자연언어 처리 기술

金 榮 澤
(正 會 員)
서울大學校 工科大學 電子計算機工學科 教授

Ⅰ. 자연언어 처리 개요

언어는 인간이 일상적으로 사용하는 자연언어와 인위적으로 구성된 인공언어로 크게 구분된다. 인간의 사고를 반영하는 자연언어는 그 복잡성과 언어외적인 지식을 필요로 한다는 점에서 인공언어와는 큰 차이를 이루고 있다. 이러한 자연언어를 모델화하여 기계적인 시스템, 즉 컴퓨터에 구현하고자 하는 학문이 '자연언어 처리(natural language processing)'이다. 이러한 자연언어 처리의 중대성은 인간의 지적능력을 모델화하여 지적 시스템(intelligent system)을 구축하고자 하는 인공지능(artificial intelligence)의 연구가 활발해짐에 따라 크게 부각되었다. 인공지능의 궁극적인 목표가 인간의 사고과정을 구현함에 있고, 인간의 사고, 학습, 의사전달의 기본수단이 자연언어라는 점을 고려하면, 컴퓨터에 의한 자연언어 처리는 필수적이라 하겠다.

자연언어 처리는 컴퓨터 과학뿐 아니라, 언어학, 심리학을 포함하는 포괄적인 학문으로서, 이들 관련 학문의 발전 및 실용적인 요구와 더불어 큰 진전을 이루어 왔다. 먼저 컴퓨터 과학적인 입장에서 볼 때, 하드웨어의 비약적인 발전은 자연언어의 복잡도를 컴퓨터가 수용할 수 있는 능력을 제공해 주고, lisp, prolog와 같은 프로그램 언어의 출현과, 관련된 소프트웨어 도구(tool)의 발전이 자연언어 처리를 용이하게 해주고 있다. 또한 Chomsky의 생성문법 이론 및 그 후 개발된 격문법(case grammar), GPSG(generalized phrase structure grammar), ATN(augmented transition network)문법등 언어 이론과 파싱(parsing)기술의 진보, 그리고 의미 네트웨크(semantic network)등 의미표현과 추론 메카니즘의 개발로 컴퓨터의 자연언어 처리능력을 크게 향상시키고 있다. 자연언어 처리의 발전은 실용적인 측면에 있어서도 컴퓨터의 대중화라는 큰 흐름에 의해, 보다 많은 사용자들에게 보다 쉽게 접근하도록 하기 위해서는 거의 절대적인 요구가 되고 있다.

기계번역(machine translation)과 자연언어 이해(natural language understanding)는 자연언어 처리에 있어서 가장 큰 두 분야로서 다음 두장에서 상세히 취급된다. 자연언어 처리는 음성인식 및 합성기술과 더불어 그 응용이 다양화하고 있으며 다음 표 1은 그 대표적인 예를 보여 준다.

표 1. 자연언어 처리의 대표적인 응용분야

응 용 분 야	기 능	비 고
기 계 번 역	원시 문안을 목표언어로 번역	1960년대부터 실용화
데이타 베이스 인터페이스	DBMS의 전위 시스템으로 자연언어 질의어를 처리	1981년부터 상업화
대 화 인터페이스	전문가 시스템과 같은 복잡한 시스템의 대화를 처리	시제품 출현
문안 편집	문안의 문체, 문법을 검사 및 개선	개발 시험중
발화 기록기 (talk writer)	말을 글로 변환하여 기록	하드웨어 개발 시험중

Ⅱ. 기계번역

1. 기계번역의 범주

기계를 사용한 번역시스템은 인간과 기계의 보조관계에 따라 MT(machine translation), MAT(machine-aided translation), 그리고 TD뱅크(terminology data banks)의 세 범주로 크게 구분된다.

MT는 컴퓨터가 인간의 개입없이 독자적으로 번역 과정을 수행하고 그에 대한 책임을 담당하는 시스템으로 가장 진보된 시스템이다. 물론 MT도 인간의 번역과 마찬가지로 전위 및 후위처리(pre-or post-processing)를 허용하는데-문안의 형식보존등을 위해-다만 구문이나 의미에 제한을 가하는 전위 처리는 생각하지 않는다.

