AN INTELLIGENT AGENT WITH STRUCTURED PATTERN MATCHING FOR A VIRTUAL REPRESENTATIVE

SEUNG-IK LEE AND SUNG-BAE CHO

Computer Science Department, Yonsei University, 134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749, Korea

E-mail: cypher@candy.yonsei.ac.kr, sbcho@csai.yonsei.ac.kr

We propose a conversational agent that can act as a virtual representative of a web site interacting with visitors using natural languages. The agent consists of three main components: dialogue act categorization, structured pattern matching, and knowledge construction and representation. *Dialogue act*s are identified by automata which accept sequences of keywords defined for each of the *dialogue act*s. We use these DAs to identify the user's intention. To make the DA analysis process more effective, subsumption architecture is used to control the interactions among the DA analysis modules. Pattern matching is used for matching the queries with responses rather than the conventional natural language processing techniques, where DA, keywords extracted from DA analysis, and the query are used. 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.

1 Introduction

Since the birth of Internet in 1969, the amount of information has been increased continuously and the same applies to each site at which search engines are used to get the desired information. Most search engines at present, e.g. YAHOO or Lycos, use keyword-based methods, which are originate from the beginning of search engine era [1]. As these methods have difficulties in representation of user's intention or content, they produce too many results for users to pinpoint the information they really want.

There is, therefore, increasing need for the conversational agents that can identify the user's intention clearly and thus provide accurate information and respond quickly with friendly interaction. This paper proposes a conversational agent that is able to have a conversation with a user using natural languages.

One of the first conversational agent, called Eliza, was born at Massachusetts Institute of Technology in 1966. Eliza was contrived for the research on natural language processing. This agent uses simple pattern matching technique [2]. ALICE (Artificial Linguistic Internet Computer Entity, http://www.alicebot.org) is written in a language called AIML (Artificial Intelligence Markup Language) that is based on XML. It enables other people besides the author to modify it for their personalities. A new idea in ALICE is to tailor the conversation for categories of individual, mainly through attempts to determine the client's age, gender, geographic location and occupation. However, it has shortcomings of not being able to respond to users reflecting their intentions because of simple sequential pattern matching that is based on keywords. Tackling this problem requires much time and effort in constructing the response database.

In addition to these systems, there are also commercial products. The Nicole of NativeMinds Corporation (http://www.nativeminds.com) acts as a virtual representative for their site. It provides information related to the site to users with natural language with expressions of his/her state of mind. In addition, there are some products like SmartBot of Artificial Life Corporation (http://www.artificial-lifee.com) and Verbot of Virtual Personalities.

The goal of this paper is as follows: Firstly, we aim to present the DA classification mechanism, which identifies the user's intention and thus facilitates the matching process. Secondly, the process of structural matching of a query with a response is presented. Here, by structural matching we mean that the matching is not just sequential comparisions of components but rather evaluation of structural expression, which has more expressive power in specifying the matching conditions than the sequential one. Another point we put stress on is that we adopt the pattern-matching rather than the language processing techniques such as parsing or language generation to overcome the limits of the latter. Lastly, the grammar for the representation of the knowledge base is described.

2 Conversational Agent

The conversational agent we propose identifies the intention of a query and responses in natural language, speaks both Korean and English, and responses differently to the same query for intelligent and diverse conversation. Fig. 1 shows the overall structure of the conversational agent. When it is preprocessed for the correction of typos and replacement of synonyms, the query is put into the DA categorization module, which classifies it into categories of *dialogue act*s (DAs) and extracts keywords for each DA. The information, i.e., DA, keywords, and preprocessed query, are used to match the most appropriate response in the script database.

To construct a system like this requires research on three essential techniques. Firstly, the system should be able to classify queries into DAs, whether it is expressed in Korean or English. This enables the pattern matching to find response close to user's intention that is roughly expressed in DAs. Secondly, pattern matching techniques should support that users can get the appropriate response to their query. Lastly, how to represent the query-response is crucial to construct the knowledge for the conversational agents.

