

언어함수를 이용한 영문 생성기의 구현에 관한 연구

A study on Implementation of English Sentence Generator using Lexical Functions

정 희 연*
Hee-Youn Jung

김 희 연**
Hee-youn Kim

이 응 재**
Woong-Jae Lee

요 약

컴퓨터의 발달과 인터넷 사용자의 증대로 자연어 처리의 연구에 관한 관심이 증대되고 있다. 그러나 대부분의 연구가 자연어 분석 및 이해에 집중되고 있어 자연어 생성에 관한 연구는 주목을 받지 못해 왔으며 자연어 생성을 자연어 분석의 역과정으로 간단하게 생각하는 경향마저도 있다. 하지만 Web상에서의 다국어간 번역, 자연어 인터페이스, 자연어 검색 시스템 등 자연어처리에 관한 필요성이 증가함에 따라 자연어 생성의 필요성도 자연히 증가하고 있는 실정이며 좀 더 체계적인 자연어 생성 시스템 개발을 위해서는 자연어 생성에 관한 보다 구체적인 알고리즘에 관한 연구가 필요하다. 본 논문에서는 영문 생성에 있어서 보다 자연스러운 문장을 생성하기 위한 알고리즘을 제안하며 특히 Igor Mel'uk (Mel'uk & Zholkovsky, 1988)의 어휘 함수(LFs)를 이용한 어휘 결합을 통하여 절(節) 길이의 설명문을 생성하는 영문 생성기의 구현에 대하여 논한다.

Abstract

The majority of work done to date on natural language processing has focused on analysis and understanding of language, thus natural language generation had been relatively less attention than understanding. And people even tends to regard natural language generation as a simple reverse process of language understanding. However, need for natural language generation is growing rapidly as application systems, especially multi-language machine translation systems on the web, natural language interface systems, natural language query systems need more complex messages to generate. In this paper, we propose an algorithm to generate more flexible and natural sentence using lexical functions of Igor Mel'uk (Mel'uk & Zholkovsky, 1988) and systemic grammar.

1. 서 론

컴퓨터의 발달과 인터넷 사용자의 증대로 기계 번역, 자연어 인터페이스, 자연어 질의 시스템 등 자연어 처리의 연구에 관한 관심이 증대되고 있다. 그러나 대부분의 연구가 자연어 분석 및 이해에 집중되고 있을 뿐만 아니라 자연어 생성을 자연어 분석의 역과정으로 간단하게 생각하는 경향

이 있다. 그렇기 때문에 좀 더 체계적인 생성 시스템 개발을 위해서는 생성에 관한 보다 구체적인 생성 알고리즘에 관한 연구가 필요하다. 또한 어휘 결합(Lexical Cohesion)을 사용한 문장들이 그렇지 않은 것들보다 이해하기 쉬우며, 질적으로도 우수하다는 실험들을 자주 볼 수 있다 (Britton & Black, 1985). 본 논문에서는 시스템적 문법의 개괄적인 설명과 함께 시스템적 문법과 인공지능의 문제 해결방법을 비교하였으며 이 개념을 이용한 영어 문장 생성 알고리즘과 Halliday & Hasan (1976)에 의해 분류된 cohesion중 어휘결합(lexical cohesion)을 이용하여 절 길이의 문장을 생성하는 영문 생성기의 구현에 대하여 논한다. 어휘결합(lexical cohesion)은 반복(reiteration)과 연어(連語,

* 서울여자대학교 컴퓨터학과 석사과정
comjung@swu.ac.kr

** 서울여자대학교 컴퓨터학과 석사과정
hylakim@hanmail.net

*** 서울여대 컴퓨터학과 부교수
wjlee@swu.ac.kr

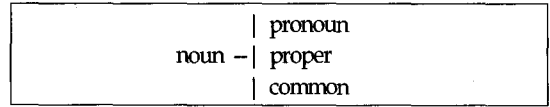
collocation)의 두 범주로 나눌 수 있다. 반복은 원래의 개념을 나타내는 대명사 혹은 한정 명사절의 사용으로 나타난다. 따라서 이는 **Gener, Syn, Conv, Anti**, 그리고 **Dimin**과 같은 언어함수를 사용한다. 예를들면, **Gener(Wafarin) = anticoagulant**와 같은 함수의 사용이다. 이에 반해 연어(連語, collocation)는 관련된 명사, 동사, 형용사 또는 부사의 사용을 나타낸다. 따라서 이는 **Real, Magn, Cap, Equip, Centr, Pred, IncepPredPlus**, 그리고 **IncepPredMinus**와 같은 언어함수의 사용이 포함된다. 본 논문에서는 *part-whole*과 같이 다분히 의미적인 언어함수를 문장 생성기에서도 같은 방법으로 사용하고자 시도하였다.

