

유전자 프로그래밍을 이용한 대화형 에이전트의 동적 답변 생성

김경민⁰, 임성수, 조성배
⁰연세대학교 대학원 인지과학 협동과정
연세대학교 컴퓨터과학과
{kminkim⁰, lss}@sclab.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Generating Dynamic Answer Sentences for Conversational Agent Using Genetic Programming

Kyoung-Min Kim⁰ Sung-Soo Lim Sung-Bae Cho
⁰Graduate Program in Cognitive Science, Yonsei University
Dept. of Computer Science, Yonsei University

요 약

최근 정보 제공에 도움을 주는 대화형 에이전트의 연구가 활발히 진행되고 있다. 그러나 대부분의 대화형 에이전트는 사용자의 요구에 미리 준비된 정적인 답변을 제공하므로 친밀감을 주는 다양한 대화를 유지하지 못한다. 본 논문에서는 BNF(Backus Naur Form)를 이용하여 한국어 문법 구조를 정의하고, 이를 기반으로 가능한 파스트리를 하나의 염색체로 표현한 후, 유전자 프로그래밍을 적용하여 다양한 문법 구조를 생성하는 방법을 제시한다. 생성된 문법 구조에 답변 스크립트의 핵심 키워드들을 매칭시킴으로써 여러 답변 문장을 구성한다. 실제 의류 정보를 소개하는 간단한 웹 사이트에 적용하여 그 가능성을 확인할 수 있었다.

1. 서론

효과적인 정보 교환 수단인 대화를 통해 사용자와의 인터랙션을 가능하게 하는 대화형 에이전트의 연구가 활발히 진행되고 있다. 여러 연구에서 정보를 교환하고 사용자의 의도를 파악하는데 대화가 효과적인 방법임은 이미 밝혀져 있으며[1], 사람간의 대부분 정보 교환이 대화를 통해 이루어진다는 사실 또한 대화의 중요성을 부각시킨다. 그러나 기존의 대화형 에이전트는 사용자 질의에 대한 답변으로 시스템 설계자가 미리 준비한 고정된 답변 문장을 반복적으로 출력하여 친밀감이 떨어진다. 또한 다양한 상황에 대한 융통성이 적어 사용자의 요구에 적절한 답변을 제공하지 못하는 경향이 있다.

본 논문에서는 유전자 프로그래밍(Genetic Programming: GP)을 이용해 문장의 문법 구조를 진화시켜 사용자 질의에 대한 다양한 답변 문장을 생성하는 새로운 방법을 제안한다. 제안하는 방법은 한국어 문법 구조 분석을 통해 BNF로 문법 구조를 정의하고, 여기서 파생되는 파스트리에 GP를 적용시키는 방식으로 문법 구조를 진화시킨다. 생성된 문법 구조에 답변 스크립트의 핵심 키워드들을 매칭시켜 새로운 형태의 답변 문장을 출력한다. 따라서 기존의 대화형 에이전트에서 흔히 나타나는 고정된 답변 문장 출력의 한계를 극복하여 사용자와 에이전트 사이에 보다 자연스러운 대화가 이루어지도록 한다.

2. 관련 연구

2.1 대화형 에이전트

대화형 에이전트는 자연어로 해당 분야의 전문 지식을 알려주는 메신저 기반의 대화형 전문가 시스템이다[2]. 대화형 에이전트 시스템은 사용자가 원하는 정보를 단순한 키워드나 메뉴 등의 방식으로 습득하는 것이 아니라 자연어로 된 문장을 입력하게 함으로써, 사용자에게 좀더 친숙한 인터페이스를 제공한다. 최초의 대화형 에이전트라 할 수 있는 ELIZA(<http://www.ai.ijs.si/eliza>)는 단순 패턴매칭 기법을 이용하며, ALICE(<http://alicebot.org>)는 AIML을 이용한 순차 패턴매

칭 기법을 사용하여 답변을 생성한다. 기존의 대화 시스템에서 사용된 방법들은 단순 스크립트 기반으로 미리 정의된 질의와 답변 쌍을 매칭시켜 출력으로 내보내기 때문에 사용자의 의도를 반영하는 정확한 답변을 제공하지 못한다. 또한 답변 데이터베이스 구축시 많은 노력이 소모된다. 본 논문에서는 트리 구조를 진화시키는 GP[3]를 적용하여 질의 문형에 따라 생성되는 문법 구조를 진화시켜 동일 질의에 대한 새로운 답변 문장들을 생성하고자 한다.