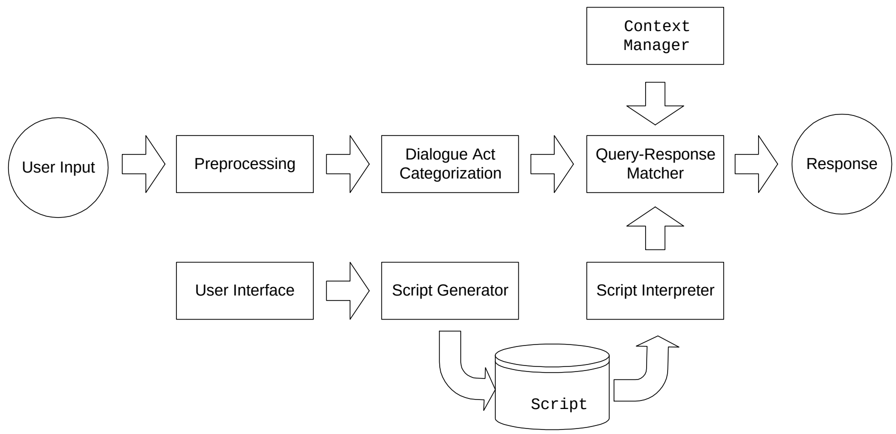

Figure 1. The structure of conversational agent

2.1 Dialogue Act Classification

If we get to know the intention of a query, it will be very useful for finding the most appropriate response. A useful approach to this is the identification of DAs. A DA represents the meaning of an utterance or a query [3]. Each query is assigned a unique DA label drawn from a well-defined set (see Table 1). Thus DAs can be thought of as a tag set that classifies queries according to a combination of pragmatic, semantic, and syntactic criteria [4]. These DA categories are usually defined so as to be relevant to a particular application, although efforts are under way to develop DA labeling systems that are domain-independent, such as the Discourse Resource Initiative's DAMSL architecture [5].

For a conversational agent, it is useful to know whether it is asked a question or ordered to do something. Queries are classified into two general categories, question and statement, which are again sub-categorized into primary or secondary, each of which consists of several DAs. As a whole, thirty domain-independent DAs are defined as in Table 1. These DAs enrich the available input for matching a query with a response. Another important role of DA information is feedback to lower-level processing such that keyword extraction is constrained by the recognized DA.

Dialogue Act Categorization module (see Fig. 1) classifies queries into DAs. Only one DA is assigned to the query in case of primary question or statement whereas several DAs are assigned to a query in case of secondary question or statement. Each question or statement has several predefined DAs as in Table 1. The Dialogue Act Categorization module is implemented by automata that are constructed

Table 1. Dialogue acts

User input	Dialogue act
Primary question	Ability, Description, Fact, Location, Method, Miscellaneous, Obiligation, Reason, Time, WhatIf, Who
Secondary question	Acquisition, Comparison, Confirmation, Cost, Direction, Example, More, Possession
Primary statement	Act, Fact, Message, Miscellaneous, Possession, Status, Want
Secondary statement	Cause, Condition, Feeling, Time