2. 시스텍 문법

시스텍 문법(systemic Grammar)은 M.A.K. Halliday에 의해 영국에서 개발된 언어학 이론으로서 언어의 구조를 기술하는 데 중점을 둔 기존의 구구조 문법들과는 확연히 다른 출발 동기를 가지고 있다. 즉, 언어를 올바르게 분석하기 위해서는 언어 자체와 그 기능적 요소, 사회적 정황이나 환경들을 분리할 수 없다는 개념에서 출발되었으며 시스텍 문법의 개발 동기가 되었던 질문들은 다음과 같다. (1) 언어의 사회적 기능은 어떤 것인가? (2) 언어는 이러한 사회적 기능들을 어떻게 만족시키는가? (3) 언어는 어떻게 작용하는가? (Winograd 1983) 이러한 생각들은 어떤 주어진 상황에서 취할 수 있는 언어 선택들의 집합인 시스템(system)이라는 개념을 도출해내었다.

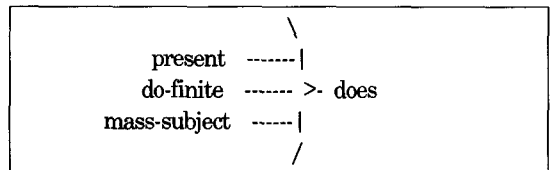
2.1 시스템, 게이트, 시스템 네트워크(System, Gate, and System Network)

시스템이란 하나의 입장 조건을 가지고 있는 선택항(자질)들의 집합으로 이들은 상호 배제적이며 입력 조건이 만족되는 경우 집합의 원소 중 하나가 선택되어 진다.



(그림 2.1) 간단한 시스템의 예

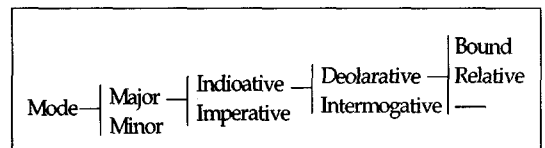
그림 2.1의 시스템은 만일 명사 자질이 선택되면 그것은 대명사, 고유명사 혹은 일 반명사 중 하나이어야 한다는 것을 의미하며 그 입장 조건은 경우에 따라 여러 자질의 논리적 조합인 경우도 있다.



(그림 2.2) 게이트(Gate)

그림 2.2는 만일 presents, do-finite, 그리고 mass-subject 자질이 선택되었으면 does 자질을 취하라는 것을 의미한다. 위의 시스템은 조합 입장 조건을 가지고 있다는 특징외에도 입장 조건이 만족되었을 때 취할 수 있는 자질에 선택의 여지가 없다는 뜻을 가지고 있다. 이러한 시스템을 게이트(Gate)라고 부른다. 시스템 네트워크란 문법을 구성하는 시스템과 게이트들 간의 상호 의존적 관계를 나타내는 그래프적 표현으로서 예를 들어 영어의 mood(양상성) 시스템을 포함하는 절 (clause) 네트워크는 그림 2.3과 같은 트리구조로 표현 될 수 있다.

따라서 문장 'It was Dr. Kim who performed the surgery'에서 절 'who performed the surgery'는 Major Indicative Declarative Relative로 분류 될 수 있다.



(그림 2-3) 영어의 Mood System의 예

2.2 언어의 기능 분석 (Functional Analysis)

문장을 구성하고 있는 원소의 역할을 분석하여 보면 어떤 관점에서 분석하는가에 따라 여러 가지 기능을 하는 것을 알 수 있으며 이들을 동시에 다차원적으로 분석해야 할 필요가 생기게 된다. Halliday는 이를 위해서 mood(양상성), Transitivity(상호작용성), Theme(주제성), 그리고 Information(정보성)/Ergativity(능동성) 4개의 계층을 도입하였다. 본 연구에서는 Information을 제외한 3 계층에서의 분석만을 논한다

2.2.1 상호작용성 분석(Analysis of Transitivity)

상호작용성 분석은 행위에 대한 분석으로서 행위를 개념적 차원에서 즉, 행위에 대한 역할이라는 관점에서 분석하며 행위는 (1) 행위 자체, (2) 행위에 참가하는 대상, (3) 행위가 수행되는 환경, 3개의 요소로 구성된다고 본다. 또한 행위의 종류에 따라 행위(Process), 행위자(Actor), 수혜자(Beneficiary), 목표(Goal) 등의 역할을 중심으로 문장을 분석한다. 그림 2.4는 문장의 상호작용성 분석을 보여준다.

Aspirin	is	Prescribed	for	Nary
Goal		Process		Beneficiary

(그림 2.4) 문장의 상호작용성 분석 (Analysis of Transitivity)

2.2.2 양상성 분석(Analysis of Mood)

양상성 분석은 발화자와 청취자 사이에 정보를 주고받는 형식을 중심으로 분석하는 것으로서 문장을 크게 Mood 와 Residue 라는 두개의 구조로 분석한다. Mood는 다시 주어부(Subject:문법적 주어)와 한정 요소(Finite elements)들로 나뉘어 지며 Residue 는 다시 술부(Predicator), 보어 (Complement), 와 부사구(Adjunct)로 나뉜다. 그림 2.5는 문장의 양상성 분석을 보여준다.

