

결합범주문법을 이용한 실시간 한국어 멀티동화 제작

김현숙[○] 박종철

한국과학기술원 전산학전공 및 첨단정보기술연구센터
(hskim, park)@nlp.kaist.ac.kr

Real Time Synthesis of Multimedia Tales in Korean
with Combinatory Categorical Grammar

Hyun Sook Kim Jong C. Park

Computer Science Division & AITrc¹, KAIST

요 약

현재 인터넷을 통하여 제공되고 있는 멀티동화는 텍스트 형태의 동화와 함께 이와 관련된 애니메이션과 음성 정보를 사전에 제작한 뒤에 사용자에게 제시하는 방법을 택하고 있다. 본 논문에서는 자연언어처리 기법을 이용하여 텍스트 형태의 한국어 동화를 실시간으로 분석한 뒤 그 내용을 삼차원 애니메이션으로 제시하는 멀티 동화 제작 시스템을 소개한다. 이 시스템은 줄거리가 완성된 동화뿐만 아니라 사용자가 변경한 줄거리를 가지는 동화에 대해서도 이에 따른 삼차원 애니메이션을 제시하여 보다 사용자 친화적인 인터페이스를 제공할 뿐만 아니라 사용자가 묘사하는 텍스트 형태의 내용을 입력 받아 이에 맞는 애니메이션을 제시하여야 하는 많은 분야에서 응용될 수 있다. 본 논문에서는 이와 같은 실시간 멀티 동화 제작 시스템을 구현할 때 해결하여야 하는 문제점들을 알아보고 이의 해결 방안과 구현에 대하여 논의한다.

1. 서론

초창기 애니메이션은 사용자와의 상호작용에 대한 고려가 없는 상태에서 단방향적인 정보를 제공하였다. 그러나 최근에는 대화형 영화나 가상 현실 시스템과 같이 사용자와의 대화가 많은 부분을 차지하는 대화형 시스템이 점차 증가하고 있는 추세이다. 이러한 대화형 시스템에서 사용자와 대화를 나누는 방법은 GUI(Graphic User Interface)나 모션 캡처, 음성 등으로 다양하게 제공되고 있다. 이 중에서 음성이나 텍스트를 통한 대화 방식에서는 원활한 의사 소통을 위하여 자연언어를 사용할 수 있도록 하는데, 이에 따라 자연언어 인터페이스의 개발이 활발히 이루어지고 있다. 자연언어를 통한 대화는 사용자가 시스템에 대한 별도의 지식 없이 대화를 가능하게 해주므로 보다 사용자 친화적인 인터페이스를 제공할 수 있다는 장점이 있다. 구체적으로는 다양한 통사 구조의 처리와 지시적인 참조, 그리고 생략이나 대응 표현이 가능하다는 점 등을 자연언어 인터페이스의 장점으로 본다[5].

본 논문에서는 텍스트 형태의 한국어 동화를 분석하여 그 내용을 삼차원 애니메이션으로 보여주는 멀티 동화 자동 제작에 관해 다룬다. 여기서 멀티 동화란 기존의 텍스트로 구성된 동화와는 달리 시각적 효과와 청

각적 효과를 비롯한 다양한 요소를 결합한 애니메이션을 의미한다. 기존의 멀티 동화는 사용자와의 대화가 없는 단방향적인 애니메이션이기 때문에 주로 동화마다 별도의 동영상 제작하는 방식으로 제작되었다. 그러나 이런 제작 방식은 동화마다 별도의 애니메이션을 제작해야 하는 번거로움이 있으며 사용자와 상호작용이 없기 때문에 현실감이나 교육 효과가 감소된다는 단점이 있다. 본 연구에서 다루는 멀티 동화 자동 제작 시스템은 실시간으로 자연언어 문장을 받아들여 분석한 후 이를 애니메이션으로 보여주므로 임의의 동화에 대한 멀티 동화의 제작 가능성을 보여주며 더 나아가 동화의 내용을 사용자의 요구대로 재구성하는 데에 쉽게 응용될 수도 있다.

본 연구에서는 동화를 분석하기 위해 결합범주문법(Combinatory Categorical Grammar)을 이용한다. 결합범주문법은 구문 분석, 의미 분석, 담화 분석 등 여러 각도의 문장 분석을 한 단계의 유도 과정을 통하여 수행할 수 있고, 많은 자연언어 문법 이론에서 문제로 삼는 병렬이나 추출 등의 현상을 간결하고 일관되게 설명할 수 있다는 장점이 있다. 본 논문에서는 결합범주문법을 사용하여 동화에 나타나는 다양한 형태의 한국어 문장을 처리한다.

2장에서는 자연언어를 통한 가상 캐릭터 제어와 관

¹ 본 연구는 첨단정보기술 연구센터를 통하여 한국과학기술원의 지원을 받았다

련된 기존 연구들을 살펴보고, 3장에서 동화를 분석하여 4장에서 결합법주문법을 이용한 처리 방법을 살펴본다. 5장에서는 현재 구현된 시스템을 소개하고 6장에서 결론과 함께 향후계획을 제시한다.