MAT에는 컴퓨터가 전체적인 책임을 맡고 있으나, 번역도중 문장의 모호성(ambiguity) 처리등을 위해 인간의 개입을 허용하는 HAMT(human-assisted machine translation)와 인간이 전체적인 책임을 담당하고 번역에 필요한 자료-문법적 지식, 사용 예등-를 온라인(on-line)으로 참고하는 MAHT(machine-assisted human translation)가 있다.

TD뱅크는 MAHT보다 하위 수준의 시스템으로 주로 오프라인(off-line)으로 특정분야의 번역에 필요한 사전(事前) 정보를 번역자에게 제공해 준다. TD뱅크는 계속적으로 새로운 용어가 쏟아져 나오는 기술분야의 용어정립과 번역에 도움이 되고 있다.

좁은 의미의 기계번역은 MT만을 의미하지만 일반적으로 MT와 HAMT를 모두 포함하므로 본고에서도 MT와 HAMT를 가리켜 기계번역이라 한다.

2. 기계번역의 발전과정

1930년대부터 시작된 기계번역에 대한 연구는 컴퓨터가 등장한 1950년대까지는 실제로 개발된 시스템이 없이 다만 가능성만 제안되었을 뿐이다.

1954년 미국의 Georgetown대학에서 개발된 러시아-영어 번역시스템을 선두로 실용적인 제1세대 기계번역시스템이 개발되기 시작했다. 이들 1세대 시스템은 문안을 독립된 문장의 집합으로 간주하고 약간의 구문분석만 수행한 경험기초적인 시스템이었다.

이후 1960년대 중반까지 기계번역은 많이 활성화 되었으나, 1966년 미국의 ALPAC 보고서가 실용적인 기계번역시스템에 대해 비관적인 결론을 내림으로써, 그때까지의 연구마저도 거의 중단되는 침체기를 맞기 시작했다.

1970년대에 들어서면서 언어학의 발전과 더불어, 현재 기계번역시스템의 대표적 구조인 트랜스퍼(transfer) 방식이 정립되면서 많은 실험적인 시스템이 개발되었다. 이를 제2세대 기계번역시스템이라 하며, 주로 구문분석에 의존하고 lisp와 같은 고급언어의 사용이 특징적이다.

이후 1980년대에 들어서면서 지식베이스(knowledge base) 등 인공지능분야에서 개발된 이론과, 의미분석에 중점을 둔 제3세대 기계번역시스템의 연구가 활발해지고 있다. 대표적인 것으로 유럽공동체 국가간의 다언어(多言語) 번역시스템인 EUROTRA 프로젝트가 현재 진행중이다.

3. 기계번역의 목적과 전망

기계번역은 그 용도에 따라 정보수집(gathering)과 정보유포(dissemination)의 두 가지로 구분할 수 있다. 정보수집은 제한된 시간내에 방대한 양의 자료로부터 필요한 정보를 추출해 내는 것으로 신속성이 주가 된다. 정보유포는 기술수출(technology export)과 같이 구매자의 요구언어로 상품정보를 제공해 주는 것으로 정확성이 주가되고 있다. 정보수집은 빨리, 대충 번역한다는 점에서 과거 기계번역의 주요 대상이 되어 왔으나, 기계번역의 정확성 향상과, 무역의 신장 및 전문 번역가의 부족으로 정보유포가 최근 새로운 대상으로 부각되고 있다.

기술분야의 번역은 그 양의 방대함에 비해 신출 용어의 급증, 그리고 전문 번역가의 부족으로 인해 기계번역의 필요성을 한층 더 해주고 있으며, 정보화 사회에 있어서 다른 언어간의 통신에서 언어장벽을 해소하기 위해 기계번역은 더욱 활성화 될 것으로 전망된다.