2.2 자연언어 생성

자연어 처리에 대한 연구는 크게 자연어 이해와 자연어 생성으로 진행된다. 자연어 생성에 관한 연구는 아직 초기 단계이지만, 인간의 언어와 같이 자동적으로 문장을 생성하여 자연스러운 대화를 유도하려는 대화 시스템의 연구 분야 등에서 점차적으로 확대되고 있다[4]. 자연언어 생성은 인간이 사용하는 언어인 자연 언어를 이용하여 컴퓨터 시스템 내의 정보를 표현하여 사용자에게 전달하는 방법을 연구하는 분야로서 자연언어 이해와 함께 자연 언어 인터페이스의 실현에 매우 중요한 부분을 담당하고 있다. 자연어 생성은 주어진 어플리케이션에 대해 텍스트 계획(Text Planning), 문장 생성(Sentence Generation), 음성 합성(Speech Synthesis) 등의 단계를 거쳐 이루어진다. 생성되는 자연어는 정의된 텍스트 시스템(Canned Text Systems), 템플릿 시스템(Template Systems), 구-기반 시스템(Phrase-based Systems), 특징-기반 시스템(Feature-based System) 등의 접근법으로 구현된다. 응용 분야로는 질의-응답 시스템, 전문가 시스템, 데이터베이스 시스템, CAD 시스템, CAI 시스템 등이 있다.

3. 제안하는 방법

입력된 질의 문장의 문형을 분류한 후, 분류된 해당 문형의 문법 구조를 진화시켜 다양한 문법 규칙을 생성한다. 생성된 문법 구조에 답변 스크립트의 핵심 키워드를 매칭함으로써 동

일한 질의에 대해 유동적인 답변 문장을 출력하여 보다 유연한 대화를 가능하게 한다.

3.1 대화형 에이전트 시스템

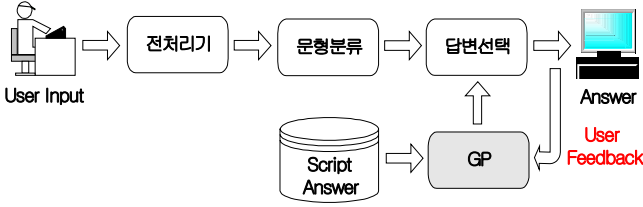


그림 1. 시스템의 구조

본 논문에서 사용되는 대화형 에이전트의 작동 과정은 그림 1과 같다. 우선 자연어 기반의 사용자 질의가 입력되면 전처리 단계에서 입력 문장에 대한 띄어쓰기 및 동의어로의 변환 과정을 거쳐 명사와 동사의 어간을 핵심 키워드로 추출한다. 동시에 문형 분석 단계에서는 입력 질의를 오토마타를 통해 설계된 30여개의 문형[표 1] 중 해당 문형으로 분류한다. 분류된 문형으로 사용자의 대략적인 질의 의도가 파악되고, 해당 문형에 따라 생성되는 초기 문법 구조에 답변 스크립트의 키워드 부분을 추출하여 적용함으로써 하나의 답변 문장을 출력한다. 이후 동일 질의에 대한 답변으로는 초기 생성된 문법 구조의 진화 과정을 통해 새로운 구조의 문장을 생성한다. 따라서 매번 유동적인 답변 문장이 생성되어 사용자의 요구에 적절히 대응한다.

사용되는 문형에 따라 일련의 키워드 리스트와 이들의 순서 정보를 이용하여 오토마타를 구성함으로써 분류를 수행한다. 문형별 오토마타를 독립적으로 적용하면 사용자 질의의 수는 많은데 각 문형에 대한 키워드 리스트는 제한되어 있기 때문에 하나의 질의가 여러 개의 문형으로 결정되어질 수 있다. 이를 해결하기 위해 포섭구조(subsumption architecture)를 이용하여 서로 활성화, 억제시킴으로써 관계를 조정한다.