2. 관련연구

게임, 교육용 시뮬레이션 등을 포함한 여러 분야에서 사용자와 자연언어를 통한 대화가 가능하게 하려는 연구들이 있어왔다. 이런 연구들 중 특히 삼차원 가상 세계안의 사물들을 자연언어로 제어하는 연구들을 살펴본다.

WordsEye라는 시스템은 장면을 묘사하는 자연언어 텍스트로부터 정지된 장면을 생성한다[1]. 자연언어에서 물체들 간의 위치 관계는 대부분의 경우 추상적으로 기술되어 있기 때문에 실제 장면을 생성하기 위해서는 각 사물들의 특성을 참조하여 구체적인 위치 관계를 알아내는 작업이 필요하다. 이 시스템은 각 사물들의 특성과 사물들 간의 위치 관계에 관해 축적된 지식을 바탕으로 가장 자연스러운 장면들을 연출하려 하였다. 그림 1은 WordsEye에서 생성된 한 장면이다.



그림 1. WordsEye에서 생성된 한 장면²

본 연구는 캐릭터들을 제어하여 동적인 장면을 생성해내는 것이 목표라는 점에서 WordsEye와 구별된다. 따라서 본 연구에서는 캐릭터들의 동작이나 상태를 동적으로 제어하기 위해서 추상적인 표현을 구체적인 수준의 행동들로 세분화하는 작업과 주변 상황을 고려하여 목적 달성에 필요한 세부 명령을 계획하는 작업 등이 별도로 요구된다.

[2]는 군사훈련 시뮬레이션에서 자연언어를 통해

가상 인간들을 제어하는 시스템에 관한 연구이다. 이 시스템은 사람이 내리는 명령에 대해 가상 인간들이 자율적으로 대응하고 사람의 실제 움직임에 가까운 사실적인 동작을 보여주고 있다. 자연언어 분석에는 TAG(Tree Adjoining Grammar)를 사용하였다. [3]은 일인칭 시점 삼차원 게임에서 사용자가 자연언어 음성 명령으로 주인공 캐릭터의 움직임을 제어하는 시스템에 관한 연구이다. 이 시스템은 자연언어 분석에 TAG의 변형인 TFG(Tree Furcating Grammar)를 사용하였다. [2]와 [3] 모두 다수의 가상 캐릭터들의 행동을 한꺼번에 지시하는 명령문은 다루지 않는 것으로 보인다. 그리고 가상 캐릭터의 동작을 제어하는 명령들을 주로 다루었으므로 상태의 변화도 다루는 동화와는 구별이 된다. [4]의 시스템에서는 일본어로 된 자연언어 음성 명령을 통하여 삼차원 환경에서 가상 캐릭터들을 제어한다. 이 시스템은 극히 단순화된 가상 환경에서 매우 간단하고 지시적인 명령들만을 대상으로 하고 있으나, 한국어나 일본어 등에 자주 나타나는 조용사나 생략 현상 등의 언어 현상들에 대한 처리를 시도하였다.

3. 데이터 분석

자연언어 명령은 보통 추상적인 표현으로 기술되기 때문에 이에 대응되는 세부 동작들을 결정하고 지시 대상이 되는 객체를 선택하는 등의 구체적인 작업이 필요하게 된다. 3.1절에서는 동화로부터 캐릭터를 제어하는 명령들을 추출할 때 해결해야 하는 요소들을 살펴보고, 3.2절에서는 문장의 의미 해석을 어렵게 만드는 언어 현상들을 살펴보기로 한다.

3.1 캐릭터 제어 명령 추출을 위해 해결해야 할 요소들

한국어 동화로부터 멀티 동화를 자동으로 생성해내기 위해 해결해야 하는 요소들 중에서 특히 캐릭터의 제어에 관련된 요소들을 중심으로 살펴본다. 다수 캐릭터 제어 명령을 처리할 때, 단일 캐릭터 제어 명령을 처리할 때에는 나타나지 않았던 새로이 등장하는 문제점들도 또한 살펴본다.

3.1.1 명령들 간 시간관계 결정

동화는 시간의 흐름에 따라 이야기가 전개되기 때문에 이를 애니메이션으로 옮길 때 시간의 흐름에 맞추어 명령들 간의 실행 순서를 결정하고 실행 시간을 적절히 배분해야 화면이 자연스럽게 이어지도록 할 수 있다.

문장들 간의 시간관계를 엄밀하게 따져서 실행 순서를 결정해 주려면 여러 가지 요소들을 고려해야 한다. [6]에서는 사건 문장 간의 시간 관계 해석에 영향을 미치는 요인을 명시적 시간부사, 관점상, 순서성 수칙, 상황상, 인과규칙, 문맥정보 등으로 보고 이들을 이용하여 사건의 유형을 네 가지 층위로 구분하여 시간 순서를

² "John uses the crossbow. He rides the horse by the store. The store is under the large willow. The small allosaurus is in front of the horse. The dinosaur faces John. A gigantic teacup is in front of the store. The dinosaur is in front of the horse. The gigantic mushroom is in the teacup. The castle is to the right of the store." 라는 텍스트에 대해 생성된 장면임.

