

자연언어처리 및 인터페이스를 위한 기술

우 요 섭*, 최 병 욱**

(*인천대 정보통신공학과 **한양대 전자통신공학과)

■ 차

■ 례 ■

I. 서언

II. 자연언어처리 동향

III. 자연언어 해석 기법

IV. 연구 경험

V. 결언

I. 서 언

인간과 기계와의 의사전달을 위한 매개로 자연언어는 어떤 다른 매체보다 친숙하고 자연스러우며, 다양하고 유연한 표현을 가능하게 한다. 메뉴나 아이콘, 형식언어에 이르기까지 많은 인터페이스 방법들이 각기 장점을 가지고 사용되어지지만, 자연언어는 초보적인 사용자로부터 전문가에 이르기까지 누구나 특별한 학습과정이 없이 사용할 수 있는 대화 수단이 된다. 예를 들어 데이터베이스의 질의어로 유연하고 강력한 SQL은 영어와 유사한 구조를 가지고 있어 사용자가 습득하기 용이하다지만, 실제의 경우 상당한 학습기간이 필요하고 데이터베이스 이론이나 사용하고자하는 데이터베이스 스키마에 관한 세부적인 사항을 알고 있어야 하므로 그리 간단하지가 않다. 메뉴를 사용한다 하더라도 극히 제한된 부분만을 표현할 수 있는 난점이 있으며 보다 많은 그리고 다양한 기능을 수용하기 위해서는 다중으로 내포된 복잡한 메뉴 구조가 필연적이다. 이는 배우기 쉽다는 메뉴 방식의 장점을 약화시키며, 이런 방법을 사용해도 사용자가 원하는 다양하고 유연한 질의가 불가능한 것이 실상이다. GUI의 icon을 이용하는 것도 또한 시각적인 효과를 줄 뿐, 근본적으로 메뉴의 단점을 극복할 수 없다.

그러나 한국어와 같은 자연언어를 사용할 때는 이러한 문제들이 해결될 수 있다. 사용자는 일상 생활에 사용하는 언어를 통해 마치 옆사람에게 질문하듯이 시스템과 대화할 수 있으므로 별도의 시스템 인터페이스 방법을 학습할 필요가 없다. 또한 데이터베이스 스키마와 같이 복잡한 구조를 기억할 필요도 없다. 자연언어가 갖는 표현 능력은 기존의 어떤 다른 인터페이스 방식보다 유연하고 강력하며, 복잡한 질의를 비교적 간단한 문장으로 표현할 수 있어, 사용자 측면에서는 가장 편한 대화 방식이라고 할 수 있다.

그러나 컴퓨터상에서 시스템이 자연언어를 이해하도록 하는 것은 간단한 작업이 아니다. 표현 능력이 강력한 만큼 자연언어는 방대한 어휘와 복잡한 문법 구조, 다양한 표현 양식, 통상적으로 발생하는 형태소 구분 의미 담화 수준의 애매성을 가지고 있기 때문이다. 이러한 문제들은 자연언어 처리를 연구하는 모든 사람들을 끊임없이 괴롭히는 난제들로 현상의 컴퓨터 기술로 완전한 해결이 불가능한 부분이다. 그 어떤 현실의 또는 가상의 시스템 보다 복잡한 자연언어를 보다 명확히 이해하기 위해서, 이론적인 측면이나 구현상의 필요성으로 인해 부가되는 자료와 프로시저의 양이 방대하므로 웬만한 문장을 이해하는데 소요되는 시간 또한 크고, 따라서 실시간 처리가 요청되는 사용자 인터페이스가 되려면 보다 많은 연구가 필

요하다고 할 수 있다.

흔히 SF 영화에서 컴퓨터 또는 기타 시스템과 자연언어를 통해 대화하는 것을 볼 수 있다. 이것은 인간이 현재 원하는 사용자 인터페이스 방식을 미래라는 도구를 통해 가상적으로 실현한 것이다. 과거의 많은 공상적인 관측들이 현실에서 실현되었듯이, 자연언어처리에 대한 기술도 날로 진보하여 보다 유용한 수단으로 실현될 것을 기대해 본다.

II 자연언어처리 동향

현재 자연언어처리가 적용되는 연구 분야로는 자연언어 인터페이스, 기계번역, 스펠 및 문법 검사기, 텍스트 처리, 텍스트 생성, 음성 처리 등이 있다. 선진제국의 연구에 있어서 각기 그 비중이 틀리지만, 국내에서는 기계번역 시스템에 보다 많은 연구가 수행되고 있고, 자연언어 인터페이스등에는 좀더 많은 연구와 투자가 필요하다고 생각된다. 세계적으로 상용화의 가능성이나 필요량을 고려하면, 대체로 자연언어 인터페이스가 40%, 기계번역이 20%, 나머지 부분들이 가 10% 내외의 시장으로 예측되고 있다. 그러나 현재까지 기계번역, 인터페이스등의 자연언어 상품의 실질적인 판매는 거의 미미하고, 시장 자체가 형성되지 않은 것으로 평가될 수도 있다. 그러므로 자연언어처리는 아직도 연구의 측면이 강하다고 보는 것이 타당하겠다.

국내에서 활발한 연구가 수행되고 있는 기계번역은 두개 이상의 언어 간에 자동 번역을 하는 연구로서, 영어나 일본어, 한국어등이 그 대상이 된다. 많은 연구가 수행되고 있지만 광범위한 영역에서의 질이 좋은 번역이라는 것은 현시점에서 사실상 어려운 문제이다. 그러나 해외에서는 일부 시스템들이 상용화되기도 하여 그 가능성을 보이고 있다. 현재 국내에서는 KT등에서 전화를 통한 통역 시스템의 일부로서도 연구가 수행되고 있고, ETRI에서는 호텔 예약 시스템에 외국인을 위한 통역 기능을 부가하는 연구도 수행된 바 있다. 기타 대학과 연구 기관에서도 많은 연구 결과가 발표되고 있다.

자연언어 인터페이스는 데이터베이스나 전문가 시스템, 기타 컴퓨터 시스템의 사용자 인터페이스로서 자연언어를 적용하고자 하는 연구이다. 이러한 분야는 사용자와 컴퓨터가 온라인 상황에서 대화를 하는 것이므로 기계번역등과는 다소 차이가 있다. 즉 실시간에 근접한 처리 속도가 필요하며, 언어 해석이 완전

하지 않더라도 사용자와 컴퓨터는 상호 지속적인 대화 속에서 자신의 표현을 수정하거나 이해에 갈 수 있다. 따라서 처리 알고리즘은 발화가 끝난 후 언어처리를 수행하기 시작하는 오프라인 보다는 발화의 진행과 동시에 해석을 수행해 가며 발화 진행 정도에 따라 해석 결과를 수정해 가는 온라인 방법론이 보다 실용적이며, 온라인 해석을 위해서는 병렬처리등의 알고리즘이 효과적일 수 있다. 또한 대화가 계속 진행될 때, 이전 대화문의 해석 결과 및 진행 과정을 참조하여 새로운 발화문을 해석하는 것이 필요하므로 단회이해등에 관한 연구가 질실한 분야이다.

문법 검사기는 해외에서 많이 상용화 되고 있는 분야이다. 현재는 주로 구문적인 정보를 이용하여 문법을 검사하고, 어휘의 적절성 정도를 지적하여 주고 있다. 언어학적인 접근 방법이나 통계적인 방법을 사용하여 구현되고 있다.

III. 제3장 자연언어 해석 기법

자연언어처리에 관한 연구는 컴퓨터 공학의 측면이 강하지만 언어학이나 계산언어학, 인지과학등의 관련 학문과의 밀접한 관계 속에서 수행된다. 연구의 방향도 또한 어떤 방향에서 접근하는가에 따라 상당한 차이가 있을 수 있다. 이러한 부분은 Shieber[2]등에 의해 논의된 바 있다.