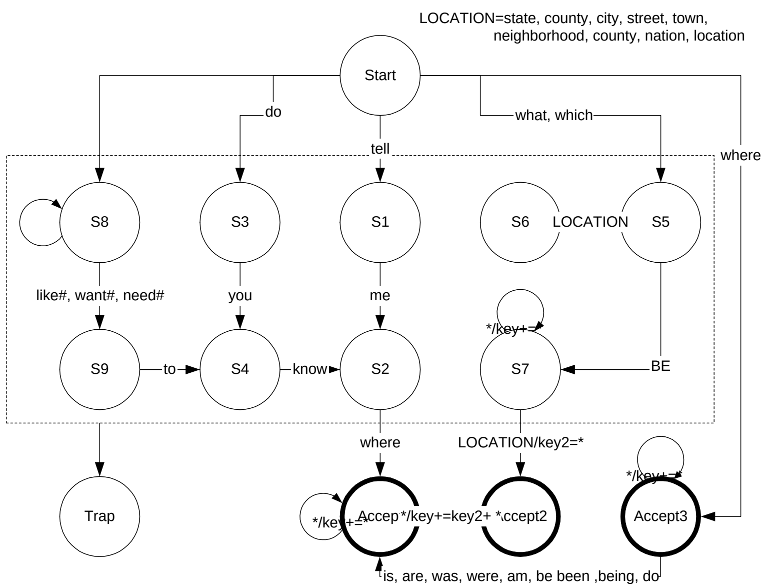

Figure 2. Automata for location question

on keywords and their sequential information.

Fig. 2 shows automata that classifies a query in English into location DA. A special meta character '#' is used to allow the ending of a word to vary according to subject, tense, etc. For example, the "want#" in the state transition from S8 to S9 can represent "want", "wants", "wanted", or the same. The classification process of the

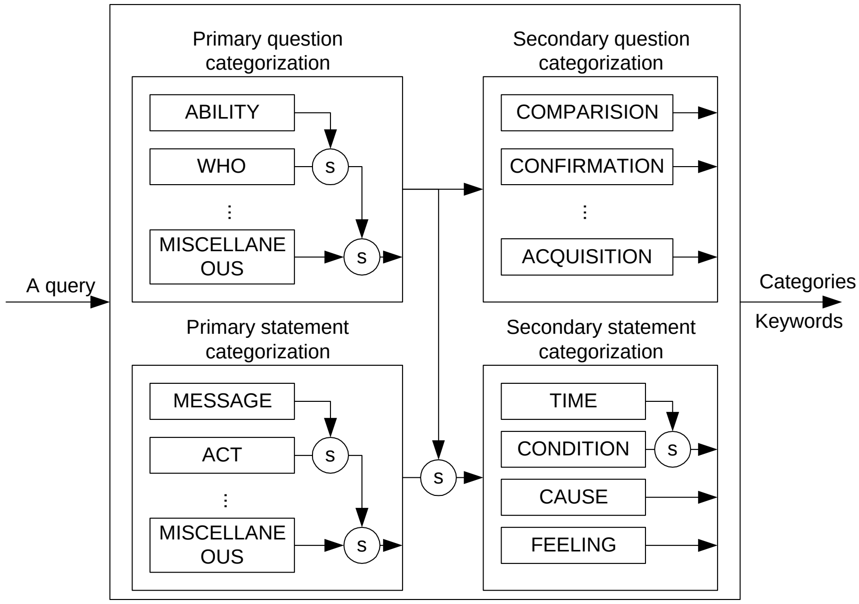

Figure 3. Dialogue acts classification using subsumption architecture

thirty DAs is as follows. At first, a query is tested whether it belongs to one of the DAs of primary question category, and if it does, once again tested for the secondary question category. A query is classified into only one of the DAs of primary category whereas several DAs of secondary category can be assigned to it.

Chances are that one query can be classified into several primary questions or statements if the automata are applied to the classification without any control or interactions between them. The reason is that there are indefinite number of queries while the number of keywords that specifies each DA is limited. This is in contrast to the formalization that only one DA of primary category should be classified to a DA of the query. To solve this problem, subsumption architecture [6] is adopted to structure and control the interactions of the automata as in Fig. 3. The subsumption architecture proposed by Brooks can represent control structures wherein each component suppresses or activates the others. The suppression is symbolized as 'S' in a circle and the activation is without it on an arrow in Fig. 3. "CAN" module, therefore, suppresses "WHO" module, which means that a query classified as "CAN" DA cannot be classified as "WHO" at the same time. Similarly, primary question category suppresses primary statement category and activates secondary question category,