결정하였다. 실제로 여러 문장 간의 시간 순서를 결정하는 작업은 이와 같이 많은 요인들을 고려해야 하지만, 멀티 동화는 동화를 앞에서부터 순서대로 한 문장씩 이에 대응되는 애니메이션을 제시하는 것이기 때문에 문장 간의 사전 순서 결정은 다루지 않아도 된다. 여러 동작들이 포함된 문장에서 동작들 간의 시간 관계는 대부분의 경우 시간 부사구와 연결 어미를 통하여 알아낼 수 있다. 연결 어미로 연결된 둘 이상의 동작 간 시간 관계는 크게 동시 발생과 순차 발생으로 나뉘어진다.

3.1.2 참조 해결

명령의 주체나 객체 등 가상 세계의 실제 참조들은 자연언어 문장에서 대개 명사나 대명사에 대응된다. 자연언어를 애니메이션 명령으로 변환하려면 이들 명사에 대응되는 대상을 찾아내야 한다. 이들 명사는 때로는 (1)에서와 같이 관형절이나 관형사의 수식을 받아 대상의 범위를 제한 받기도 하고, (2)에서와 같이 대상의 범위를 제한하는 수식어가 필요함에도 불구하고 생략되는 경우도 있다.

- (1) 꿈이 멀리 사라지자 나무 위에 있던 친구가 내려와 엎드려 있던 친구에게 물었어.
- (2) 나무꾼은 이렇게 대답하면서 손가락으로 집 안을 가리켰습니다. 사냥꾼이 나무꾼의 손가락을 못 보고 그냥 떠나버려자 ...

(1)에서 첫번째 '친구'는 '친구'에 대응되는 참조들 중 현재의 상태가 "나무 위에 있는" 참조이고, 두 번째 '친구'는 현재의 상태가 "엎드려 있는" 참조를 지칭한다. (2)에서 '손가락'은 "나무꾼의 손가락"을 지칭하지만 수식어가 생략되어있기 때문에 이 참조를 해결하려면 문맥을 이용하여야 한다. 참조를 해결하려면 물체들의 속성 및 대상의 현재와 과거의 상태와 동작에 대한 지식 획득을 위한 문맥 처리가 필요하다.

참조를 지칭하는 이러한 표현들은 문맥으로부터 파악해 낼 수 있는 경우에 (3)와 같이 조음사로 대체되거나 (4)와 같이 아예 생략되기도 한다.

- (3) 꿈이 쿵쿵거리며 그 사람의 머리맡에서 냄새를 맡는 사이에 그는 두 눈을 꼭 감고 숨을 죽이고 있었지.
- (4) 겁이 난 오빠와 여동생은 뒷문으로 나가 버드나무 위로 올라갔어요. 호랑이가 험레벌떡 쫓아왔어요.

(3)에서 "그 사람"과 '그'가 조음사에 해당하며, (4)는 두 번째 문장의 목적어가 생략된 경우이다.

3.1.3 사물의 위치 결정

사물의 위치 결정에 관련된 문제들로는 사물 간의 위치 관계를 묘사한 표현의 처리나 사물의 위치가 문장에서 생략되어 있는 경우에 위치의 결정, 그리고 다른 사물들과 겹쳐질 경우에 재배치하는 문제 등이 있다.

자연언어 문장 내에서 사물들 간 위치 관계는 추상적으로 제시되므로 애니메이션을 위해서는 정량적 수치로 표현되어야 한다. [1]에서는 사물 간의 위치 관계를 처리하기 위해 기본 설정값과 함께 사물의 속성 중 다

른 사물과의 위치 관계에 관해 미리 축적된 지식을 이용하였다. 예를 들어 "상자는 의자 밑에 있다"라는 문장에서 상자와 의자와의 위치 관계는 축적된 지식에 따라 상자가 의자의 앉은 부분 밑의 의자 다리 사이 공간에 위치하는 것으로 그려진다.

사물의 위치가 문장에서 생략된 경우는 이를 문맥으로부터 또는 상식으로부터 추론해 낼 수 있는 경우이다. 각각의 경우에 대한 예문은 다음과 같다.

- (5) 두 친구가 함께 산길을 가고 있었어. 그런데 갑자기 큰 곰 한 마리가 나타난 거야.
- (6) 가을이 되자 부지런한 농부 덕분에 많은 과일들이 주렁 주렁 열렸어요.

(5)에서 곰의 위치가 생략되어 있는데 문맥적으로 현재 초점이 되는 장소인 "두 친구들이 있는 곳"의 앞이 된다는 것을 알 수 있으며, (6)에서 과일이 생성되는 장소는 문장 내에 직접적으로 나타나 있지는 않지만 상식적으로 '나무'라는 것을 알 수 있다.