자연언어처리는 형태소 해석과 구문 해석, 의미 해석, 문맥 담화 해석의 여러 수준에서 자연언어를 해석하는 부분과 인터페이스 하고자 하는 시스템에 따라 해석 결과를 가공하는 부분, 해석의 역과정에 대응하는 자연언어 생성 부분으로 구분할 수 있다. 자연언어 인터페이스 시스템 측면에서 보면 언어를 해석하는 부분이 가장 어려운 난제로 지적될 수 있고 실제 이 부분에 많은 연구들이 집중되고 있다. 언어의 생성 부분은 시스템이 적용되는 영역에 따라 응답되는 문장의 종류나 패턴이 한정되므로 비교적 용이한 부분이라고 생각된다. 그림 2.1은 언어 해석의 수준과 이를 처리하는 자연언어 시스템의 방법론의 일례를 보인 것이다. 자연언어처리 기술을 소개하는데 있어서 언어학등에 기초한 문법 이론들은 매우 중요하다. 그리고 이러한 문법 이론들을 컴퓨터 상에 구현하는 기법들도 그림에 보인 이상으로 다양하다. 그러나 많은 이론과 방법들을 짧은 지면에 소개하는 것은 다소 무리하므로 상세한 내용은 본 고에서 제하였다.

형태소 해석은 언어가 가지는 유의미한 최소 단위

THEORY	Levels of Structure	Implementation Techniques
Template-matching Rule-based	Sound (Phonetics, Phonology)	Linear Predictive Coefficient Filter Bank Analysis Backpropagation
Morpheme-based Word-based	Word (Morphology)	Hash Tables
Generative Transfor- -mational Grammar Generalized Phrase Structure Grammar Lexical Functional Grammar	Utterance (Syntax)	CF Rules ATN Chart Unification Parallel Parsing
Model Theory Case Grammar Semantic Primitives	Meaning (Semantics)	Production Rules Conceptual Dependency Scripts
Memory-based Reasoning Speech Acts Discourse Grammar	Context (Pragmatics)	Frames Semantic Networks Logic

그림 2.1 자연언어 시스템의 해석 수준

인 형태소 별로 입력 문장을 분할하는 부분이다. 명사 동사등의 품사 구분과 어미환용 해석, 불규칙 환용등의 해석, 띄어쓰기 검사 및 정정, 한글 영어 알파벳, 일본문자, 특수문자등의 다양한 사중 처리, 미정의 단어 처리등이 기존에 마련된 사전 (dictionary) 시스템을 이용하여 수행된다. 이러한 형태소 해석은 한국어나 일본어, 영어등의 언어에 따라 서로 다른 과정을 거치는 것이 일반적이다.

형태소들이 결합하여 문장을 이루기 내기 위해서, 결합을 제약하는 구분적인 문법 규칙 (grammatical rules)들이 언어마다 존재한다. 형태소 해석 결과들로부터 문법적 지식에 따라 문장의 구분적 연결 과정을 분석하는 것이 문법 해석이다. 여기에는 주로 언어학적인 문법 이론들과 컴퓨터 공학 측면의 해석 알고리즘이 함께 적용된다. 문법이론은 규칙이나 프로시저의 형태로 시스템에 부가되며 이를 입력문에 적용하는 해석 알고리즘은 트리나 그래프, 표의 자료구조를 운용하는 방법론이 일반적이다. 문법 규칙의 정도에 따라 자연언어처리 시스템이 해석할 수 있는 언어의 범위가 한정되는 것이 일반적이고, 문법을 기술하는

방법론이나 해석 알고리즘에 따라 복잡도 (complexity)의 정도나 처리 효율이 결정된다고 할 수 있다. 문법 규칙을 적용해야 하는 요소들의 갯수에 따라 문법 해석 시간은 기하급수적으로 증가하는 것이 일반적이므로, 한국어나 일본어의 경우 수사나 접미사, 접두사, 환용 어미등의 형태소들을 미리 결합시키는 선처리 과정을 거쳐 규칙을 적용시키는 방법이 많이 사용되고 있다. 자연언어해석에 적용되는 문법이론의 종류는 상당히 많고, 또한 해석 알고리즘도 다양하다. 학문적인 측면에서는 보다 광범위한 언어 현상을 수용하는 이론이 연구되어 지고 있지만, 공학적인 측면에서 실용적인 시스템 구축을 위해서는 제한된 영역이 일반적이므로 이에 적당한 이론이나 효율적인 알고리즘을 개발하는 연구도 많이 수행되고 있다.

단순히 구분적인 오류없이 형태소들을 결합시킨다고 하여 언어에 적합한 문장은 아니다. 각 어휘가 가진 의미들이 결합하여 하나의 큰 의미적 관계로 적절히 조화되는가 하는 것을 검사하는 부분이 의미해석이다. 술어와 기타 어휘간의 의미적인 관계를 검토하여 결정하는 것이 대표적인 처리 과정이고, 보통은

Fillmore의 격문법 (case grammar) 이론이나 몬테규 문법 (Montague grammar), 상황의미론 (situational semantics) 등의 논리에 기반을 둔 형태가 많이 사용되어진다.

현상의 대부분 자연언어처리 시스템들은 문장단위의 처리에 기반을 두는 것이 보통이다. 그러나 자연언어 인터페이스 시스템등에서 사용자가 시스템과 대화하는 것은 각각의 문장 단위가 아니라 이전 대화내용과의 연장선상에서 이루어 지는 것이다. 이렇게 여러 문장간의 연관 관계를 규정하고 그 속에서 언어 이해가 수행되도록 하는 연구가 담화이해 또는 문맥 해석이다. 방법론적으로는 문장을 해석할 때마다 그 결과를 적절히 메모리에 쌓아두고 새로운 발화가 진행될 때 이전의 담화이력 메모리를 참조하여 해석하는 방법론이 대부분이다.

이상에서의 자연언어 해석 과정은 기존에는 순차적으로 수행하는 것이 일반적이었으나, 한단계에서의 해석 애매성을 다른 단계의 처리로 해결될 수 있는 경우가 많으므로 병행하여 수행하는 것이 현상이다.

해석된 결과를 시스템의 종류에 따라 적절한 적용 영역의 데이터로 바꾸는 부분이 필요하게 되며 이를 변환과정이라고 한다. 예를 들어 한영 번역 시스템의 경우라면 한국어 해석 결과를 역어의 기반 구조로 변환하는 것이 이에 해당하며, 데이터베이스의 인터페이스 시스템이라면 자연언어 해석 결과를 형식질의어 SQL 등으로 변환하는 부분이 될 것이다. 여기에서는 시스템의 적용 대상 영역에 따라 제한된 지식을 미리 표현하여 두고 이를 사용하여 변환하는 방법이 많이 사용되어 지고 있다.

자연언어를 해석하는 방법은 다양하다. 언어를 사용하는 문법이론을 개발하는 언어학 측면의 연구와 이러한 문법이론을 구현하기 위한 알고리즘과 형식론 (formalism)에 관심을 갖는 계산 언어학, 상식 (common sense)와 영역 지식 (domain knowledge)에 기반을 둔 인공지능 기반의 연구들로 구분될 수 있다.

문법 규칙을 이용하여 언어를 해석하는 형식론으로는 문맥 자유형 해석 (context-free parsing), 단일화 기반의 해석 (unification based parsing), ATN (Augmented transition networks), 차트 해석 (chart parsing), 병렬 해석 (parallel parsing) 기법들이 현재 가장 많이 사용되고 있다고 보여진다. 인공지능 기법에 의한 지식 표현 방식은 전통적으로 논리, 생성 규칙 (production rules), 의미 네트워크 (semantic networks), 개념 의존 구조 (conceptual dependency structure), 프레임 (frames)

에 기반한 방법들이 적용되어 지고 있다.

현재 국내에서 연구되는 자연언어를 컴퓨터 처리하는 방법은 단일화 문법 (unification grammar)에 기반한 연구, 격이론 (case theory) 또는 의존 문법 (dependency grammar)에 기반한 연구, 대규모 corpus를 구축하여 어휘간의 공기 빈도를 이용하는 통계 확률 기반의 방법, 용례 기반 (example based)의 방법, 세계 지식이나 담화 메모리를 이용하는 언어 이해에 기반한 연구들이 주로 연구되고 있다.