4. 언어학적 기술에 따른 구분

지금까지 개발되었거나 개발중인 기계번역 시스템이 사용한 언어학적 기술은 크게 3가지 기본 특성에 의해 구분된다. 즉 직접(direct)과 간접(indirect), 피봇(pivot)과 트랜스퍼(transfer), 그리고 지역(local)과 전역(global) 방식이다. 여기서 주의해야할 것은 현재 모든 시스템이 정도의 차이는 있으나 의미분석을 수행하고 있으므로 의미분석은 더이상 분류특성이 되지 못한다는 점이다. 다음에 이들 각 특성에 대해 살펴보기로 한다.

직접방식은 처음부터 특정언어를 목표로 하여 이 목표 언어의 번역에 필요한 최소한의 분석만 수행하는 방식이다. 이 방식은 한일과 같이 같은 어족간의 번역에 적합하며 GAT를 예로 들 수 있다. 간접방식은 원시언어의 분석이 목표언어와는 무관하게 수행되는 것으로 한영과 같이 서로 다른 어족간에 적합하며 EUROTRA를 예로 들 수 있다.

피봇방식-interlingua 방식이라고도 함-은 개별언어와는 무관한 보편의미-이를 피봇이라 함-로 원시문장을 변환 후 이 피봇으로부터 목표문장을 생성하는 방식이다. 이 피봇방식은 언어 공통의 의미 표현을 가정하는

야심적인 방식이지만 CETA가 시도했으나 성공하지 못했고 가능성 여부도 아직 불투명한 상황이다. 트랜스퍼 방식은 원시문장을 분석하여(analysis) 중간표현으로 바꾼 후 이로부터 목표언어의 중간표현으로 변환(transfer)후 이 중간표현으로부터 목표문장을 생성(synthesis)하는 방식으로 현재 거의 대부분의 시스템이 이 방식을 택하고 있다. 직접방식은 위의 두-피봇과 트랜스퍼-방식중 어디에도 속하지 않으며 그림 1 은 이들간의 관계를 잘 나타내고 있다.

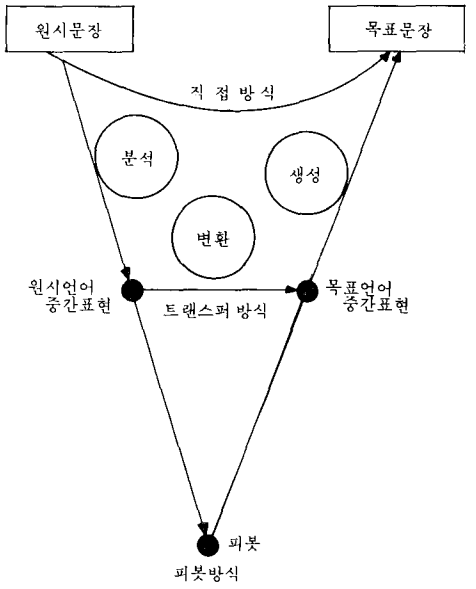


그림 1. 기계 번역의 방식

지역방식과 전역방식은 정도의 문제로 번역시 처리의 범위, 즉 윈도우(window)를 어느정도로 잡느냐의 문제와 관계된다. 지역방식은 단어를 처리의 기본단위로 하고 필요에 따라 전후를 참조하는 방식으로 SYSTRAN을 예로 들 수 있다. 전역방식은 일반적으로 문장단위 이상의 문맥내에서 단어의 의미를 살펴보고, 처리하는 방식으로 번역이 좀더 정확해질 수 있으며 METAL을 예로 들 수 있다.

5. 기계번역의 수행과정

여기서는 현재 서울대 자연언어 연구실에서 IBM사와 공동으로 개발중인 영한 기계번역시스템인 KSHALT(korean system for human-assisted language translation)를 중심으로 구체적으로 번역이 어떻게 수행되는가를 살펴보기로 한다. KSHALT는 전체적으로 그림 2 와 같은 구조를 갖고 있으며 IBM 매뉴얼을 대상으로 하고 있다.

KSHALT는 기본적으로 트랜스퍼방식을 취하고 있으며 영어 파서와 영문 나무구조 분석이 분석단계에 영한 변환이 변환단계에 한국어 생성이 생성단계에 각각 해당한다.