다른 사물들과 겹쳐질 경우에 재배치하는 문제는 충돌 탐지와 재배치될 위치를 결정하는 세부문제로 나뉠 수 있는데, 충돌 탐지는 본 연구와 거리가 멀다고 생각되어 자세히 다루지 않는다. 재배치될 위치를 결정하려면 그 사물이 놓여질 수 있는 공간의 범위를 찾아야 하는데, 이 때 사물의 속성도 고려해야 한다.

- (7) 파수원을 하는 농부가 있었어요.

(7)에서 농부의 위치가 구체적으로 제시되어 있지는 않지만, 농부의 속성에 따르면 농부를 재배치할 위치의 범위는 공중은 될 수 없으므로 화면의 하부, 즉 지표면 위로 한정되어야 한다.

3.1.4 캐릭터가 실행할 세부 명령의 결정

캐릭터의 움직임 또는 상태 변화를 지시하는 명령은 동작이나 상태를 묘사하는 표현으로부터 추출할 수 있는데, 감정을 묘사하는 표현으로부터도 얻어낼 수 있다. 감정을 묘사하는 표현의 경우는 얼굴 표정 뿐만 아니라 부수적으로 생기는 동작들까지 애니메이션에 포함시켜야 내용이 효과적으로 전달될 수 있다. 예를 들어, '걱정되다'와 같이 감정을 묘사하는 표현은 얼굴 표정과 함께 손을 턱에 괴는 등의 부수적인 동작을 고려할 수 있다. 동작, 상태, 감정의 표현으로부터 추출해낸 명령을 하위 레벨의 세부 동작들로 나누어서 지시하고, 동작이나 상태의 형태와 캐릭터가 이동할 행로 등의 지정이 필요하다. 그러나 문장에 언급된 두 동작 사이에 실행되어야 하는 중간 동작이 생략되어 있는 경우들도 많아서 이를 유추해서 동작들을 자연스럽게 이어지도록 만드는 처리도 필요하다.

동화에서 묘사되는 동작은 추상적이기 때문에 실제로 캐릭터들에게 지시하는 명령은 하위 레벨의 구체적인 동작들로 구성되어야 한다. 예를 들어 '걷다'라는 추상적인 동작이 실제로 화면에 그려지기 위해서는 사람의 경우 두 다리를 앞 뒤로 움직이는 하위 레벨 명령

으로 나뉘어서 지시되어야 한다. 감정을 나타내는 얼굴 표정을 지시하는 명령 역시 얼굴의 각 근육을 조절하는 하위 레벨 명령들로 구성되어야 한다. 캐릭터의 동작과 얼굴 표정을 애니메이션에서 정교하게 표현하는 것은 대량의 그래픽 작업이 필요한 것으로 본 연구와 직접적인 관련이 없기 때문에 고려하지 않기로 한다.

움직임의 수단은 자연언어 동화에 직접적으로 드러나있지 않는 경우가 많다.

- (8) 갈증이 나는 까마귀는 물이 있는 곳을 찾아 헤맸어요.
- (9) 물을 마시러 황소 한 마리가 물가로 왔어요.

(8)에서 “헤매는 동작”의 수단은 까마귀의 기본 속성을 따라 나는 것이 될 수도 있지만 상황에 따라 걷는 것이 될 수도 있다. 마찬가지로 (9)에서 “오는 동작”의 수단도 황소의 기본 속성을 따르면 걷는 것이지만 상황에 따라 기거나 뛰는 것일 수도 있다. 이렇게 움직임의 수단은 움직임의 목적이나 캐릭터의 속성, 그리고 상황에 따라서 달라지게 된다.

이동 속도, 동작의 정도, 상태의 변화 정도 등을 나타내는 양식은 문장에서 보통 부사로 표현된다.

- (10) 한참을 그러고 있다가 꿈이 드디어 느릿느릿 그 자리를 떠났다.
- (11) 그는 두 눈을 꼭 감고 숨을 죽이고 있었다.

(10)의 ‘느릿느릿’은 이동 속도를, (11)의 ‘꼭’은 눈을 감는 동작의 정도를 나타낸다. 각 부사에 적절한 의미를 부여하여 이를 애니메이션에 반영하는 것이 필요하다.

가상 세계의 상황을 분석하여 장애물을 피해 목표 지점으로 진행하는 이동 행로를 찾고자 하는 것은 여러 분야에서 많이 연구되어왔기 때문에 본 연구에서는 자세히 다루지 않는다.

한 동작이 실행되기 이전에 먼저 수행되어야 할 동작이 문장에서 자주 생략된다. 예를 들어 ‘걷다’라는 동작의 실행은 캐릭터가 서 있는 상태에서만 가능하므로 이를 만족시키지 않는 상황에서는 캐릭터가 일어나는 동작을 먼저 실행하도록 명령해야 한다[2]. 상태의 종료 명령이 문장에 직접 제시되지 않은 경우에 문맥을 통해 상태를 종료시키는 명령을 유추해내야 하는 경우도 있다.