Probably Dr.Kim will perform the operation tomorrow					
djunct	Subject	Finite	Predicator	Complement	Adjunct
Mood			Residue		

(그림2.5) 문장의 양상성 분석 (Analysis of Mood)

2.2.3 주제성 분석(Analysis of Theme)

Theme 시스템은 Theme과 Rheme 의 두 부분으로 구성되어 있다. Theme은 문장이 시작되는 부분으로 일반적으로 문장의 주된 관심 부분을 의미하며 그 나머지 부분을 Residue라 부른다. 영어의 Theme-Rheme 구조에서는 그 위치에 따라 알 수 있으며 Theme 요소들이 항상 제일 앞에 위치하며 이는 다시 Textual, Interpersonal,과 Topical의 세 부분으로 확장된다. 그림 2.6은 문장의 주제성 분석을 보여준다.

Probably Dr Kim will perform the operation tomorrow		
Interpersonal	Topical	Rheme
Theme		

(그림 2.6) 문장의 주제성 분석

2.3 인식 규칙(Realization Rules)

2.2절에서는 문장 분석을 언어의 기능에 따라 3차원의 독립적인 계층으로 분류하였지만 이들 간에는 상호 관계가 존재하며 따라서 하나의 문장을 분석 할 때 이들을 동시에 분석해야 할 필요가 생긴다. 여기서 우리는 다차원적인 문장 분석을 하나의 결과로 묶어 줄 수 있는 필요성을 느끼게 되는데, 바로 인식 규칙(realization rule)을 통하여 이러한 작업을 수행 할 수 있다. 인식 규칙에 대한 범용적인 표기는 존재하지 않으므로 본 연구에서는 NIGEL [Mann 1985] 과 SLANG [Patten 1988]에 사용된 인식 규칙에 기반한 인식 규칙들을 사용한다. 이를 크게 나누면 다음과 같은 세 가지 형태로 분류 할 수 있다.

- (1) 구성 요소의 기능에 따라 문장 구조를 생성하는 규칙
- (2) 구성 요소의 배치 순서에 관한 규칙
- (3) 함수와 자질(feature)간의 관계에 관한 규칙

이들 중 대표적인 몇 개의 규칙으로는 문장의 구조를 생성하는 융합 규칙(conflation), 확장규칙(expansion), 문법적 기능과 자질(feature)간의 관계에 관한 규칙으로 선선택(preselect) 등을 들수 있다.

- 융합 규칙(conflation, 기호 ‘/’)

융합 규칙은 다른 시스템의 분석 결과에서 두 개의 다른 기능이 하나의 문장 구성 요소에 의해 인식되어 진다는 것을 의미한다. 예를 들면, 융합 규칙 (Goal/Subject)은 'Aspirin was prescribed by Dr. Kim.' 에서 구성소 'Aspirin'은 그림 2.7과 같이 Transitivity System 에서는 Medium 함수로, Mood System 에서는 Subject 함수로 인식되어 진다는 것을 의미한다.

Aspirin	was prescribed by Dr. Kim	
Transitivity	Medium	...
Mood	Subject	...

(그림 2.7) 융합 규칙의 예

- 확장 규칙(Expansion, 기호 ‘(와)’)

확장 규칙은 하나의 함수가 여러 개의 하부 함수로 확장되어 진다는 것을 의미한다. 예를 들면 확장 규칙 (Theme (Interpersonal))과 (Theme(Topical))은 그림 2.8과 같은 Theme 함수의 확장 구조를 말해준다.

Probably	Dr. Kim
Interpersonal	Topical
Theme		

(그림 2.8) 확장 규칙의 예

- 선선택 규칙(preselection, 기호 ‘:’)

선선택 규칙은 함수에 의해 인식된 구성요소보다 자기 보다 하위 범주에 속한 네트워크로부터 자질(feature)을 선택하도록 한다. 예를 들면 선선택 규칙 (finite : second-person)은 절 네트워크(clause network)의 함수 'Finite'가 하위 레벨인 동사 네트워크로부터 'second-person'의 자질을 선택해야 한다는 것을 의미한다. 이와 같이 선선택 규칙은 레벨이 서로 다른 네트워크간의 인터페이스 역할을 하므로 특별히 중요한 규칙이라 할 수 있다.

3. 시스템적 문법과 Problem Solving

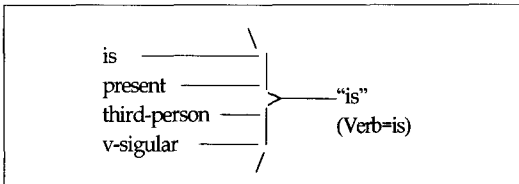
생성 규칙 시스템은 조건부와 실행부로 구성된 규칙들의 집합으로 이루어진다. 규칙의 조건부는 그 규칙이 적용되기 위해 만족해야하는 조건들을 의미하며 실행부는 취해야 할 행동들을 의미한다. 시스템이란 하나의 입장 조건을 가지고 있는 선택 사항들의 집합으로 각 선택 사항들은 인식 규칙들을 가지고 있으므로 일단 선택 사항이 선택 되면 첨부된 인식 규칙들이 실행되어진다. 여기서 우리는 생성 규칙의 조건부 와 시스템의 입장 조건사이에서, 또한 생성 규칙의 실행부와 시스템의 인식 규칙 사이에서 유사성을 발견 할 수 있으며 따라서 시스템적 문법이 생성 규칙으로 표현될 수 있다는 점을 생각할 수 있다.