단일화 문법에 관한 국내 연구는 기존의 구구조 문법 (phrase structure grammar)을 확장하는 측면에서 수행되고 있다. 일반화 구구조 문법 (generalized phrase structure grammar), 어휘 기능 문법 (lexical functional grammar), 중심어 주도형 구구조 문법 (head-driven phrased structure grammar) 등의 언어 이론을 중심으로 이를 한국어에 적용하는 연구가 활발하다. 문법을 처리하는 컴퓨터 형식론으로는 차트 해석법과 PATR-II 방법이 많이 적용되고 있다.

의존 문법이나 격 문법은 국내 자연언어처리 연구에서 오랜동안 사용되어오는 방법이고, 한국어, 일본어와 같은 언어에 특히 유용한 방법론으로 평가되고 있다. 단일화 문법등의 연구에 있어서도 의미해석 기법으로 격이론을 적용하는 경우가 가능하다고 생각된다.

최근의 연구로 대규모의 corpus를 이용하여 언어 데이터를 수집 분석하고, 문해석에서의 구조적인 애매성을 해소하기 위해 분석된 확률 데이터로 단어간의 공기 빈도를 활용하는 연구도 다수 수행되고 있다. 이를 위해 corpus를 구축하는 작업과 이를 활용하는 방법의 연구가 활발히 진행되고 있다. 번역 시스템에 있어서는 corpus에 구축된 대역 문장을 용례로 하여 처리 대상 문장과의 의미적인 거리를 평가하고 최소 거리의 대역 용례를 참조하여 번역을 수행하는 용례 기반의 번역 (example based machine translation) 기법이 연구되기 시작하고 있다. 이 방법은 80년대말 일본에서 시작된 방법으로 고품질의 번역이 가능하고 용례의 추가에 의해 번역의 질을 향상시킬 수 있는 등의 이유로 관심을 모으고 있다.

담화이해에 관한 연구는 자연언어 인터페이스나 통역전화등을 위해 연구가 수행되고 있다. 대화의 진행에 따라 이미 처리된 데이터를 체계적으로 저장하고, 미리 작성된 실세계 상황 지식이나 대화 계획에 따라 담화의 예측과 유도를 가능하게 하며, 새로운 발화의 해석과 이해를 용이하게 한다. 자연언어 인터페

이스의 경우, 대화의 반복에 의해 사용자의 의도를 파악하여 시스템이 정보를 제공하는 것이 일반적이다. 시스템에 따라 사용자의 의도는 제한된 영역으로 한정되므로 대화의 사전 계획이 가능하고, 이 계획에 따라 대화를 유도할 수 있는 계획 인식 (plan recognition) 방법이 유용하다.

IV. 연구 경험

필자들은 자연언어처리 연구로서 데이터베이스의 한국어 인터페이스 시스템, 한국어 온라인 해석기, 호텔 예약 시스템, 자동통역을 위한 한일 일간간 기계번역 시스템등의 연구를 진행하여 왔다. 본 장에서는 데이터베이스 인터페이스와 호텔 예약 시스템의 개요를 설명하여 자연언어 인터페이스 기술의 일단을 보이고자 한다.

4.1 데이터베이스의 자연언어 인터페이스

본 절에서는 자연언어를 질의어로 사용하여 데이터베이스 내에 저장된 자료로부터 정보를 획득하는 지식인 정보 제공 시스템인 NAULI (NAUrUral Language Interface)[15.16]에 관하여 설명하고자 한다. NAULI는 도서및 논문 기타 문헌 관리 데이터베이스의 한국어 인터페이스 시스템으로, 사용자가 한국어 텍스트를 통해 원하는 정보에 대해 질의하면 이를 해석하여 데이터베이스를 검색하고 그 결과를 출력하는 시스템이다. 적용 영역은 문헌 관리이지만 이식성을 고려하여 설계하였으므로, 타 영역에도 비교적 용이하게 적용될 수 있다고 판단된다.

먼저 개략적인 구조를 보면, NAULI는 언어처리를 수행하는 언어 해석부와 대상영역 데이터베이스에 관한 처리를 수행하는 질의어 해석부로 구분된다. 시스템의 유용성과 실용적인 측면을 고려하여 언어 처리부를 대상 영역과 독립적으로 구현하는 방식으로 설계하였다. 그림 4.1은 시스템 처리 개요이다. 전처리 과정과 형태소 해석 (어간 추출), 어절내 해석, 어절간 해석을 행하여 논리식을 얻는 언어 해석부와 논리식으로부터 SQL 문을 얻는 질의어 변환부, 본 시스템을 DBMS에 연계시키는 인터페이스와 대상 데이터베이스로 이루어진 데이터베이스부로 구성된다. 언어 해석부는 한국어 질의문에 대하여, 형태소 해석과 구분·의미해석을 수행하는 부분이다. 이 언어해석 과정을 위하여 범용 사전, 고유명사 사전, 동사 어미 사전, 조사 사전 등을 구축하였다. 어절간 해석 과정은

단일화에 기반을 둔 한국어 해석 방식을 정의하여 적용하였으며, 이때의 해석결과는 구분해석목과 결합가 등의 표층격 정보에 기반을 둔 의미표현 형식이 되도록 하였다. 질의어 해석부는 언어해석 결과를 SQL 문으로 변환하는 부분이다. 언어 해석결과를 SQL 문으로 변환하는데 필요한 정보는 지식베이스 내에 기술하였으며, 자연언어 질의에 빈번하게 발생하는 생략문과 정보 결핍문도 처리할 수 있다.

4.1.1 언어 해석부

자연언어처리의 입장에서 볼 때, 한국어는 다른 언어와 구별되는 많은 특징을 가지고 있다. 이러한 한국어 문장을 처리하기 위해서 간단한 단문 수준의 문장 이외에 내포문, 중문, 혼합문 등 기타 제반 언어 현상을 효율적으로 처리할 수 있는 문법 이론이 다양하게 연구되어져 왔다. 그러나 현재 이러한 연구가 한국어가 가지는 다양한 언어 현상의 상당 부분을 해결하지 못하고 있는 실정이다. NAULI는 언어 이해의 측면에서, 데이터베이스 인터페이스에 입력 가능한 질의어를 대상으로 분석하고 이에 따른 언어 해석기를 구현하였다. NAULI 언어 해석부의 처리 단계는 전처리 과정과 형태소 해석, 어절내 해석, 어절간 해석으로 구성되어 있다. 각 모듈의 역할과 방법론을 간단히 설명한다.

(1) 전처리 및 형태소 해석

전처리 과정에서는 입력문에서 나타나는 각종 부호와 영어, 일본어, 한자 등의 외국어 성분을 한국어 성분과 구분한다. 형태소 해석과정에서는 어휘사전을 이용하여 각 어절 내의 어간과 어미 또는 어간과 조사 등을 구분하게 된다.

(2) 어절내 해석

형태소 해석과 어간의 추출 과정이 종료되면, 어절 내에서 어간 이외의 부분들이 가지고 있는 정보들을 속성화하여 어간 내에 포함시키는 과정이 어절내 해석 과정이다. 이 처리 과정에서는 조사 사전과 어미 사전을 참조하여, 어간의 품사에 따라 각각 조사 처리와 어미 처리로 구분하여 수행한다.