- (12) 까마귀는 물병 속에 부리를 넣었어요. 그렇지만 물병 주둥이가 너무 길어서 물이 닿지 않았어요. 까마귀는 너무 지쳤어요.

(12)에서는 부리를 물병 속에 넣고 있는 상태의 종료 시점이 생략되어 있는데, 이 경우 까마귀의 지친 모습을 묘사하기 전에 먼저 이 상태를 종료시키는 동작이 실행되는 것이 자연스러워 보인다. (2)에서도 손가락을 가리키는 상태의 종료 시점이 생략되어 있는데, 이 경우에는 사냥꾼이 그 자리에서 벗어났을 때 이 상태를 종료하는 것이 자연스럽다. 생략되어 있는 상태 종료 명령을 유추해내기 위해서는 동작의 의도나 문맥 등을 이용한 이와 같은 추론 과정이 필요하다.

3.1.5 다수 캐릭터 제어 명령 처리

하나의 명령으로 둘 이상의 다수의 캐릭터들을 제어할 수 있다는 점은 자연언어 사용의 장점 중 하나로 꼽을 수 있다. 다수 캐릭터 제어 명령은 보통 복수 명사를 사용하거나 병렬 구문을 사용하여 나타난다. 다수 캐릭터 제어 명령은 해당 캐릭터들 각자에게 적절한 명령을 지시하는 것이므로 단일 캐릭터 제어 명령문들로 구성된다. 이 때 단일 캐릭터 제어 명령과는 별도로 참조 해결 문제와 타 캐릭터들과의 상호작용 문제 등이 생겨난다.

- (13) 하지만 까마귀들은 이 나무 저 나무로 날아다니면서
- (14) 사람들은 다시 몽둥이를 들고 산으로 올라갔어요.

(13)에서 각 까마귀들에게 나무의 참조를 적절히 배분해야 하는 참조 해결 문제가 발생한다. 상황에 따라 공통된 참조일 수도 있고 캐릭터들마다의 별도의 참조일 수도 있다. 타 캐릭터들과의 상호관계 문제라는 것은 주로 (14)과 같이 캐릭터들에게 무리 행동을 하도록 지시하는 경우에 해당되는데, 무리 내의 캐릭터들끼리 상호작용을 하지 않으면 자칫 군대 행진하는 듯한 딱딱한 모습으로 묘사될 위험이 있다. 이를 위해서는 상황을 분석하여 무리들끼리의 관계를 알아낼 필요가 있다.

3.2 의미 해석을 어렵게 만드는 언어 현상들

자연언어 문장에 나타나는 구조적 중의성과 생략 현상, 조음 표현 등의 언어 현상은 문맥이나 가상 세계의 상황을 참고하지 않으면 의미를 바르게 분석해 낼 수 없다.

구조적 중의성은 주로 병렬문과 부사, 그리고 양화사와 관련되어 있다. 중의적인 표현들에 의해 한 문장이 여러 가지 의미로 해석되는데 의미에 따라 애니메이션 결과가 판이하게 달라지게 되므로 문맥에 맞게 분석하는 것이 필요하게 된다. 다음의 예는 각각 병렬문과 부사, 양화사에 의한 구조적 중의성이 나타나는 문장들이다.

- (15) 그 곳에는 치르치르의 할아버지와 할머니께서 살고 계셨어요.
- (16) 언제나 등에 짐을 잔뜩 싣고 다니는 당나귀가 있었어요.
- (17) 어느 날 산에 나무를 하러 간 꿍은이는 새끼 호랑이 세 마리가 꼬리에 하얀 형질을 달고 있는 것을 보았어요.

(15)는 밑줄 친 부분이 “[치르치르의 할아버지]와 [할머니]” 또는 “치르치르의 [할아버지]와 [할머니]” 두 가지 해석이 모두 가능하다. (16)에서 부사 ‘언제나’는 “싣고 다니는”과 ‘있었어요’를 모두 수식할 수 있다. ‘언제나’가 “싣고 다니는”을 수식한다면 어떤 당나귀가 있어서 그 당나귀가 언제나 짐을 잔뜩 싣고 다닌다는 의미가 되고, ‘있었어요’를 수식한다면 어느 장소에 짐을 잔뜩 싣고 다니는 당나귀가 끊이지 않고 존재한다는 의미가 되어 수식 구조에 따라 여러 가지 해석이 가능하다. (17)에서 “하얀 형질”을 각자 호랑이가 가지는 것인지 하나를 함께 가지는 것인지 상

황에 따라 달리 해석할 수 있다.

또한 한국어에 특징적으로 자주 나타나는 언어현상 중 복수 표지의 생략 현상과 명령의 실행에 필요한 성분의 생략 현상 등은 멀티 동화를 제작할 때 가상 세계의 참조를 결정하기 위해 해결해야 하는 문제들이다.

(18) 일곱 형제는 밤새도록 큰 돌을 날라다 개울에 징검다리를 놓았지.