인공지능의 세계에서 문제 해결이란 문제 영역에서의 탐색으로 귀결된다. 한편 시스템적 문법의 시스템은 입장 조건이 주어진 경우 택할 수 있는 의미적인 대안들의 집합으로 정의되며, 시스템적 문법의 순회는 이러한 대안들 중 문맥에 따라 적절한 선택을 하는 것으로 정의 할 수 있다. 우리는 여기서 인공지능의 탐색과 시스템적 문법의 순회간의 유사성을 다시 한번 볼 수 있다. 이러한 유사성은 시스템적 문법이 인공지능의 탐색 기법을 이용하여 구현 될 수 있다는 중요한 착안을

하도록 한다. 인공 지능 분야의 연구에 의해 많은 탐색 기법들이 개발되었으나 시스템의 문법의 구현에는 전향 연결 및 후향 연결을 이용한 탐색기법을 적용한다.

3.1 전향 연결과 Gates

전향 연결이란 조건부가 만족되는 규칙의 실행부가 실행되며 이로 인하여 조건부가 만족되는 새로운 규칙의 실행부가 실행되는 추론방식을 의미한다. 시스템의 문법의 게이트는 자질(feature)의 조합에 결과에 따라 하나만의 선택을 취할 수 있는 단순화 시스템으로 정의되었다. 이것은 바로 위의 전향 연결의 적용이 적당한 경우라 할 수 있다. 따라서 시스템의 문법의 게이트(gate)는 전향 연결의 형식으로 구현되었다. 예를 들면, 그림 3.1의 게이트는 그림 3.2와 같은 전향 연결 규칙으로 표현될 수 있다.



(그림 3.1) 게이트

```
/* rule */
IF (GOAL(unmaked-declarative-theme))
THEN (ADD_MEM(unmarked-declarative-theme)
      (CONFLATE(Topical, Subject)
      (ADD_GOAL(declarative)))
/* eor */
```

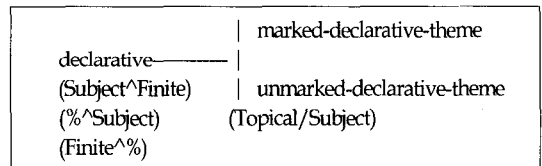
(그림 3.2) 전향 연결 규칙

3.2 후향 연결과 시스템(Systems)

상황에 따라서는 주어진 목표에서 출발하여 그 목표를 이루기 위한 조건들을 역으로 추적하는 경우가 있다. 즉, 추론 엔진은 주어진 목표와 실

행 부분이 일치하는 규칙을 찾는다. 이러한 규칙의 조건부에 아직 만족되지 않은 부분이 있으면 이것은 하위목표(subgoal)로 설정되며 이 하위목표를 같은 방법으로 해결해 나간다. 이러한 추론 방식을 후향 연결이라고 한다.

이러한 상황을 그림 3.3과 같은 시스템에 적용시켜 보면 'declarative' 자질은 후향 연결 규칙으로 해석 될 수 있다. 즉, unmarked-declarative-theme 자질이나 unmarked-declarative-theme 자질을 선택하는 것이 목표라면 Topical 함수와 Subject 함수를 융합하고, declarative 자질을 하위목표로 설정한다. 이것은 그림 3.4와 같은 규칙으로 표현될 수 있으며 따라서 시스템의 문법의 시스템은 후향 연결 규칙을 이용하여 구현한다.



(그림 3.3) declarative 시스템

```
/* rule */
IF(singular-pronoun .AND. third .AND.
   (subjective .OR. objective))
THEN(LEXIFY(Noun, it))
/* eor */
```

(그림 3.4) 후향 연결 규칙

4. 문장 생성기(Sentence Generator)

본 연구에서 구현한 문장 생성기는 생성 규칙으로 이루어진 시스템의 추론 엔진의 역할을 담당한다. 문장 생성기의 주된 임무는 문법 문제를 해결하기 위해 전향 연결과 후향 연결의 추론을 수행하는 것이며, 인식 규칙(realization rules)의 결과를 바탕으로 구문 구조를 생성해 나가는 것이다.