List 1 A part of BNF grammar for the query-response script database

```
<topic_decl>::=TOPIC QSTRING <cond_stmt_list> ENDTOPIC
<cond_stmt_list>::=<cond_stmt>|<cond_stmt_list> <cond_stmt>
<action_list>::=<action>|<action_list> <action>
<cond_stmt>::=<if_cond> <action_list> <continuation>
<continuation>::=DONE|CONTINUE|NEXTTOPIC
<action>::=<say>|<say_one_of>
<say>::=SAY <concat_string>
<say_one_of>::=SAYONEOF <items>|SAYONEOF <concat_string>
               <items>::=<item>|<items> <item>
<item>::=ITEM <concat_string>
<if_cond>::=IF <expr> THEN
<expr>::=<expr> OR <expr>|<expr> AND <expr>|NOT <expr>|
         '(' <expr> ')'|MEMORY|MEMORY MATCH <concat_string>|
         MEMORY CONTAIN <concat_string>|HEARD <concat_string>|
         MATCH <concat_string>|CONTAIN <concat_string>|ALWAYS
```

Table 2. Comparision functions

Name	Function
CONTAIN	Checks if the specified string is substring of the target string
MATCH	Checks if the specified string is the same of the target string
HEARD	Same as CONTAIN except that the target string is a user query

and primary statement category activates secondary statement category.

2.2 Knowledge Representation

In order to respond to a query, a script database should be constructed prior to the conversation. The script consists of a list of queries (or conditions) and response pairs. A part of the script grammar is illustrated in List 1 using BNF notation.

Topic is the primary component that represents a query-response pair in the grammar. A topic begins with TOPIC keyword followed by name and one or more conditional statements followed by ENDTOPIC keyword. A conditional statement is represented as IF (condition) THEN (action). A condition is a Boolean expression composed of the operand and their operators, AND and OR. An operand is evaluated to true or not true and includes DAs and the results of comparison functions. The comparision functions are in Table 2. List 2 shows an example script. When a user asks the location or direction of something and "lab#," "softcomputing," or "soft"

List 2 A Part of Script

```
TOPIC "location of lab"
  IF ((?LOCATIONQUESTION OR ?DIRECTIONSQUESTION) AND
                              HEARD ("lab#", "softcomput-
ing", "soft"&"computing")))
  THEN
    SAYONEOF
              ITEM "It is located at the 3rd engineer-
ing building
          in yonsei university"
                 ITEM "529, the 3rd engineering build-
ing, 134, yonsei
              university, shinchon-dong, seodaemoon-
gu, seoul"
              ITEM "The 3rd engineering building in yon-
sei university"
  DONE
ENDTOPIC
```

& “computing” appears in the query, one of the items below the “SAYONEOF” is randomly selected and presented as a response to the user.

2.3 Structured Pattern Matching

Scripts according to the grammar in List 1 are interpreted and loaded into memory by script interpreter when the agent starts running. The conditional part in each topic is transformed into a Boolean expression and the keywords listed in the conditional part are transformed into a regular expression as in Fig. 4. The procedure of matching a query with a response is as follows. For all the topics, the conditional part of a topic is compared with the query, DAs, and the keyword list extracted during DA classification process. This returns scores of all the topics as a result. Different types of matching components, like DAs, keywords, or Boolean operators, are assigned different scores as in Table 3. In all the topics assigned scores, the highest topic is selected to be presented to a user as a response to his query.

3 Experimental Results

To show the conversational capability of the agent, we put it a role of introducing web site of a certain research laboratory after constructing the necessary knowledge