(3) 어절간 해석

어절내 해석 과정의 결과를 구성 성분으로 하여, 문장 전체의 구조를 파악하고 문장의 의미를 논리식으로 추출하는 것이 어절간 해석 과정이다. 또한, 해석

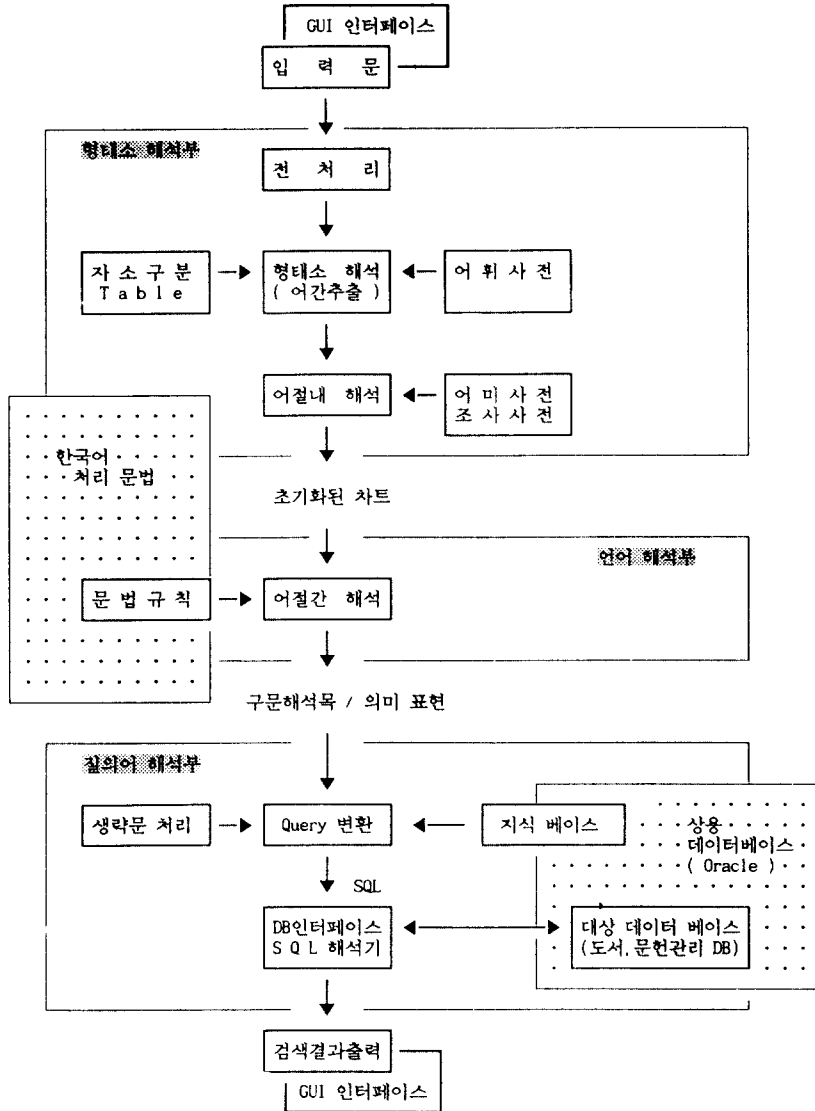


그림 4.1 NAULI의 시스템 구성도

과정의 애매성으로 인하여 발생하는 비결정성을 해결한다. 이를 위하여 NAULI는 단일화 문법에 기반을 두고 구축되었다.

그림 4.2는 NAULI가 사용하는 문법 규칙의 일례이다. PATR-II[3]와 같은 주석 (annotations) 부가 방식으로 구현하지만 선언적인 주석 (declarative annotations) 대신에 절차적인 주석 (procedural annotations)이 가능하도록 확장하고 있다. 이러한 처리 방식은 단일화

연산을 세분화하여 처리 효율을 높일 수 있다고 생각되며, 문법 규칙에서 이러한 연산들을 선택할 수 있어 규칙 기술의 유연성을 제공하고 비 단일화 연산이나 프로시저를 규칙내에 수용할 수 있는 장점이 있다.

NAULI에서 사용하는 문법 규칙은 3부분으로 나누어진다. 구구조 규칙부와 하위 절점 1, 하위 절점 2에 관한 속성 천이부이다. 구구조 규칙이 적용될 때, 다양한 함수를 사용하여 속성 구조의 검색과 천이, 구조

```

((VP -> V NP)
  ((fequal '($daul syn head sform) 'ap)
   (funify '($moth sem def) '($daul sem)
    ...))
  ((fsemk '($daul syn head subcat) '($dau2 syn head semk))
   (funify '($moth sem) '($dau2 sem)))
  (fvarcn '($moth sem (caar (tra_feat '($dau2 sem))))
   '($moth sem def (subcat_flag)))
  (fset '($moth syn head) '($dau2 syn head)
   ...))
...]
```

그림 4.2 언어해석 규칙의 일례

변환 등이 가능하고 문법 규칙을 유연하게 기술할 수 있도록 하기 위하여 절차적(procedural)인 문법 기술을 수용하였다. 이때 사용되는 함수는 단일화를 위한 funify, 속성값의 존재 여부를 판단하는 fequal·fnotexist, 속성값을 비교하는 fequal·fnotequal, 하위범주 속성 원칙을 통제하는 fsemk, 연결 속성을 생성하는 fplus, 기타 fset, tra-feat 등이 있다. 이러한 함수들의 인자들도 또한 함수 표현을 가질 수 있도록 하였다. 규칙에 기술된 \$moth, \$daul, \$dau2는 각각 지정된 구구조 규칙이 나타내는 부분 해석목의 상위 절집, 하위 절집 1, 하위 절집 2를 의미한다. 속성값은 하나를 갖는 것이 일반적이나 기술상의 문제나 처리의 효율을 위하여 복수의 값을 가질 수 있도록 하였고, 이것은 연결(conjunctive) 또는 이절(disjunctive) 속성구조를 통해 가능하다. 작성된 언어해석 규칙들은 차트해석[6](chart parsing) 기법을 이용하여 구현된 해석기를 통해 시스템에 적용하였다.

4.1.2 질의어 해석부

질의어 해석부는 일계 술어논리식(the first order predicate logic)을 중간표현으로 하여 변환 전처리 모듈과 질의어변환 모듈로 구분하여 처리하였다. 이때 원활한 변환을 위하여 데이터베이스 구조와 기타 영역 정보에 관한 실세계 모델로 지식베이스를 구축하였다. 이 지식베이스는 SQL 생성을 위한 기반 자료로 사용되어질 뿐만 아니라, 언어이해에 기반을 둔 의미 해석을 위해서도 필수적이라고 할 수 있다. 또한 데이터베이스의 front-end로 사용되는 NAULI에 입력되는 질의문들은 특정 구문이 생략된 문장이나 필요한 정보가 결핍된 질의어의 형태로 시스템에 입력되는 경우가 빈번하므로 이러한 생략유형을 분석하고 적절

한 처리를 할수 있도록 하였다. 기존의 연구들에서 생략문의 처리는 직전의 문장을 참고하여 해결하는 방식이 많지만, NAULI에서는 정보 검색문에서 발생하는 생략문들을 이전에 시스템에 제공되었던 질문문들과 연계하여 재분석함으로써 해결함은 물론, 기타 시스템 내부에서 해결이 불가능한 기타 정보 결핍문들의 유형을 분석하고 이를 사용자와 대화적(interactive)인 방법을 통하여 필요한 정보를 추출, 사용자의 의도를 파악하는 지적인 시스템이 되도록 하였다.

(1) 지식베이스

지식베이스는 데이터베이스의 구조와 저장된 데이터에 관한 지식, 각 어휘의 동의어등에 대한 지식을 포함하는 영역 의존적인(Domain-dependent) 정보들로 구축하였다. 따라서 변환 전처리 모듈, 질의어 변환 모듈, 생략문 처리 모듈등 질의어 해석부의 제 과정에 주된 정보원으로 참조된다. 지식베이스의 구조는 언어 해석 결과에 내포된 자연언어의 어휘들을 실제 데이터베이스의 관계(relation), 속성(attribute), 값(value)등과 각기 연계시키는 형식이며, 여기서는 이를 효율적으로 표현하기 위한 중간 단계로써 속성들의 대표값으로 DOMAIN을 설정하였다.