실질적인 생성 작업은 생성할 문장의 의미 목표(semantic goal)를 입력받는 것으로 시작된다. 문

```

prescription
  (Subject=ASK(DDTRTMNT))
  (Beneficiary=ASK(DDPNAME))
semantic - (Predicator : prescribe)
           (Prescription : unmarked-declarative-theme)
           (Prescription : singular-subject)
           :
/* rule */
IF (GOAL(prescription))
THEN
  (ADD_MEM(prescription))
  (LEXIFY(subject,ASK(DDTRTMNT)))
  (LEXIFY(Beneficiary,ASK(DDPNAME)))
  (PRESELECT(Predicator,prescribe))
  (PRESELECT(Prescription, unmarked-declarative-theme))
  (PRESELECT(Prescription, singular-subject))
           :
    
```

(그림 4.1) 의미 지식베이스를 위한 후향 연결 규칙의 예

장 생성기는 시스템의 문법, 사전, 의미 지식베이스와 연결되어 있으며 의미 지식베이스 또한 생성 규칙 시스템으로 구현되어 있다. 따라서 의미 지식베이스내의 관련 규칙들에 관한 추론을 통하여, 입력받은 의미 목표를 달성할 수 있으며 이는 의미 문제의 해결을 의미한다. 이때 사용된 규칙들에 첨부되어 있는 인식규칙들은 문법적 목표로 설정되며 이 단계에서 생성할 문장의 의미 목표가 문법적 목표로 바뀌게 된다. 예를 들면, 아래와 같은 의미 지식 베이스내의 의미 자질 'prescription'은 그림 4.1과 같은 후향 연결 규칙으로 해석될 수 있다.

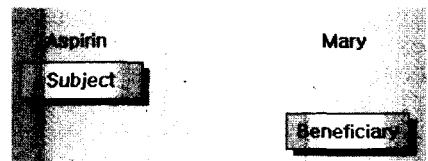
문장 생성기의 다음 임무는 문법 문제를 해결하기 위하여 문법 지식베이스를 탐색하는 것인데 이는 바로 생성 규칙으로 이루어진 시스템의 문법의 탐색을 의미한다. 이를 위한 전향, 후향 연결들의 추론을 실행하는 동안 규칙들에 첨부된 인식 규칙들이 실행되면 문장을 구성하는 어휘 요소들이 모두 결정된다. 즉, 문법 문제가 해결된 것이며 그 결과는 의미 목표, 문법 목표를 모두 만족시키는 문장의 생성이다. 자세한 문장 생성 과정을 다음절에서 볼 수 있다.

- 문장 생성의 예

다음의 예는 'Aspirin is prescribed for Mary'라는 문장을 생성 해나가는 과정을 보여준다. 문장 생성기는 스키마 구조를 통하여 전달된 의미 목표에 따라 문장을 생성하게 되는데 아래의 예는 그림 4.1의 의미 목표 'prescription'이 입력 된 경우의 예이다.

문장 생성의 첫 단계는 목표를 의미 지식베이스의 탐색을 통하여 주어진 의미 자질에 첨부된 인식 규칙들을 문법 목표로 설정하는 것으로서 본 예에서는 그림 4.1에 나타난 인식규칙들의 처리를 보여준다. 단계적으로 살펴보면, Transitivity 시스템에서 process는 prescribe로 설정되며 Subject와 Beneficiary는 각각 LEXIFY 규칙과 ASK() 함수에 따라 Aspirin과 Mary로 결정되어 진다. 현재 상태의 문장의 구조는 그림 4.2에 나타나 있다.

주어진 문법 목표 unmarked-declarative-theme과 non-attitudinal 로부터 문장 생성기는 후향 연결을 시작한다. 따라서 자질 declarative는 새로운 하위 문법 목표로 설정되며 그림 4.3에서 볼 수 있듯이 unmarked-declarative-theme 자질은 융합규칙 (Topical/Subject)를 가지고 있으므로 이 후향 연결의 실행은 Topical 함수와 Subject 함수가 융합되는 부수적인 효과를 얻게 되며 이 단계에서의 문장의 구조는 그림 4.4에 나타나 있다.

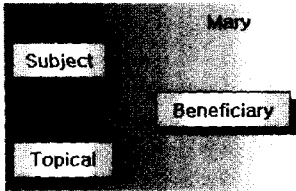


(그림 4.2) Subject와 Beneficiary의 결정

```

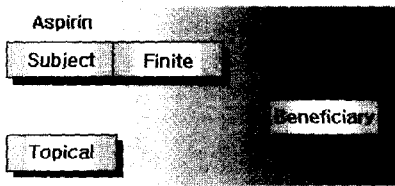
/ | marked-declarative-theme
- |
  | | unmarked-declarative-theme
declarative-< (Topical/Subject)
(Subject^Finite) | | non-attitudinal
(%^Subject) | - |
(Finite^%) \ | attitudinal
    
```

(그림 4.3) sample system



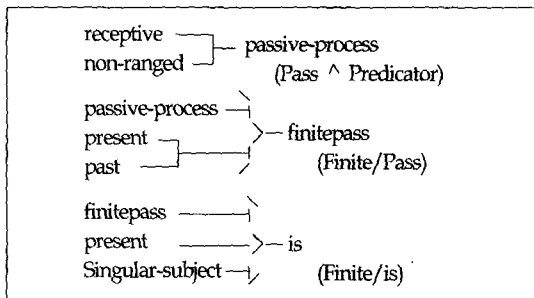
(그림 4.4) 융합규칙 적용

유사한 방법으로, 문장 생성기는 declarative와 present에서 indicative로의 후향 연결을 실행하며 그 효과는 인식 규칙 (Subject ^ Finite), (% ^ Subject), (Finite ^ %)의 실행이다. 이 단계에서 Subject와 Finite의 문장에서의 상대적인 위치가 결정되며 현재 상태의 문장의 구조는 그림 4.5에 나타나 있다.