(19) 농부는 커다란 그물을 사다가 과일 나무 위에 쳤어요. 그리고 그는 가만히 멀리서 지켜보았죠.

(18)에서 ‘돌’은 단수형으로 쓰였지만 상황에 따라서는 한 사람이 여러 개의 돌을 날라야 하는 경우도 가능하므로 복수의 의미로 해석되어야 할 때도 있다. (19)에서는 두 번째 문장에서 주어와 목적어가 생략되어 있다. 동작이나 상태 변화의 지시에 꼭 필요하지만 문장에서 생략되는 성분들은 주어와 목적어를 비롯하여 보어, 장소 부사어 등 다양하다. 이렇게 중의적인 표현이나 생략 현상을 상황에 맞게 처리해야 동화의 의도에 부합하는 애니메이션을 생성할 수 있다.

4. 처리

본 장에서는 결합범주문법을 이용하여 동화의 의미를 분석하고 캐릭터를 제어하는 명령들을 추출하는 방법에 대해 논의한다. 먼저 분석에 사용할 도구인 결합범주문법을 간략하게 소개한 다음에 이를 동화 도메인에 적용하는 방법을 살펴본다.

4.1 결합범주문법

결합범주문법은 범주문법에 결합자를 접목시킨 문법 체계이다. 많은 문법체계에서는 하나로 합쳐질 수 없는 문장 성분들이 결합범주문법에서는 하나로 결합될 수 있기 때문에 병렬 구조 등의 복잡한 문형들을 별도의 제약조건 없이 처리할 수 있다는 장점이 있다. 그리고 어휘 범주에 통사 정보 외에 의미 정보, 담화 정보 등을 추가하면 축약 규칙³들을 통해서 통사, 의미, 담화 등의 여러 단계의 분석을 한 번의 유도과정으로 해결할 수 있다. 그림 2는 결합범주문법으로 간단한 문장을 처리하는 유도 과정을 보인 것으로, 통사와 의미 분석이 동시에 이루어짐을 보여준다.

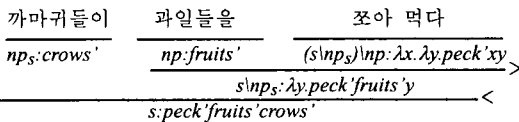


그림 2. 결합범주문법에서의 유도과정

³ 결합범주문법의 축약 규칙에는 Application, Coordination, Composition, Type-raising 등이 있다[7].

4.2 어휘에 할당되는 범주들

본 연구에서 각 어휘에 할당하는 범주는 (20)과 (21)의 형태를 갖는 np와 s들로 구성되어 있다.

(20) np(격,명사구ID):(FEATURES,스크립트)

(21) s(동작성/상대성):스크립트

동화 문장의 의미 기술의 기본 단위가 되는 각 단위 명령의 의미는 술어논항구조의 형태와 유사하게 술어와 명령의 실행에 직접적으로 필요한 요소들을 괄호 안에 나열한 형식으로 표현되는데, (20)과 (21)에 포함되어 있는 스크립트는 이러한 형태의 단위 명령들을 시간 순서에 맞게 나열한 것이다. FEATURES는 다의어의 문맥 상 의미를 분별해내기 위해서 사용되는 자질들의 집합이다⁴. 예를 들어 “과일이 익다”의 각 어휘에 할당되는 범주와 파싱 결과는 다음과 같다.⁵

(22) lex(‘과일’ , np(origin,[ID]):(f(ripen=fruit),[])) :- get_ID(fruit@1, ID).

(23) lex(‘이’ , np(subj,W):Semantics,np(origin,W):Semantics).

(24) lex(‘익다’ , s(verb):[doinorder,S1,ripen(Subj)] np(subj,Subj):(f(ripen=fruit),S1)).

(25) s(verb):[doinorder,[],ripen(id(1))]

(26) ripen(id(1))

(22)~(24)의 범주들에 결합범주문법의 축약 규칙들을 적용하여 파싱한 결과가 (25)이고 정리하면 (26)으로 간단히 표현된다.⁶ ‘익다’는 다의어로서 문맥에 따라 “열매가 자라서 완전하게 되다”, “날것이 먹을 수 있는 상태가 되다”, “맛이 들다”, “살이 벌겍게 되다”, “서투르지 않다” 등의 다양한 의미를 가지므로 문맥에 맞게 의미를 결정하기 위해서 과일과 결합할 때에만 “열매가 자라서 완전하게 되다”의 의미를 가지도록 ripen이라는 자질을 설정하였다. ripen 외에도 몇 가지 자질들이 있어서 문맥에 맞게 다의어의 의미를 결정하도록 하였다.

4.3 캐릭터 제어 명령 추출을 위한 처리

본 절에서는 3장에서 언급한 멀티 동화 제작 시 해결해야 할 문제점들의 처리에 대해 살펴보도록 한다. 본 연구에서는 동작 및 감정 처리 엔진을 따로 구현하거나 동작 라이브러리를 사용하지 않기 때문에 캐릭터들의

⁴ 파싱 이후에 다의어의 문맥에 따른 의미를 분별하는 방법도 있으나, 본 연구에서는 파싱 과정에서 의미를 결정하는 방법을 선택하였다.