지식베이스의 간략한 구성 예를 테이블 형태로 보면 것이 그림 4.3이다. 이때 지식베이스는 Oracle 상에 구현하였고, 이것은 Oracle 상에 구축된 데이터베이스와 동일한 방법으로 접근할 수 있어 시스템 구현이 용이한 장점이 있다. 그림에서와 같이 각 관계와 속성은 각기 적절한 DOMAIN을 가지고 있으므로, 자연언어의 어휘들은 이러한 DOMAIN을 통하여 데이터베이스 구조와 연관되어 진다. 즉, 명사인 어휘항목은 속성에 해당하는 DOMAIN이 할당되고 관계, 속성,

값중 하나의 분류를 가진다. 이때 DOMAIN은 각기 적절한 관계나 속성과 연관되도록 한다. 또한 동사인 어휘항목들은 동사의 각 하위범주에 해당하는 데이터베이스 구조들을 지정하고 있다. 실제 자연언어 어휘들은 그 수가 상당히 많으므로, 지식베이스에 기술할 정보의 양을 최소화하기 위하여, 데이터베이스라는 축소된 영역에서 같은 의미 정보를 갖는 어휘들을 분류하여 동의어 테이블을 구성하였고, 따라서 지식베이스에 기술된 어휘들은 동의어 테이블에 의해 분류된 개념어 (대표어)들이 되도록 하였다. 동의어 테이블에서 #YEARS는 함수로서, '1980년대'와 같은 경우 1980년부터 10년간이라는 의미이므로 이를 처리하게 된다.

(2) 변환 전처리

변환 전처리 과정에서는 언어해석의 결과인 의미

표현을 일게 술어 논리식의 연결 정규형 (conjunctive normal form)으로 변환하는 역할을 한다. 이것은 언어 해석 결과와 형식 질의어 SQL 사이에 직접적인 연관 관계를 설정하기가 어렵기 때문에, 중간단계로 술어 논리식을 정의하여 두 표현 간의 변환을 용이하게 하기 위한 것이다. 또한 언어해석 결과중에는, '명사+명사'의 관계등과 같이 대상 영역 지식에 의해 재해석되어야 할 부분이 있기 때문에, 언어이해에 기반을 둔 front-end시스템으로써는 이러한 중간변환 단계가 필요하다고 할 수 있다.

변환 전처리 과정은 SEM 변환과 LOGIC 변환의 2 단계로 처리된다. SEM 변환은 언어해석 결과인 속성 구조를 논리식으로 변환하는 부분이다. 반복적으로 내재된 속성 구조를 평면적으로 전개하고, 이때 필요한 변수를 설정하며 각 변수에 대한 양화사 (Quantifier)를 부여한다. 데이터베이스 영역에서는 '책 1권', '3명

명 사	DOMAIN	CLASS
인공지능 개론	BOOK	VAL
Introduction to Artificial Intelligence	BOOK	VAL
문헌	BOOK	ATT
문헌	PAPER	ATT
長尾眞	MAN	VAL
최병욱	MAN	VAL
NGH	PUBLISHER	VAL
한국	NATION	VAL

* 동사

동 사	REL_ARGO	ATT_ARGO	REL_ARG1	ATT_ARG1
가지다	OWNER	NAME	OWNER	BOOK
출판하다	BOOKLIST	PCODE	BOOKLIST	BNAME
출판하다	BOOKLIST	PCODE	BOOKLIST	YEAR

* DOMAIN

DOMAIN	RELATION	ATTRIBUTE
MAN	AUTHOR	NAME
MAN	OWNER	NAME
PUBLISHER	PUBLISHER	NAME
PAPER	CONTENT	NAME
FIELD	FIELD	NAME

* 동의어

단 어	개 념 어
저술하다	쓰다
내다	쓰다
내다	출판하다
대한민국	한국
도서	책
년대	#YEARS

그림 4.3 지식베이스 정보의 일례

만을'과 같이 양화 관계가 확실하게 지정된 언어표현이 아니라면, 대부분의 경우 데이터베이스 내의 모든 정보를 검색해야 하므로 'ALL'의 양화 표현이 되도록 하는 것이 적절하다고 할 수 있다.

LOGIC 변환은 SEM 변환결과인 속성구조로부터, 논리식 구조로 변환하는 부분이다. 논리식의 구조는 술어 (predicate)와 항목 (term)으로 형성되므로, 각 속성에 대한 PRED와 ARG 값들을 병렬로 하여 (Pred Term1 Term2.)의 형식으로 부분 논리식을 얻을 수 있고, 부분 논리식을 연결하여 전체 논리식을 생성한다.

(3)질의어변환

논리식을 실제 DBMS Oracle의 형식 질의어인 SQL로 변환하는 부분이 질의어 변환 모듈이다. 질문/응답 시스템은, 사용되는 논리식에서 질문의 변항 값에 대응하는 답을 보이는 정리증명의 일종으로 고려할 수 있으므로, 논리질의어에 표현된 변항을 SQL의 SELECT절에, 상수항은 WHERE절에 각기 기술하며, FROM절에는 데이터베이스 스키마와 그림 4.3의 지식베이스로부터 JOIN될 릴레이션을 찾아 나열하는 것을 기본 알고리즘으로 하였다. 그러나 SQL 문을 생성하는 방법은 특정한 알고리즘만으로 규정하기가

어려운 문제이므로, 많은 경험적인 방법들을 부가하여 설계하였다. 예로서 주어등이 두개인 문장은 독립된 2개의 문장으로 구분하여 처리하며, 따라서 최종적으로 두개의 SQL 문이 생성된다. 즉 "Winston과 Gazdar가 쓴 책은?"과 같은 질의문이 "Winston이 쓴 책"과 "Gazdar가 쓴 책"을 의미하는지, 두 사람이 공저한 책을 의미하는지는 애매하며 일의적인 처리가 불가능하다. 많은 사례를 조사한 결과, 전자의 의미로 사용되는 경우가 더 많았고, 따라서 본 연구에서는 이 질의문을 두 문장으로 구분하여 처리하였다. 그러나 "같이 쓴", "공저한"과 같이 후자의 의미임이 명확할 경우는 두 문장으로 구분하지 않는다. 또한 내포문이 존재하는 경우는 내포문에 해당하는 SQL을 WHERE절에 삽입한다. 이러한 경험적 방법들은 입력 가능한 다양한 형태의 언어 (의미 구조) 형식에 따라 다수의 처리 방법이 필요하므로 demon의 형태로 부가하였다.

그림 4.4는 질의어 변환 과정의 기본 알고리즘이다. STEP 3.의 시스템 내부 표현 구조는 논리식을 프로그램 상에서 통제할 수 있도록 리스트 구조로 변경하는 것이며, 이후의 단계에서는 이 구조를 대상으로 처리된다.

- STEP 1. 의미표현을 입력
- STEP 2. 논리식으로 변환 (변환 전처리 과정).
- STEP 3. 시스템 내부 표현 구조 (리스트 구조)의 생성
- STEP 4. 시스템 내부 표현 구조 중 임시변수가 있으면,
 - 1) 임시변수의 값을 결정 변수 메모리 내에 등록
 - 2) 이 값을 REF에 할당
- STEP 5. 시스템 내부 표현 구조의 탐색
 - IF 의미 구조가 있으면,
 - THEN IF PRED이면
 - THEN STEP 6.로
 - ELSE IF ARG이면
 - THEN STEP 7.으로
 - ELSE REF등이면 SKIP
 - ELSE STEP 8.로
- STEP 6. 술어 처리 루틴 실행
 - 1) ARG 탐색
 - 2) 지식베이스 참조, 데이터 베이스 상에서 ARG들 사이의 최단 경로 결정
 - 3) WHERE 절에 추가
 - 4) STEP 5.로
- STEP 7. 항 처리 루틴 실행
 - : 지식베이스 참조, 상수형/변수형 명사 결정
 - 1) IF 변수형 명사이면,
 - THEN SELECT 절에 추가
 - ELSE IF 상수형 명사이면,

```

THEN WHERE 절에 추가
ELSE 함수이면,
    THEN 함수실행, WHERE 절에 부가
    ELSE SKIP
    
```

- 2) STEP 5.로
- STEP 8. FROM 절 형성 루틴
: 생성된 SELECT, WHERE 절로부터 FROM 절 생성
- STEP 9. 최적화 루틴
: 중복된 구조의 삭제
- STEP 10. SQL 질의어의 출력

그림 4.4 질의어 변환 과정

(4) 데이터베이스부

데이터베이스부는 구현된 자연언어 인터페이스 시스템을 상용의 DBMS Oracle에 접속하기 위하여는 인터페이스 모듈이다. 시스템의 출력 결과인 SQL을 이용하여 DBMS를 접근하는 이 모듈은 Embedded SQL을 허용하는 Pro*C precompiler를 이용하여 구현하였다. 또한 일반 사용자의 사용 환경 구축의 편의를 위한 도구로써 GUI 환경의 사전 및 지식베이스 편집기

와 사용자 입출력 편집기를 구현하였다.