KSHALT의 특징은 분석단계를 목표언어와는 독립적으로 수행되는 영어 파서단계와 목표언어에 밀접한 영문 나무구조 분석의 두단계로 나눔으로써 새로운 목표언어로의 적용시에 융통성을 제공할 수 있는 모듈 기능을 강화한 점이다. 그럼 각 단계의 처리내용을 예와 함께 구체적으로 살펴보자.

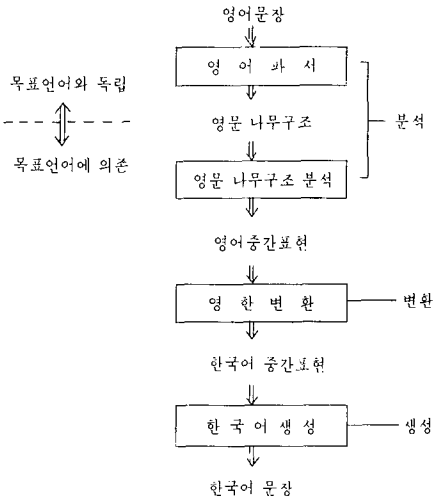


그림 2. KSHALT의 구조

(1) 영어 파서단계

이 단계에서는 원시언어인 영어 문장의 구문분석을 수행하여 구문론적으로 가능한 그 문장의 모든 나무구조를 가중치(weight value)와 함께 출력한다. 그림 3 은 리스트 구조로 표현된 출력을 도식화한 것이다. 물론 실제 출력시에는 수나 시제등 기타 정보들도 포함되지만 여기서는 보이지 않았다.

(2) 영문 나무구조 분석

이 단계에서는 파서에서 생성된 나무구조를 목표언어로의 번역에 적합한 형태로 변환하여 영어 중간표현을 출력한다. 이때 수행되는 주요기능은 가능한 나무구조가 2 개이상인 경우 가장 타당한 1 개를 선택하는 나무구조 선택과 그림 4 와 같이 목표언어의 특성에

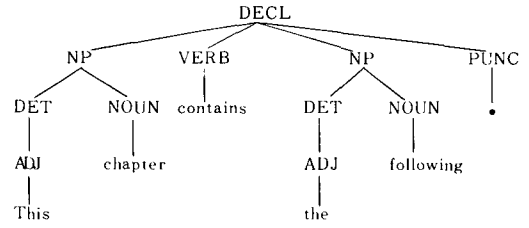
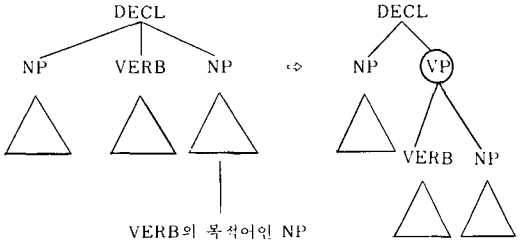


그림 3. 영어 파서의 출력예



(a) 파서의 출력 (b) 그룹화된 결과

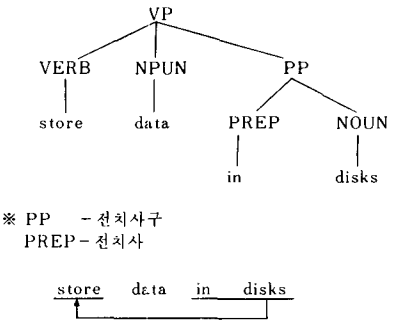
그림 4. 그룹화

맞도록 몇개의 노드(node)를 묶어 주는 그룹화(grouping) 및 그림 5 와 같은 전치사의 수식관계 결정, 그리고 접속사의 연결범위 결정등이다. 이런 기능을 수행하기 위해서는 목표언어의 특성을 이용함과 동시에 부분적인 의미분석이 함께 수행된다. 특히 전치사의 수식관계와 접속사의 연결범위 결정, 그리고 뒤에서 설명될 어휘치환등을 위해 단어를 의미에 따라 분류한 의미마커(semantic marker)가 사용된다.