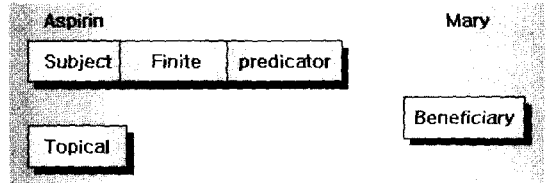


(그림 4.5) Finite 위치결정

한 시점에 이르면 agentive에서 receptive로의 후향 연결이 실행되며 이는 그림 4.6과 같이 연속적인 전향 연결의 실행을 유발한다. 따라서 (Pass ^ Predicate), (Finite / Pass), (Finite : is)의 인식 규칙이 실행되는 효과를 얻게 되며 현재 상태의 문장의 구조는 그림 4.7에 나타나 있다.

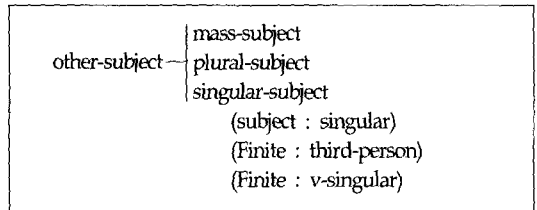


(그림 4.6) 전향 연결

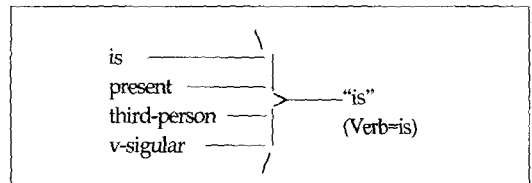


(그림 4.7) predicator 결정

이 단계에서 선선택 규칙 (Finite : is)에 의해 또 하나의 문법 목표가 설정된다. 한편 인식 규칙 (Finite : present)가 이미 문법 목표로 주어져 있으므로 함수 Finite는 'is'와 'presnet' 두개의 문법 목표를 갖게 된다. 또한 그림 4.8의 시스템을 따라 singular-subject에서 other-subject로 후향 연결을 실행하여 함수 Finite는 third-person과 v-singular라는 두개의 문법 목표를 추가하게 된다. 이 네개의 목표는 모두 동사 네트워크에 존재하므로 만족하게 되고 그림 4.9의 전향 연결 규칙(Gate)이 행되어 어휘 is가 결정된다.



(그림 4.8) subject 시스템



(그림 4.9) is 생성

이와 같은 반복적인 전향, 후향 연결의 실행은 최상위 레벨인 clause 자질이 만족될 때까지 계속되며 최종적으로 생성된 문장과 그 구조는 그림 4.10과 같다.

Aspirin		is Prescribed for		Man	
Subject	Finite	predicator	Residue		
Mood		Beneficiary			
Goal	Process		Beneficiary		
Topical	Rheume				
Theme					
Medium					

(그림 4.10) 최종결과

5. 참조 표현의 선택

일단 문장생성이 완료되면 좀 더 자연스러운 문장생성을 위해 어휘 대치가 수행된다. 어휘선택과 문법적인 선택 중 어느쪽이 먼저 고려되어야 하는지에 대해서는 그동안 많은 논쟁이 있었으나 본 시스템에서는 어휘선택이 문법선택보다 먼저 이루어지고 있다. Britton & Black (1985), 그외의 많은 연구들은 서술식 문장에서 참조 표현 방법으로 대명사 보다는 명사구의 사용을 선호한다는 결과를 보여준다. 더구나 적절한 참조 명사구의 선택에서 언어함수의 역할은 대단히 중요하다 한편, 주어 명사구를 생성함에 있어 시스템은 복용법과 아스피린에 관한 언어함수가 필요하다는 점을 인식한다. 이와 관련된 언어함수들은

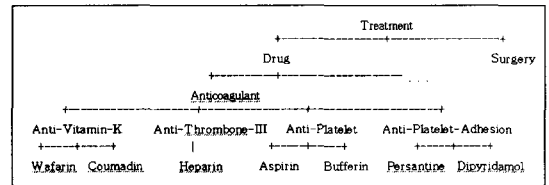
- Sing(drug) = dose
- Mult(dose) = dosage
- Gener(Aspirin) = drug

등이다. 시스템은 $dose = \text{Sing}(\text{Aspirin})$ 이라는 결론에 도달하며 따라서 우리는 Aspirin, dose 그리고 dosage 간의 연어 관계(collocational relationship)를 발견할 수 있다. 그 결과로 dosage 앞에는 정관사가 나열되며 이로 인하여 독자는 이것이 새로운 정보가 아니라는 것을 인식하게 된다. 따라서 시스템은 "dosage of Aspirin"과 같은 전치사구를 생성하지 않는다. 현재 대화의 초점(focus)을 갖는 다른 약품이 존재하지 않으므로 다음과 같은 2번

째 문장이 생성된다.