표 1. 문형 분류

문형	Keyword Type
1차 질문형	Any, Can, Description, Fact, Location, Method, Other, Reason, Should, Time, Whatif, Who
2차 질문형	Compare, Confirm, Cost, Directions, DoHave, Example, More, Obtain
1차 평서문형	Any, Message, Act, Have, Is, Want, Fact, Other
2차 평서문형	Time, Conditional, Cause, Feeling
시스템 정의형	RobotName, WhatUserMeant, ...

3.2 문법 정의

언어처리 분야에서는 일반적으로 주어진 문장을 트리 형태로 분석한다. 이 트리의 각 노드는 문장 구성 성분을 나타내는데, 그 기능에 따라 명사구(NP), 동사구(VP) 등으로 구분된다. 또한 문장에 허용된 구조를 지정하기 위해서 일반적으로 문법을 사용한다. 문법은 인간의 언어 능력에 의해 해석되어 무한한 자연언어를 생성할 수 있는 유한한 규칙으로 정의될 수 있다. 따라서 문형에 따라 생성되는 여러 문법 규칙을 적용하여 하나의 입력 질의에 대해 다양한 답변 문장을 생성한다.

본 논문에서 사용하는 문장의 구조를 표현하기 위한 문법은 사용자 질의 문장의 문형을 분석하여 해당 문형에 따라 초기

문법 구조를 표현하고, 이후 해당 문형에 따라 문법 트리들을 진화시켜 다양한 문법 구조를 생성한다. 문형에 따라 생성되는 문법은 다음과 같은 정의에 따라 트리로 구성된다.

$$\begin{aligned}
 S &\rightarrow VP + e \\
 VP &\rightarrow V \mid NP + c \mid Z + VP \mid NP + j + VP \mid V + e + VP \\
 NP &\rightarrow N \mid N + j + NP \mid Z + NP \mid VP + e + NP \mid N + NP
 \end{aligned}$$

여기서 S는 문장을 뜻하며, VP는 동사구, NP는 명사구, V는 동사와 형용사의 어근, N은 명사, Z는 부사, e는 어미, c는 지정사, 그리고 j는 조사를 의미한다.

3.3 GP를 이용한 답변 생성

사용자 질의 문형에 따라 적절한 답변 형식이 다르다는 점을 착안하여 본 논문에서는 각 문형별로 가능한 문법 구조에 GP를 적용하여 진화시킴으로써 다양한 문법 트리를 생성한다. 또한 생성된 여러 문법 구조를 통해 동일 질의에 대한 다양한 답변 문장을 유도한다. 그림 2는 입력 질의에 대한 답변 문장 생성의 유도 과정을 보여준다.

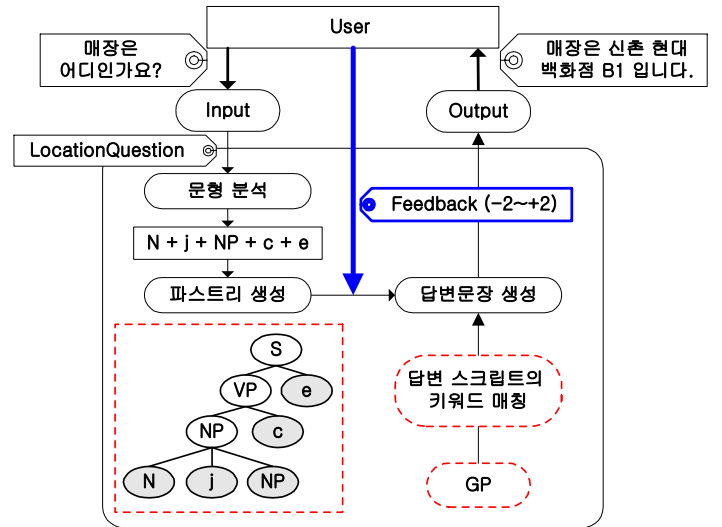


그림 2. 답변 생성의 구조

표 2. 스크립트 파일의 예

```

<SCRIPT>
<TOPIC> 주제별 세부 토픽 </TOPIC>
<CLASS> 문형 선택 </CLASS>
<KEYWORD> 질문형(키워드 단위로 입력) </KEYWORD>
<ANSWER> 답변형(핵심 키워드만 입력) </ANSWER>
</SCRIPT>

<SCRIPT>
<TITLE> shop_location </TITLE>
<CLASS> ?LocationQuestion </CLASS>
<KEYWORD> 매장 어디 </KEYWORD>
<ANSWER><C> 신촌 현대백화점 B1 </C></ANSWER>
</SCRIPT>
    
```