4.1.3 시스템 구현 결과

NAULI는 SUN Sparc에서 동작하며, LISP과 C로 구현되어 있다. 동작 예는 그림 4.5, 4.6과 같고 실제로 중간 결과 출력 없이 GUI 환경에서 입출력부만 사용자에게 제공된다.

입력문 >> (知識の表現と利用은 어느 나라 사람이 썼는가?)

언어해석 결과 (의미부) ::

```

(SEM (PRED 쓰다)
      (TOP 知識の表現と利用)
      (SUBJ ((REF 사람)
              (GEN 나라 $V2)
              (DEF ((PRED 어느)
                    (ARGO $V2))))))
    
```

변환 전처리 결과1 (SEM 변환) ::

```

(SEM (BODY (PRED 쓰다)
            (TOP 知識の表現と利用)
            (SUBJ $V2))
      (QUANT ALL)
      (VAR $V2)
      (DESCR (PRED 나라)
              (ARG $V2)
              (DEF (PRED 어느)
                    (ARG $V2))
              (QUANT1 ALL)
              (VAR1 $V3)
              (DESCR1 (PRED 사람)
                      (ARG $V3))
              (DEF1 (PRED 출신국)
                    (ARGO $V3)
                    (ARG1 $V2)))
    
```

변환 전처리 결과2 (LOGIC 변환) ::

(ALL \$V2 ALL \$V3 (나라 \$V2) (어느 \$V2)
 (사람 \$V3) (출신국 \$V3 \$V2)
 (쓰다 \$V3 知識の表現と利用))

QUERY 변환 (SQL 생성) ::

```

SELECT AUTHOR, A.CODE, AUTHOR, NATION, BOOKLIST, B.CODE,
~      BOOKLIST, B.NAME
FROM   AUTHOR, AUTHORLIST, BOOKLIST
WHERE  BOOKLIST, B.NAME = '知識の表現と利用'
~      AND AUTHOR, A.CODE = AUTHORLIST, A.CODE
~      AND AUTHORLIST, B.CODE = BOOKLIST, B.CODE ;

```

출력문 >>

ACODE	NATION	BCODE	BNAME
10342	일본	20032	知識の表現と利用
10343	일본	20032	知識の表現と利用

processing time : 0.320 sec

그림 4.5 시스템의 출력 결과(I)

입력문 >> (WINSTON이 쓴 책 이름을 수정하라 \$)
 SYSTEM >> '책 이름'의 새로운 값은 무엇입니까?
 입력문 >> (ARTIFICIAL INTELLIGENCE \$)
 SYSTEM >> 'WINSTON이 쓴 책의 이름'을 모두 수정합니까?
 입력문 >> (아니오 \$)
 SYSTEM >> 'WINSTON이 쓴 책의 이름'의 원래 값은 무엇입니까?
 입력문 >> (ARTIFICIAL INTELLIGENT \$)
 출력문 >> 수정되었습니다 \$

BCODE	BNAME	ACODE	NAME
10017	ARTIFICIAL INTELLIGENCE	00150	WINSTON

그림 4.6 시스템의 출력 결과(II)

4.2 호텔 예약 시스템

호텔예약 시스템은 예약자가 전화문을 통해 호텔을 예약할 때, 호텔 담당자의 역할을 수행하는 시스템이다. 이 과정에는 음성의 인식과 합성을 포함하지만 여기서는 설명하지 않는다. 전화를 통해 입력되는 문장은 번역 시스템등에서 주로 적용되는 텍스트 문장이 아니라 일상 회화와 같은 담화문이다. 따라서 정형화된 문장이 아니라, 언어 표현의 임의성을 전제로 하고 비문법적 요소를 포함하므로, 자연언어 처리분야에서는 대표적인 난제로 지적되는 부분이 된다. 또한 화자의 의도나 대화의 배경이 되는 상식 등을 시스템

내에 형식화하여야 하고 담화의 전후 관계나 지시사 문제, 화제의 관리 및 일관성 유지등이 필요하다. 이러한 담화이해의 문제에 관하여는 기존의 여러 연구가 있었으나, 현상의 기술적 수준으로는 실용적인 또는 보편적인 시스템을 구축하는 것이 불가능한 실정이며, 따라서 담화의 예측 처리 및 불연속적인 담화상황의 변화등을 수용하는 이해 시스템을 목표로 한다면, 우선 적절한 소규모의 적용 영역을 대상으로 하는 것이 필연이라고 판단된다. 필자의 연구에서는 한국어 문장을 구분 및 의미 차원에서 해석하고, 이때 획득된 제 정보로부터 담화 정보를 추출하고, 술어를 중

심으로 한 각 명사구들의 의미 제약을 사용하여 화제를 유도 및 관리하는 설계방식을 정의하여 적용하였다. 이때 화제처리를 위하여 화제추출 테이블 및 화제 관리 지식 베이스를 정의한다.

4.2.1 호텔예약 시스템의 설계

시스템의 처리 개요는 그림 4.7과 같다. 시스템은 크게 언어 해석부, 문맥 처리부, 화제 처리부, 한국어 생성부의 4가지 부분 시스템으로 구성한다.

언어 해석부는 사용자에게 의해 입력된 한국어 문장은 전처리 과정을 거쳐 형태소 해석, 구문 해석을 거치며 의미표현식을 추출하는 과정으로 4.1절의 데이터베이스 인터페이스의 언어 처리부와 유사하므로 여기서는 설명하지 않는다. 단 연구의 진척에 따라 현재 언어해석부는 UNIX의 다중 처리 환경에서 ON-LINE

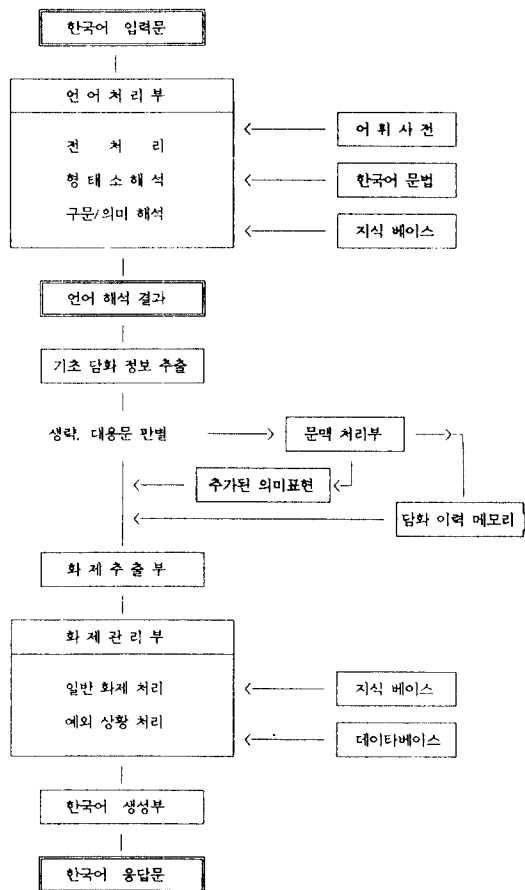


그림 4.7 호텔예약 시스템의 구성도

해석 방식으로 변환되었다.