(1) The dosage was 325 mg., the average oral dose.

다음 생성될 문장은 아스피린의 적용 증상을 포함한다. 여기서 아스피린이라는 단어는 아직 문맥상에 존재 할 만큼 가까운 범위 내에서 이미 사용되었으므로 단어 대치의 사용이 선호된다. 가장 단순한 대치로는 상위 개념의 용어로의 대치를 들 수 있다. 상위개념 관계는 다른 언어 함수들과 마찬가지로 Gener 계층구조를 이용하여 시스템 내부에 저장되어 있으며 그 일부분을 그림 5.1에서 트리 형식으로 보이고 있다.



(그림 5.1) Gener 계층 구조의 예

상위 개념 대치는 용어의 보다 일반적인 개념을 갖는 명사나 구의 대치를 의미한다. 예를 들면 그림 5.1 에서처럼 Warfarin의 상위 개념은 Anti-vitamin-K, Anti-Vitamin-K의 상위 개념은 Anticoagulant, Anticoagulant의 상위 개념은 drug, 그리고 drug의 상위 개념은 treatment이다. 한 용어에 관해서 계층구조상 최상위와 최하위 레벨간에 존재하는 여러 개의 상위 개념이 존재할 수 있다.

상위 개념 대치의 가장 큰 문제는 애매성을 유발할 수 있다는 것이다. 예를 들어, Warfarin이 이미 언급된 상황에서 Aspirin에 관한 상위 개념 대치를 수행하려 한다고 가정하자. 이 경우 (AVK, Anticoagulant, drug, treatment) 라는 Warfarin의 상위 개념 리스트가 이미 존재하므로 Aspirin에 관한 Anticoagulant나 drug과 같은 대치는 애매한 참조를 유발하게 된다. 그러나 현재까지 생성한 문장에서는 아직 다른 약품이 언급되지 않았으므로 다음과 같은 문장을 생성 할 수 있다.

(1) *This Anticoagulant is indicated in cerebral embolism of cardiac origin*

마지막 문장에서는 대명사 *it*의 대치를 사용하여 결국 전체 문장은 다음과 같다. *Aspirin was prescribed for Mrs. Washington. The dosage was 325 mg., the average oral dose. This Anticoagulant is indicated in cerebral embolism of cardiac origin. It is contraindicated in peptic ulcer.*

6. 적절한 연어(Collocation)의 생성

Myers (1991)는 과학 서적에서의 적절한 연어 사용의 중요성에 대해 논하였으며 Justeson & Katz (1991)는 부사와 동사의 연어사용에 있어 반의어의 중요성에 대하여 지적한 바 있다. 본 시스템의 경우, 아스피린이 체온에 미치는 영향을 설명하기 위해 생성기는 온도에 관한 **IncepPredMinus** 동사들을 찾아본다. 그 결과 **IncepPredMinus** (temperature) = [to] decrease, [to] go down, [to] fall, or [to] drop 등이라는 것을 알 수 있으며 다음과 같이 (4)-a, (4)-b의 문장을 생성하게 되지만 (4)-c와 같은 문장을 생성하지는 않는다.

(4)-a. *If the radius of the blood vessels in the skin increases then the temperature of the body decreases.*

(4)-b. *If the radius of the blood vessels in the skin goes up then the temperature of the body goes down.*

(4)-c. **If the radius of the blood vessels increases then the temperature falls.*

“if-then”으로 표현된 원인-결과 관계 때문에 생성기는 동사 [to] cause를 사용하여 원인 관계를 보다 명확히 표현하려 할 수 있다. 이런 경우, 생성기는 “if”-와 “then”- 절의 동사에 관한 적절한 명사의 표현을 위하여 명사화 함수 **Si**를 사용한다. 언어함수 **IncepPredPlus**(blood vessel) = [to] dilate

이며 **So**(dilate) = dilation이므로 다음과 같은 문장 (4)-d를 생성할 수 있다.

(4)-d. *Dilation of the blood vessels in the skin causes the temperature of the body to drop.*

또한 언어 함수

CausePredMinus(temperature) = [to] reduce를 이용하면 다음과 같은 문장(4)-e를 생성할 수 있다.

(4)-e. *Dilation of the blood vessels in the skin reduces the temperature of the body.*

만일 아스피린의 투여에도 불구하고 환자의 상태가 더욱 악화된다면 환자는 더욱 적극적인 치료를 필요로 하며 생성기는 이를 설명하여야 한다. 본 언어 영역에서는 **Degrad**(status) = [to] deteriorate 이므로 생성기는 다음과 같은 문장을 생성하게 된다.