우선 사용자 입력 질의의 문형을 분석한 후, 3.2절에서 정의한 문법을 기초로 트리의 깊이가 d 인 n 개의 임의의 트리를 생성하고, 각 염색체의 적합도(적합도 비율은 0~1)를 0.5로 설정한다. 염색체 트리들 중에서 해당 문형의 문장 성분을 포함하며 염색체의 적합도가 평균값 이상인 것들 중 임의의 염색체 트리를 선택한다. 선택된 염색체 트리의 문법 구조에 답변 스크립트로부터 추출된 핵심 키워드를 매칭시킨다. 사용되는 스

스크립트 파일의 형식은 표 2와 같다. 스크립트의 답변 부분에는 질의에서 얻고자 하는 답변의 핵심 키워드만을 가진다. 그 외 요소들은 질의 문장내의 키워드 및 성분을 이용하여 적절한 초기 답변 문장을 생성한다. 생성된 답변 문장이 출력되면 사용자로부터 -2점부터 +2점까지 5단계로 피드백을 받는다. 초기 답변 문장의 문법 트리와 입력 질의의 문형에 따라 생성된 임의의 트리들간의 진화 과정을 통해 새로운 문법 구조의 답변 문장이 계속적으로 생성된다.

염색체의 진화 과정은 일반적인 진화연산의 작동 과정을 따른다. 염색체의 적합도를 기준으로 내림차순으로 정렬한 후, 상위 80%를 선택한다. 선택 과정에서 제거된 수만큼의 염색체는 살아남은 두 염색체의 교차 연산을 통해 생성된다. 교차는 두 염색체 트리에서 동일한 깊이의 노드를 교환하여 이루어지며, 생성된 염색체의 적합도는 교차된 두 염색체의 평균값을 취한다. 또한 일정 비율(5%)로 돌연변이 연산을 수행한다. 돌연변이 연산은 염색체 트리의 임의의 노드를 선택하여 임의적으로 변경한다.

4. 실험 결과

본 논문에서는 시스템의 성능 평가를 위해 의류 사이트 소개를 대상으로 생성 답변에 대한 사용자 피드백을 통해 간단한 테스트를 실시했다. 초기 집단의 크기를 100으로 하고, 앞서 제시한 문형 분류 단계에서 사용되는 30여개의 문형(표 1) 중 LocationQuestion, WhoQuestion, MethodQuestion, TimeQuestion, DescriptionQuestion의 5가지 문형에 대해 사용자 입력 질의에 따라 생성된 답변을 테스트했다. 우선 제시된 5가지 문형에 따라 예상되는 질의 문장들을 임의로 입력하여 동일한 입력 질의에 대해 각 10번씩 대화를 반복하면서 사용자 피드백을 받았다. 이것으로 문법 구조 진화를 통해 실제로 자연스러운 문장들이 생성되는지와 약간 어색한 문법 구조의 답변 문장이 생성되더라도 사용자 피드백을 통해 이후 생성 답변에 얼마나 반영되는지를 확인한다.

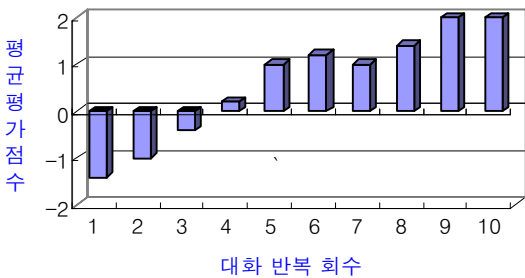


그림 3. 답변 문장에 대한 사용자 평가

그림 3은 질의 문장들의 대화 반복 회수에 따라 생성되는 답변 문장들의 점수 변화를 보여준다. 점수는 -2점에서 +2점까지 5단계로 평가한다. 그림 3의 결과에서 나타나는 것처럼 수차례의 대화를 반복함에 따라 사용자의 평가 점수 분포가 점차 개선되는 것을 확인할 수 있다. 따라서 사용자 피드백을 통해 보다 자연스러운 문장이 생성됨을 알 수 있다. 다음으로는 실제 대화의 예(표 3)를 통해 동일 질의 문장에 대한 새로운 문법 구조의 답변 문장 생성을 확인하고, 사용자 피드백에 따른 이후 생성 답변의 변화 여부를 평가한다. [대화 1]의 예를 살펴보면, 처음 질의 문장에 대한 답변으로 서술어가 생략된 문장이 출력되었다. 이에 대해 사용자의 피드백 값으로 -1을 주었더니 거의 완벽한 답변이 출력되었다. 괄호로 표시된 답변 트리의 문법 구조 표현에서 나타나는 것과 같이 처음 답변과 다른 구조의 답변 문장이 생성되었다. [대화 2]에서는 스크립트