⁵ 본 연구에서는 형태소 분석기를 별도로 사용하지 않기 때문에 조사가 붙은 어절을 제외하고는 모두 어절 단위로 범주를 할당한다.

⁶ f(a=A,b=B,...)는 feature들을 하나로 묶어놓은 것이며 괄호 안에 들어가는 feature들은 순서에 관계없다. 연산자 @은 TYPE@NUM으로 사용되어서 TYPE형태의 가상 세계 참조들 중 NUM개를 선택한다는 의미로 사용된다. get_ID는 명사에 부여한 고유한 이름을 만드는 루틴이다.

감정이나 내부적 움직임보다는 주로 캐릭터들의 위치가 변화되는 동화가 대상이 된다.

4.3.1 명령들 간 시간관계 결정

3.1.1절에서 언급한 바와 같이 한 문장에 포함된 여러 명령들 간의 시간관계는 대부분의 경우 시간 부사구와 연결 어미에 의해 결정될 수 있다. 이들이 명백한 동시 발생을 의미하는 경우를 제외하고는 모두 순차 발생으로 해석해도 멀티 동화에서는 큰 무리가 없다고 보고, 시간 관계를 나타내는 시간 부사구와 연결 어미 등에 동시 또는 순차 발생의 의미가 부여되도록 범주를 할당한다.

4.3.2 사물의 위치 결정

문장에서 추상적으로 제시된 사물들 간 위치 관계를 정량적 수치로 변환할 때에는 [1]에서와 같이 기본 설정값을 사용한다. 예를 들어, “~의 옆”이라는 위치 관계는 대상이 되는 사물에 관계없이 미리 정해진 일정한 간격만큼 왼쪽 또는 오른쪽으로 떨어져 있는 것으로 해석한다.

사물의 위치가 문장에서 생략되었을 때 이를 상식으로부터 추론할 수 있는 경우의 처리를 위해 이런 상식을 객체속성 사전에 축적해 놓는다.

본 연구에서는 둘 이상의 사물이 겹쳐져 있는지의 여부를 판단하는 충돌 탐지는 단순하게 사물의 중심좌표가 일치하는지의 여부를 판단하는 것으로 처리하도록 한다. 재배치할 수 있는 공간의 범위를 찾을 때 사물의 속성을 고려하는 경우가 있다고 하였는데, 이것은 사물 고유의 속성이므로 객체속성 사전에 포함시킨다. 재배치될 수 있는 공간의 범위를 찾은 다음에는 그 공간 내에서 임의의 간격만큼 임의의 방향으로 옮기도록 한다.

4.3.3 캐릭터가 실행할 세부 명령의 결정

동작이 실행되기 전에 만족되어야 할 전제조건이 현재 상황에서 충족되고 있는지의 여부를 조사하여 만약 전제조건이 만족되지 않은 경우에는 추가 동작을 명령한다. 예를 들어, 까마귀가 과일을 쪼아 먹는 동작이 실행되기 위해서는 까마귀는 과일의 근처에 있어야 하므로 이 전제조건이 만족되지 않은 경우에는 과일의 근처로 오도록 명령을 추가해야 한다.

이동의 기본 수단에 대한 정보는 사물 고유의 속성이므로 객체속성 사전에 포함시킨다.

목적지까지의 이동 행로를 찾는 문제는 본 연구에서는 이동 행로를 따로 계산하지 않고 주위 환경에 관계없이 무조건 직선 방향으로 진행하는 것으로 처리하도록 한다.

5. 구현

본 연구에서 구현한 멀티 동화 제작 시스템의 전체적인

구조는 다음과 같다⁷. 본 시스템은 크게 문장분석기와 번역기로 구성된 자연언어처리 모듈과 애니메이션 모듈로 나뉜다.