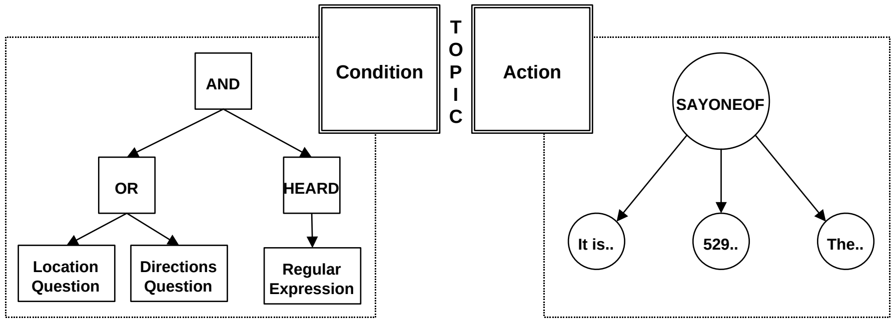

Figure 4. Internal representation of the script in List 2

Table 3. Score assignment

Target	Score
Dialogue act	2
Keyword	Number of characters
AND operator	Sum of two operands's scores
OR operator	Max of two operands's scores
NOT operator	0 : if operand is true, else the score

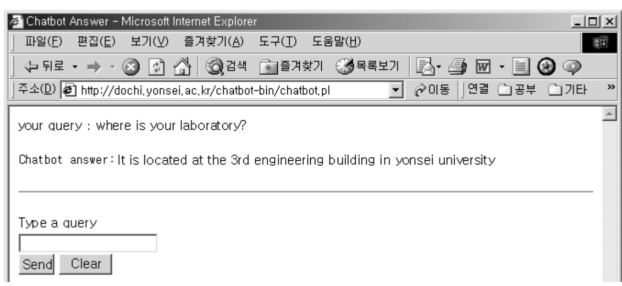

Figure 5. Location question 1

base or script. Fig. 5 shows a response of the agent to a query concerning the location of the laboratory. Fig. 6 shows a somewhat different response to a location question of the same intention. As can be seen in Fig. 5 and Fig. 6, the agent shows intelligent behaviors by responding a bit differently to the queries in the same intention. What

Figure 6. Location question 2

follows next shows a stream of conversations between a user and the agent. The agent is able to interact with users naturally presenting the requested information to them.

```
User  : who are you?
Agent : I am a virtual representative of this site to help you
        find what you want. So, what do you want to know?
User  : where is your laboratory?
Agent : It is located at the 3rd engineering building
        in yonsei university
User  : hmm, can I ask your name?
Agent : I am Walwalri.
User  : I want to know name of your professor
Agent : My professor's name is Sung-Bae Cho.
```

4 Conclusion

In this paper, we have constructed a conversational agent that can give responses to the queries of users in natural language. The agent can accept a query in both Korean and English, and give responses more consistently with the intention of users. The consistency originates from identifying the user's intention by the classification of DAs and applying them to the structured pattern matching.

Because the queries are expressed in natural language, the information and intention of users are not lost compared with the situation of expressing in several keywords so that the agent gives response that is more accurate. Furthermore, the conversational agent has an advantage of making users feel natural and friendly in finding the required information because of mutual interactions with the agent using natural language.

As further works, we plan to study on the automatic construction of the script from web pages in order to reduce the time and effort in the construction of scripts. Maintaining contextual information in a conversation is another research topic to guarantee more intelligent and consistent interactions. Finally, giving the initiative to both sides could make the conversation more natural than the current implementation of initiative given to only the user.

References

1. McBryan, O.A., GENVL and WWWW: Tools for Taming the Web. *Proceedings of the 1st International Conference on the World Wide Web*. (1994) pp. 58-67.
2. Weizenbaun, J., ELIZA - a Computer Program for the Study of Natural Language Communication between Man and Machine. *Communications of the ACM* **9(1)**. (1965) pp. 36-45.
3. Austin, J.L.: *How to do Things with Words*. (Clarendon Press, Oxford, 1962).
4. Stolcke, A. *et. al*, Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. *Computational Linguistics* **26(3)**. (2000) pp. 339-373.
5. Core, M.G. and Allen, J.F., Coding Dialogs with the DAMSL Annotation Scheme. *Working Notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines*. (1997) pp. 28-35.
6. Brooks, R.A., A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*. (1986) pp. 14-23.
7. Cho, J. *et. al*, Efficient Crawling through URL Ordering. *Proceedings of the 7th International Conference on the World Wide Web*. (1998) pp. 161-172