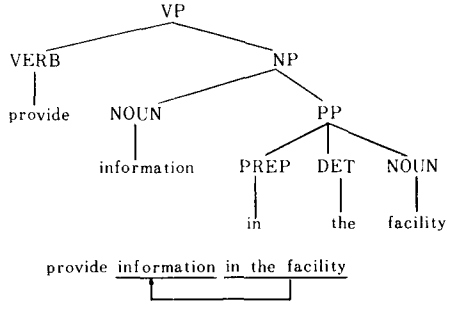
(3) 영한변환

이 단계에서는 먼저 두 언어간의 구조적 차이를 줄이기 위해 원시언어내에서 구조변환을 수행한다. 다음은 KSHALT에서 가주어를 갖는 문장의 처리예이며 이밖에도 so～that 구문처리등이 포함된다.

다음으로 이 단계의 주된 작업인 어휘치환이 수행된다. 이 부분에서는 원시언어대 목표언어의 치환 단어수가 일대일이 아니라는 점과 원시언어의 한 단어 또는 구에 대한 목표언어의 치환 경우가 여러가지인 다의성(多意性)이 문제가 된다. 단어수가 일대다인 경우는 별문제가 없고 다대일이나 다대다는 숙어와 유사한 방법으로 처리된다. 그리고 다의성이 있는 경우는 단



(a) PP가 VERB를 수식하는 경우



(b) PP가 NOUN을 수식하는 경우

그림 5. 전치사의 수식관계와 나무구조

It is important for the user to specify the file.

⇨That the user specify the file is important.

그림 6. 영한 변환단계에서의 구조변환 예

어의 의미마커와 문장의 나무구조를 바탕으로 어느 정도의 윈도우내에서 하나를 선택하게 된다. 그림7은 다의성의 예를 보여주고, 그림 8 은 이 단계까지 거친 결과를 보여주고 있다.

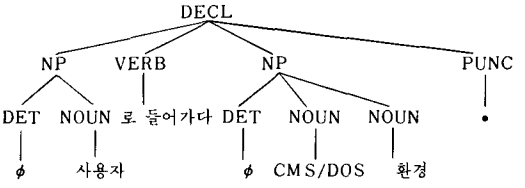
Computing system→ 컴퓨터 시스템

Computing power → 계산 능력

Knowledge of the system→ 시스템에 대한 지식

Computing power of the system→ 시스템의 계산능력

그림 7. 다의성의 예



A user enters the CMS/DOS environment.

※ φ는 공백을 의미함

그림 8. 영한 변환단계의 결과 예

(4) 한국어 생성

이 단계에서는 먼저 목표언어 어순에 맞게 나무구조를 변경하고, 목표언어의 특성에 따라 삭제 및 첨가가 수행된다. 목표언어가 한국어인 KSHALT의 경우는 반복되는 뒷 주어는 삭제하며, 긴 복합명사에서의 관형격 조사의 첨가 및 수량명사의 첨가등을 수행한다. 그림 9 는 이들처리가 끝나 한국어 어순이 된 결과를 보여 준다.

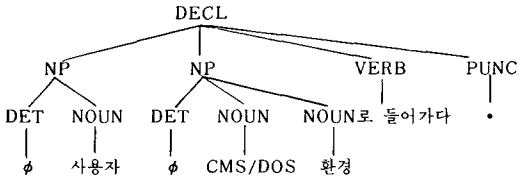


그림 9. 그림 8 의 어순 변경 후의 결과

다음에 KSHALT에서는 한국어 특성에 따라 시상과 음운론적 연결등을 고려하여 자연스러운 한국어 문장이 되도록 하기 위해, 조사, 어미, 보조어간등을 첨가하는 형태소 생성을 수행한다. 이때 매개모음, 모음조화, 부정형, 피동 및 사동형, 그리고 불규칙 용언등에 대한 처리가 주가 되며 띄어쓰기도 함께 고려되어 최종적인 결과가 생성된다. 그림10이 최종결과 예를 보여주고 있으며, 이는 필요에 따라 후위 처리의 입력이 되기도 한다.