언어 해석부의 처리 결과 추출된 속성 구조의 성분들은 문맥 처리부에서 재해석된다. 생략문, 대용현상과 같은 문맥적 현상을 고려하여 담화처리에 필요한 정보를 추출한다. 이 의미 구조의 정보와 화제 추출 TABLE을 이용하여 화제를 추출한다. 추출된 화제는 지식 베이스에 미리 설정되어 있는 지식으로부터 시스템이 예상하고 있는 화제와 비교되고 이전에 입력된 다른 화제들과 함께 일관성 있게 관리되며 호텔의 상황에 대한 데이터 베이스를 검색하여 적절한 응답을 사용자에게 제공한다.

4.2.2 문맥 처리부

담화문의 경우 문장의 일부를 대용어구로 대치하거나 불필요한 문장의 반복을 피하기 위해 문장 성분의 일부를 생략하는 경우가 빈번히 발생한다. 시스템에서는 언어 처리부에서의 의미 표현식에서 추출한 술어, 격, 의미속성의 정보로부터 담화문에서 빈번히 발생하는 문맥적 현상인 생략문 및 대용문을 처리한다.

생략문은 크게 문장간(intrasentential) 생략 및 문장내(intersentential) 생략으로 나눌 수 있다. 일반적인 담화문에서는 문장내의 경우보다는 문장간 생략이 주를 이루고 있다. 생략문의 판별은 언어 처리부의 의

```
user> 오늘밤 방을 예약하고 싶습니다. --- (1)
hotel> 오늘밤 빈방이 없습니다. --- (2)
user> 내일밤에도요? --- (3)
```

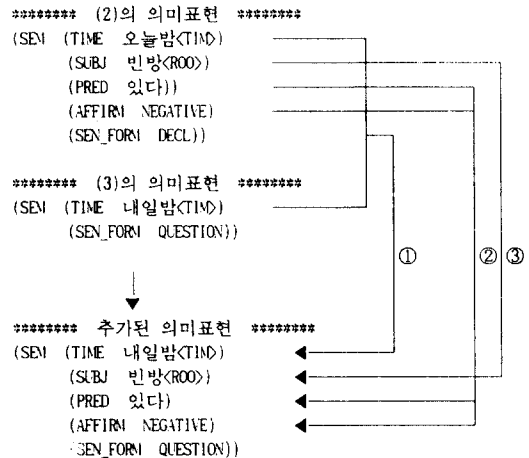


그림 4.8 간단한 생략문의 처리의 일례

미 표현에서 얻은 성분들을 이용한다. 우선 술어 성분의 존재를 확인한다. 술어성분이 존재할때 그 술어에 해당하는 결합가 패턴에서 필수격 성분의 유무를 조사하고 결핍된 필수격 성분들을 이전 담화문의 이력 메모리에서 추출한다. 술어성분이 존재하지 않을시에는 분석된 성분들의 의미속성을 비교하여 이전 문장에서 술어를 찾아 이전과 동일한 과정을 거친다. 그림 4.8은 간단한 생략문의 처리과정을 표시하고 있다.

대용문은 한 발화문내 또는 연속된 두 개의 발화문에서, 앞 문장의 명사와 동일한 명사가 다음 발화문에서 나타날 때 사용한다. 대용문의 처리는 대용어가 지시하는 이전 문장의 명사를 대치함으로써 처리하는 것이 일반적이며 대용문의 술어 성분의 결합가 (Valenz)와 대용어의 의미소성 (Distribution, Semantic marker)을 이용하여 처리한다. 그림 4.9은 대용문 처리의 일례이다.

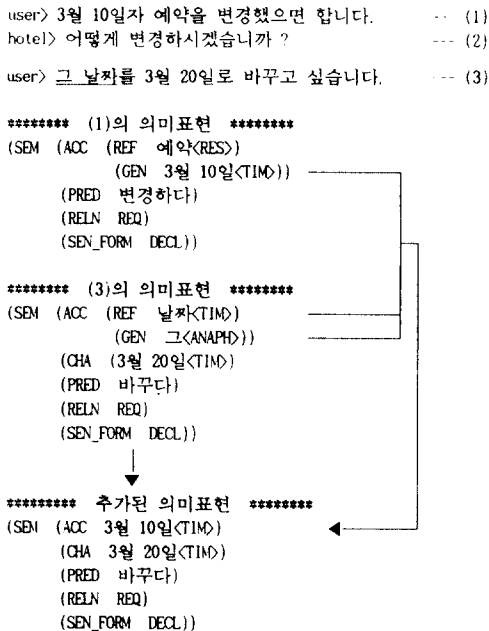


그림 4.9 간단한 대용문의 처리의 일례

4.2.3 화제 처리부

화제 처리부는 크게 화제 추출, 화제 관리부로 나누어진다. 문맥 처리부를 지친 최종적인 의미 표현의 정보를 이용하여 각 문장 단위의 화제를 추출한다. 언

어진 화제는 시스템이 예상하고 있는 화제와 비교되며 예외적인 화제로 천이되었을 때는 예외 상황으로 규정, 이에 적당한 응답을 사용자에게 제공하여 예외 상황을 해결하고 다시 원래의 화제로 복귀하게 된다. 예상된 화제라면 담화상황 관리부에서 이전에 입력된 다른 화제와 함께 관리되며, 호텔 예약 상황의 데이터베이스를 검색하여 적절한 응답을 사용자에게 제공한다.

호텔 상황에 관한 화제는 주화제와 그것에 종속되는 부화제로 구분한다. 주화제는 예약, 예약 문의, 예약 확인, 예약 취소 및 예약 변경으로 구성된다. 담화상황이 진행되면 우선 그림 4.10의 주화제 추출 Table의 정보를 참조하여 주화제를 추출한다. 얻어진 주화제에 종속하는 부화제들을 그림 4.11의 부화제 추출 Table을 이용함으로써 각 문장단위의 부화제들을 추출하게 된다. 이 화제 추출 Table은 술어성분을 중심으로 한 결합가와 화제간의 관계를 나타내고 있으며 호텔 예약에 관해 수집된 246문장의 발화문들을 기초로 작성하였다.

예상되는 모든 담화 상황은 필수 상황과 예외 상황으로 분리되어 그림 4.12와 같이 프레임 형식으로 지식베이스에 추가되도록 한다. 담화는 프레임에 지정된 선형 순서에 의거하여 화제를 예측 진행하되 예외 상황이 발생하면 순서를 변경한다. 담화의 진행은 프레임에 지정된 필수적인 예약상황을 모두 만족시키면 종료되며 필수 요소가 모두 자연스럽게 충족되도록 화제를 유도한다. 그림은 화제를 관리하기 위한 영역 의존적인 지식베이스의 일례이다.

화제	PRED	ACC	SUBJ	TIME	CONT	RELN	SEN_FORM
예약	예약하다	ROO/ROC		TIM		REQ	
예약	필요하다	ROO/ROC		TIM			
예약	되다		RES	TIM			
예약문의	얼마이다				MON	QLE	
예약문의	있다		ROO/ROC			QLE	
예약변경	변경하다	RES/ABS				REQ	

그림 4.10 주화제 추출 테이블

소화제	PRED	ACC	SUBJ	TIME_#	CONT	...	RELN	SEX_FORM	ORDER
객실 / 시간	예약하다	ROC/ROO		TIM			REQ		1/2
객실 / 시간	있다		ROC	TIM				QUESTION	1/2
예외 상황	있다		TBE					QUESTION	5.2
객실료	얼마이다		MON					QUESTION	4
.									.
예외 상황	포함되다		FOD			...			5.1

그림 4.11 부화제 추출 테이블(주화제: 예약)

	예 약	Feature	Value
1	객실 (N_ROOM)	TYPE	Single Room, Double Room, ...
		PERSON_NUM	1, 2, 3, ...
2	시간 (N_TIME)	STIM	6월 26일, ...
		ETIM	6월 29일, ...
		DURATION	4일, ...
3	이름 (N_NAME)	NAME	김철수
		TEL	290-0373
4	객실료 (N_FEE)	FEE	28만원, ...
		FEE_FOOD	nil, ...
5	예외 상황	5.1 FOOD	nil, include
		5.2 TMP_BED	nil, include
		5.3 CHECK_IN_TIM	오후 6시, ...
		5.4 DEPOSIT	nil, 3만원, ...
		5.5 WAITING_LIST	nil, include
		5.6 COMPANY_NAME	한양상사, ...
		5.7 ROOM_POSITION	북향, 남향, ...
E T C	.	.	.