내 답변 부분의 다중 라인 처리로 새로운 구조의 답변 생성과 더불어 답변 내용의 다양한 변화를 확인할 수 있었다.

문법 구조 진화와 사용자 피드백을 통해 동일 질의가 입력되더라도 매번 새로운 답변 문장들이 생성되어 자연스럽게 다양한 답변 생성에 효과를 보인다. 물론 핵심 키워드를 제외한 문장 성분들의 생략 등으로 간소화된 답변 문장 등이 생성되기도 하지만, 이것은 완벽한 문법 구조 정의의 미흡으로 조사 등 연결어의 부재 현상이 나타나기 때문이다. 이같은 문제점은 문법 구조 정의에 있어 한국어의 특성을 고려한 좀더 체계적인 문법 구조 정의로 해결할 수 있다.

표 3. 대화의 예

[대화 1]	
YOU: 매장은 어디인가요?	
BOT: 신촌 현대백화점 B1 (N+N+NP)	
Feedback[-2~2]: >> -1	
YOU: 매장은 어디인가요?	
BOT: 매장은 신촌 현대백화점 B1입니다. (N+j+NP+V+e)	
Feedback[-2~2]: >> 2	
[대화 2]	
YOU: 어떤 스타일이 좋을까요?	
BOT: 기본 면바지와 셔츠가 좋습니다. (N+N+j+N+j+V+e)	
Feedback[-2~2]: >> 1	
YOU: 어떤 스타일이 좋을까요?	
BOT: 앞주름 스커트와 셔츠 스타일이 좋아요. (N+N+j+N+N+j+V+e)	
Feedback[-2~2]: >> 2	

5. 결론

기존의 대화 시스템들은 미리 완벽한 문장의 답변 스크립트를 저장해 그대로 출력으로 내보내기 때문에 여러 답변 스크립트들 중 랜덤으로 출력된다 하더라도 대화의 유연성이나 다양성 측면에 있어서 다소 부족함이 있었다. 본 논문에서는 입력 질의 문장의 문형에 따른 문법 구조의 진화를 통해 보다 다양한 답변을 생성한다. 따라서 스크립트 설계시 완벽한 답변 저장 형식이 아닌 질의 문장의 답변에 해당하는 핵심 키워드들의 저장만으로도 새로운 문장을 생성할 수 있음을 알 수 있다. 그러나 여기에도 몇 가지 문제가 있다. 한국어 자체가 문장 성분 분석에 있어 너무 다양한 변수들이 존재하기 때문에 그것들을 모두 수용할 완벽한 문법을 정의하기에 어려움이 따른다. 그 결과, 문장 중 조사의 부재 등으로 인해 약간 어색한 문장이 생성될 수도 있다. 따라서 진화에 사용되는 문법 구조 정의에 대한 심도있는 연구가 필요하다.

감사의 글

이 논문은 한국학술진흥재단(2002-005-H20002)의 연구과제에 의해 지원되었음.

참고문헌

- [1] J. Allen, D. Byron, M. Dzikovska, G. Ferguson, L. Galescu and A. Stent, "Towards conversational human-computer interaction," *AI Magazine*, vol. 22, no. 4, pp. 27-38, Winter 2001.
- [2] S. Macskassy and S. Stevenson, "A conversational agent," *Master Essay, Rutgers University*, 1996.
- [3] J. R. Koza, *Genetic Programming: Automatic Discovery of Reusable Programs*, The MIT Press, 1994.
- [4] W. Weiwei, L. Biqi, C. Fang and Y. Baozong, "A natural language generation system based on dynamic knowledge base," *Proc. of the Third Int. Conf. on ICSP*, pp. 765-768, 1996