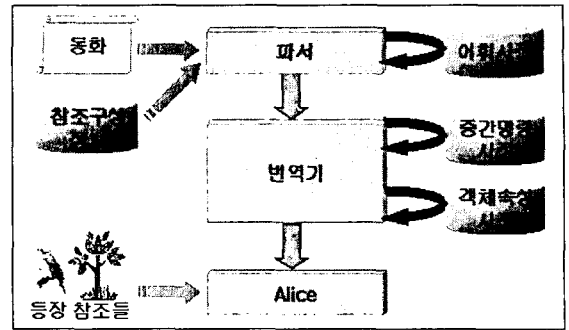


그림 3. 시스템의 전체적인 구조

사용자는 먼저 동화에 등장하는 참조들을 Alice⁸에 생성해놓고 이들 참조들의 구성 정보와 동화 텍스트를 시스템에 넘겨준다. 번역기에서는 자연언어 문장의 파싱 결과로 생성되는 중간수준 명령들⁹을 Alice 에서 실행 가능한 형태의 스크립트로 변환한다. 번역하는 과정에서는 사용자로부터 넘겨받은 참조 구성 정보를 사용하여 참조를 해결하고, 중간명령 사전과 객체속성 사전을 이용하여 각 객체의 특성에 맞는 애니메이션 스크립트를 완성한다. 완성된 스크립트는 Alice 를 통해서 애니메이션으로 보여준다. 그림 4 는 이 과정을 거쳐서 생성된 애니메이션의 한 장면이다. 파일들과 까마귀들은 애니메이션 도중에 화면에 등장하는 것이므로 사물들끼리 겹쳐지지 않도록 각자의 위치를 정해주어야 하는데, 본 연구에서는 충돌 탐지를 정식으로 처리하지 않고 재배치 시 임의의 간격만큼 임의의 방향으로 옮기는 것으로 구현되어 있기 때문에 보기에 자연스럽게 보이는 부분들이 생성되기도 한다.

본 연구에서 사용된 데이터로는 동화 문장들에 대한 어휘사전과 이 어휘사전에 쓰인 중간명령들을 번역하기 위한 중간명령 사전이 구축되었고 각 객체의 구현에 필요한 속성들을 사전에 기술하였다. 중간명령 사전에는 ‘위치시키다’, ‘익다’, ‘일하다’, ‘농사짓다’, ‘쪼다’, ‘움직이다’ 등을 포함한 중간명령들이 들어있으며, 객체속성 사전에는 각 객체의 기본 이동 수단이나 기본 이동 속도, 계절에 따른 변화 등을 포함한

⁷ 자연언어처리 모듈인 파서와 번역기는 SICStus Prolog로 작성되었으며, 애니메이션 모듈은 Alice라는 공개프로그램을 사용하고 있다. (<http://www.alice.org>)

⁸ Alice는 Carnegie Mellon 대학에서 제작한 삼차원 애니메이션 툴이다[9]. 내부적인 동작 라이브러리는 구현되어 있지 않으며, 캐릭터와 카메라 등을 기본적인 수준으로 제어할 수 있는 스크립트 명령을 받을 제공한다.

⁹ 중간명령은 4장에서 언급한 바와 같이 숨어논항구조와 유사한 형태로 이루어져 있다.

객체의 특성들이 들어있다.

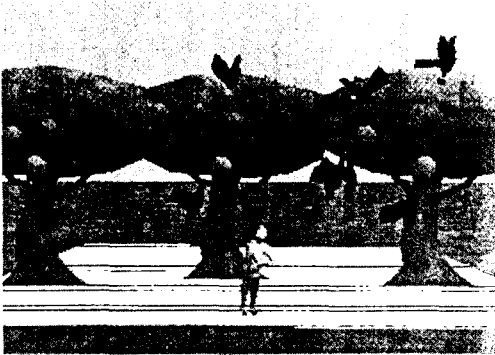


그림 4. “잘 익은 과일들을 까마귀들이 쪼아 먹는 거예요”

Conversation, 2000.

[5] Kenneth Wauchope, “Eucalyptus: Integrating Natural Language Input with a Graphical User Interface”, Technical Report, Naval Research Laboratory, 1994.

[6] 김진석, “사건 간의 시간 해석”, Korean Journal of Linguistics, v.22, no.1, pp.27-55, 1995.

[7] Mark Steedman, The Syntactic Process, The MIT Press, 2000.

6. 결론 및 향후계획

본 논문에서는 결합범주문법을 이용하여 텍스트 형태의 한국어 동화를 분석한 후 삼차원 애니메이션으로 보여주는 멀티 동화 제작 시스템을 설계 및 구현하였다. 멀티 동화 제작 시스템을 제작할 때 해결되어야 하는 문제점들 및 결합범주문법을 이용한 처리 방법, 그리고 구현된 시스템을 살펴보았다.

본 논문에서는 병렬 문형이나 부사 등의 처리는 고려하지 않고 비교적 간단한 문형들만 고려했으므로 앞으로 이러한 복잡한 문형들에 대한 처리가 필요하다. 또한 현재까지는 문맥을 고려하지 않았는데 문맥을 참고하여 해결할 수 있는 문제들과 관련된 언어 현상들의 처리가 이루어져야 한다. 그리고 이미 구축된 지식 사전에 앞으로 구현에 필요하게 될 객체의 속성이나 상식 등의 지식을 추가 기술해나가는 작업도 필요하다.

7. 참고 문헌

[1] Bob Coyne and Richard Sproat, “WordsEye: An Automatic Text-to-Scene Conversion System”, SIGGRAPH, 2001.

[2] Rama Bindiganavale et al., “Dynamically Altering Agent Behaviors Using Natural Language Instructions”, Autonomous Agents, 2000.

[3] Marc Cavazza et al., “Situating AI in Video Games: Integrating NLP, Path Planning and 3D Animation”, AAAI symposium on AICG, 1999.

[4] Yusuke Shinyama et al., “Kairai - Software Robots Understanding Natural Language”, 3rd International Workshop on Human-Computer