그림 4.12 화제 관리 지식 베이스의 일례

4.2.4 한국어 생성부

자연스러운 대화가 가능한 질의응답 시스템을 구축하기 위해선 해석된 입력문의 의도에 맞게 적절한 문장으로 생성할 수 있어야 한다. 완전한 문장을 생성

하기 위해서는 한국어의 문장 구조에 관한 연구가 우선적으로 필요하며, 타당한 어휘의 선정, 표현 양태의 결정등이 필수적이다. 또한 문장들중에서도 더욱 자연스러운 문장을 결정하기 위해서는 화용론적인 연

구도 필요하다. 그러나, 이와 같은 언어 생성은 또한 심층적인 연구가 필요하고, 보통 제한된 영역에서의 질의 응답 시스템에서는 생성하는 문장이 한정되는 것이 일반적이므로 간단한 생성부로 대신하였다.

호텔 예약과 같이 지정된 대상영역에서는 영역에 한정된 부분 시스템으로 패턴정합에 근거한 생성부

를 구축하는 것이 효율적이라 판단된다. 따라서 시스템에서는 질의응답 도중 얻어진 주화제 및 부화제 변수들과 생성문장을 연관시키는 방법을 이용한다. 그림 4.13은 문장 생성을 위한 패턴의 예이고, 그림 4.14는 이러한 문장 생성을 통한 대화의 일례이다.

#Sent_pattern(17)

주화제 : 예약 부화제 : 객실료

"[N_NAME.NAME]씨 [N_TIME.DURATION]동안의 객실료는 [N_FEE.FEE]입니다. 예약하시겠습니까?"

그림 4.13 한국어 생성의 일례

```

hotel> 한양호텔 예약부입니다.
user> 5월 20일부터 23일까지 Single Room을 예약하고 싶습니다.
hotel> 예, 빈방이 남아 있습니다. 성함하고 연락처를 말씀해 주시겠습니까?
user> 김철수라고 합니다. 그리고 저의 전화번호는 290에 0373입니다.
hotel> 김철수씨, 4일동안의 숙박료는 28만원입니다. 예약하시겠습니까?
user> 예, 예약해 주세요.
hotel> 김철수씨 5월 20일부터 5월 23일까지 Single Room로 015호실로 예약이 되었습니다.
      맞습니까?
user> 예, 맞습니다. 감사합니다.

```

:: 대화 도중 얻어진 필수정보

- 1) N_ROOM = Single Room
- 2) N_TIME STIM = 5월 20일
 ETIM = 5월 23일
 DTIM = 4
- 3) N_NAME.NAME = 김철수
 N_NAME.TEL = 290-0313
- 4) N_FEE = N_FEE.SROO * N_TIME.DTIM = 28만원

그림 4.14 전형적인 담화 진행의 일례

IV. 결 언

컴퓨터를 이용한 기계와 인간과의 대화 과정에 있어서 가장 용이하고도 유연한 매체는 자연언어라는 관점에서 자연언어 처리 및 인터페이스 기술의 일단을 기술하였다. 어떠한 별도의 학습 과정이나 제한이 없이 주위 사람과 대화하는 것과 동일한 방법으로 컴퓨터나 기타 시스템과 정보를 주고 받는다는 측면에

서 보면, 자연언어 인터페이스는 사람에게 가장 친숙한 방법임이 틀림없다. 일반적인 시스템에서의 절충(trade-off) 관계와 마찬가지로, 인간에 친숙하려면 그만큼 시스템이 갖는 부담은 크게 된다. 즉 자연언어 시스템은 영역내 지식에 해당하는 방대한 지식베이스와 복잡한 정보처리 과정을 필요로 하는 것이다. 현재 자연언어처리 기술에서는 극히 제한된 부분에서의 인터페이스만이 가능하고, 특히 현시점에서의 상

업성이 모자라는 것이 사실이지만, 언어학적인 그리고 컴퓨터 공학적인 많은 연구들이 세계적으로 집중되고 있는 현실을 감안할 때, 실용화의 가능성은 충분하다. 좀더 대상 영역에 관한 정보들을 체계화하고, 언어 정보를 적절히 제약하는 등의 방법을 통하여 어느 정도는 현실에서도 시스템 구현이 가능하다. 언어 처리에서 항상 제기되는 방대한 양의 애매성 처리 문제들이나 실시간 언어 해석에 관련하여 좀더 다양한 방법론들이 제시될 것으로 기대한다.

참 고 문 헌

1. S. M. Shieber : "Semantic-Head-Driven Generation Algorithm for Unification-Based Formalisms", *Proc of the 27th Annual Meeting of the Association for Computational Linguistics*, 1989
2. S. M. Shieber : "Separating Linguistic Analyses from Linguistic Theories", in *Natural Language Parsing and Linguistic Theories*, D. Reidel Pub., 1988
3. S. M. Shieber : *An Introduction to Unification Based Approaches to Grammar*, CSLI, 1986
4. A. Ramsay : "Disjunction without tears", *Computational Linguistics* Vol. 16 No. 3 September, 1990
5. R. Kasper, W. Rounds : "A Logical Semantics for Feature Structures", *Proc. of the 24th Annual Meeting of the Association for Computational Linguistics*, 1986
6. G. Gazdar, C. Mellish : *Natural Language Processing In LISP*, Addison-Wesley, 1989
7. G. Gazdar, E. Klein, G. K. Pullum, I. A. Sag : *Generalized Phrase Structure Grammar*, Blackwell, 1985
8. S. Carberry : "A Pragmatics-Based Approach To Ellipsis Resolution", *Computational Linguistics*, 1989
9. Steven L. Lytinen : "Semantics-first Natural Language Processing", *AAAI '91*, Vol. 1, 1991
10. R. E. Frederking : *Integrated Natural Language Dialogue-A Computational Model*, Kluwer Academic Pub., 1988
11. K. K. Obermeier : *Natural Language Processing Technologies in Artificial Intelligence*, Ellis Horwood Limited
12. 長尾眞 : *畫像と言語の認識工學*, コロナ社, 1989
13. 宇津呂 武仁, 松本 裕治, 長尾眞 : "第三者の観点による對話の理解に関する考察", *情報處理學會 自然言語處理研究會研究報告 83-17*, 1991
14. 菊井玄一郎, 鈴木雅實 : "表層表現の型を用いて對話文の生成について", *日本情報處理學會 自然言語處理研究會 研究報告 81-11*, 1991
15. 우요섭, 최병욱 : "데이터베이스를 위한 자연언어 인터페이스 NAULI의 설계 및 구현 (I)-언어해석 과정을 중심으로-", *대한전자공학회 논문지* Vol. 28-B No. 4, 1991
16. 우요섭, 최병욱 : "데이터베이스를 위한 자연언어 인터페이스 NAULI의 설계 및 구현 (II)-질의어 변환 과정을 중심으로-", *대한전자공학회 논문지* Vol. 28-B No. 4, 1991

우 요 섭

- 1986년 : 한양대학교 공과대학 전자통신공학과 (공학사)
- 1988년 : 한양대학교 대학원 전자통신공학과 (공학석사)
- 1992년 : 한양대학교 대학원 전자통신공학과 (공학박사)
- 1992년~현재 : 시립인천대학교 정보통신공학과 조교수

※ 관심분야 : 인공지능, 자연언어이해, 사용자 인터페이스 등

최 병 욱

※ 관심분야 : 인공지능, 자연언어이해, 화상이해, 신경회로망, 멀티미디어 등