we apply it to introduction of a certain research laboratory. Each classifier list has the same number of classifiers with manual initialization.
Table 1 shows a stream of conversations about "laboratory location." In conversation 1, new answer sentence was created in spite of the same question. In conversation 2, there was no matching classifier in the beginning. In this case, the agent responded by creating classifiers randomly but there was no predicate in answer sentence. After user's feedback, it responded as a complete sentence.

**Table 1. An example of conversation**

**[Conversation 1]**
User: Where is your laboratory?
Agent: It is located at the 3rd engineering building in Yonsei university
User: Where is your laboratory?
Agent: It is located at 529, the 3rd engineering building in Yonsei university

**[Conversation 2]**
User: Tell me where your laboratory is.
Agent: 529, the 3rd engineering building, 134, Yonsei university, Shinchon-dong, Seodaemoon-gu, Seoul
User: Tell me where your laboratory is.
Agent: It is located at 529, the 3rd engineering building, 134, Yonsei university, Shinchon-dong, Seodaemoon-gu, Seoul

**Fig. 5. User evaluation score against the response sentences**
In addition, we have generated various types of sentences through several conversations. After repeated learning, the agent responded using only complete sentences that had been received higher scores. Fig. 5 shows the user evaluation scores against the response sentences about the topic “agent.” Because the agent can generate various types of sentences by itself, not only it makes the designer spend less time to write scripts but also the familiarity on the agent increases.

5 Conclusions

As the conversational agents respond with fixed sentences, they tend to lack the flexibility and variety. In this paper, we have proposed a conversational agent based on LCS that can give various types of answer sentences to user’s queries in natural language. By generating new sentences, it increases the variety and flexibility of conversations. However, this agent does not have knowledge enough to understand the structures of sentences and meanings. The reason is that the sentence is constructed simply by mixing the given phrases. Therefore, we will make the agent by itself to generate the correct sentences through the investigation of grammar.

Acknowledgements. This work was supported by Brain Science and Engineering Research Program sponsored by Korean Ministry of Science and Technology and Biometrics Engineering Research Center.

References