(5) *The staff determined to attempt surgical intervention, since the patient's neurological status was deteriorating.*

본 시스템에서는 기존의 언어 함수 이외에, 특별히 의학 영역에 적합한 언어 함수를 추가하였다. 이는 언어 함수 **Sympt**로서 장애와 그 증상간의 관계를 나타내며 의학 영역에서는 유용하게 쓰일 수 있다. 또한 **Sympt**(intracerebral hemorrhage) = vomiting, **Sympt**(intracerebral hemorrhage) = sudden severe headache라는 것을 알고 있으므로 우리는 장애에 관해 언급하고 그의 전형적인 증상에 대해 언급하는 문장은 물론, 증상을 먼저 언급하고 그와 관련된 장애에 관해 언급하는 문장도 생성할 수 있다. 뇌졸중 환자는 치료 후에도 반복적이고 지속적인 구토에 시달릴 수 있다. 생성기가 (6)-a와 같은 문장을 생성할 때 언어 함수 **Magn**을 이용하여 **Magn**(vomitting) = severe라는 정보를 추가할 수 있다. 따라서 생성기는 (6)-b와 같은 문장을 생성한다.

- (6)-a. *The patient suffered from big vomiting.*
 (6)-b. *The patient suffered from severe vomiting.*

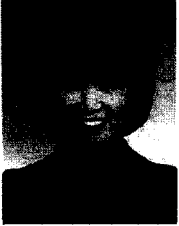
7. 결 론

시스템릭 문법을 이용하여 영어 문장을 생성하는 알고리즘, 특히 생성 규칙 시스템을 이용하여 시스템릭 문법을 구현하는 방법과 전향 연결 및 후향 연결의 추론 방식을 이용하여 문장을 생성하는 알고리즘에 관한 논의와 어휘 결합은 자연스러운 서술식 문장의 생성에 필수적이기 때문에 언어함수는 어휘적으로 연결된 문장을 생성하는데 있어 근본적인 기능을 제공하며 이에는 참조 표현의 선택과 적절한 언어의 구성 기능 등이 포함된다. 본 연구에서 구현한 시스템은 어휘적으로 잘 결합된 문장의 생성에 있어서도 언어 함수의 사용이 매우 효과적이라는 것을 보여 주었다.

참 고 문 헌

- [1] Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics. Stanford: Association for computational Linguistics, 201-206, 1987.
- [2] Patten, T. Systemic Text Generation as Problem Solving, Cambridge University Press, Cambridge, 1988.
- [3] Makoto Nagao. "자연언어처리" Britton, B, & J. Black. Understanding Expository text. Hillsdale:
- [4] Hovy, E.H. Generating Natural Language under Pragmatic Constraints. Hillsdale: Lawrence Erlbaum Associates, 1988.
- [5] Justeson, J. S. & S. M. Katz. "Co-occurrences of Antonymous Adjectives and Their Contexts", Computational Linguistics, 17.1:1-20. 1991.
- [6] Lee, W. & M.W. Evens "Generating Cohesive Text Using Systemic Grammar and Lexical Relations", Proceedings of the 3rd Midwest Artificial Intelligence and Cognitive Science Society Conference, 41-45, 1991.
- [7] Matthiessen, C.M.I.M., "Lexico(grammaral) Choice in Text Generation", Paris *et al.*, 242-292, 1991.
- [8] McDonald, D.D. "On the Place of Words in the Generation Process", Paris *et al.*, 229-247. 1991.
- [9] Myers G. "Lexical Cohesion and Specialized Knowledge in Science and Popular Science Texts", Discourse Processes, 14.1:1-26. 1991.
- [10] Stede, M. "Lexical Choice Criteria in Language Generation", Proceedings of the 5th Conference of the European Chapter of the Association for Computational Linguistics, Utrecht: Association for Computational Linguistics, 454-459, 1993.
- [11] Mel' uk, I.A. Lexical Functions: A Tool for the Description of Lexical Relations in a Lexicon. Lexical Functions in Lexicography and Natural Language Processing. ed by L. Wanner. John Benjamins Publishing Company, 37-102, 1996.

● 저자 소개 ●



정 회 연

1997년 서울여자대학교 컴퓨터학과 졸업(학사)

2001년 서울여자대학교 컴퓨터학과 석사과정

관심분야 : 인공지능, 자연어처리, 음성인식, 멀티미디어, 소프트웨어공학, 통신



김 회 연

1997년 서울여자대학교 컴퓨터학과 졸업(학사)

2001년 서울여자대학교 컴퓨터학과 석사과정

관심분야 : 인공지능, 자연어처리, 음성인식, 멀티미디어, 네트워크, 통신



이 응 재

1981년 연세대학교 공과대학 전기공학 학사

1985년 University of Illinois 전산학 석사

1991년 Illinois Institute of Technology 전산학 박사

1997년~현재 서울여대 컴퓨터학과 부교수

관심분야 : 인공지능, 자연어처리, 음성인식, 멀티미디어, GIS