이상에서 KSHALT를 중심으로 한 번역과정의 구체적인 예를 살펴보았는데, 언급되지 않은 내용중에서 사전(lexicon)이 각 단계에서 중요한 역할을 하며, 이들 각 단계에 필요한 사전구성이 쉽지 않은 작업이라는것만 상기시키고 본 절을 마치기로 한다.

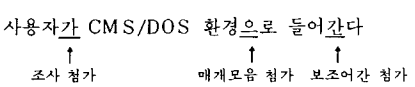


그림10. 그림 9 의 형태소 생성을 마친 최종결과

6. 한국어 처리의 특성

기계번역에서는 일반 언어 공통의 문제점 뿐 아니라, 원시언어와 목표언어의 특수성에서 오는 문제도 심각하다. 영한 번역의 경우 대표적인 것이 분사구문이나 부정사구의 용법을 명시적으로 파악해야 한다는 점이며, 한영번역의 경우는 한국어의 생략된 주어 및 수를 파악하여 영어 문장 생성시에 삽입해야 하는데 어느경우나 상당히 어려운 문제점을 내포하고 있다.

한국어 처리의 또다른 특성은 영어등의 굴절어와는 달리 첨가어인 관계로 반드시 형태소 분석이 필요하다는 점이다. 그림11은 형태소 분석의 한 예를 보여 주고 있는데, 한국어의 형태소는 일본어의 음절단위보다도 세분된 음운단위까지 분할되고 또한 모호성도 갖고 있어서 쉽게 처리되지는 않는다. 이 형태소 분류는 한국어 처리의 기본적인 작업으로써 현재 많은 연구가 진행되고 있다.

사람이라 부른다 = 사람(명사) + 이(서술격조사) + 라(인용조사)
부르(동사) + ㄴ(시제보조어간) + 다.(종결어미)

감기는 = 감기(명사) + 는(조사)
감(동사) + 기(피동보조어간) + 는(진행형 어미)

그림11. 형태소 분류와 모호성의 예

Ⅲ. 자연언어 이해

1. 자연언어 이해시스템

자연언어가 각종 응용시스템에서 비중을 더해가면서 지적인 언어 이해시스템에 대한 연구가 활발해지고 있다. 즉 단순한 구문분석의 차원을 벗어나 활용하기에 편리하도록 설계된 내부지식 표현구조로 자연언어 문장의 의미를 정확하게 분석, 추출해 내는 자연언어 이해에 관한 연구가 다각도로 이루어지고 있다.

자연언어 이해에 관련된 연구분야를 좀더 세분하면, 첫째, 이해된 의미를 표현하는 내부지식 표현구조의 설계에 관한 연구, 둘째, 자연언어 문장으로부터 내부지식 표현을 추출해 내는 의미분석 방법에 관한 연구, 그리고 마지막으로 추출된 내부지식 표현의 활용에 관한 연구의 세가지로 구분된다.

2. 내부지식 표현구조의 설계

문장의 의미표현 형식은 응용시스템의 조작에 적합하도록 설계되어야 하며, 이제까지의 자연언어 이해연구를 통해 볼 때 초기의 연구는 Shank의 CD(conceptual dependency) 이론이 중심이 되어왔다.

이 CD이론은 의미소(semantic primitives)를 통해 모든 동사의 의미를 심층적으로 표현함으로써 문장의 의미를 정확히 표현하고자 하는 것으로서, 언어이해가 표면상의 언어표현에 원존하지 않는 측면을 정확하게 포착하기 위해 개념레벨의 이해를 강조한다. 즉 단어의 의미를 보다 기본적인 의미소와 개념소로 분해해서 문과 단어의 의미를 설명하고자 하는 CD이론에 기반한 연구는, 이전의 연구가 주로 구문분석 중심이었던 것에 반해 구문분석과 의미분석을 통합적으로 수행하는 접근방법을 취했으며 대개의 경우 의미분석에 중점을 두어 구문분석을 소홀히 하는 경향이 있었다.

그러나 같은 동사라 할지라도 사용되는 경우에 따라 다양한 의미를 지닐 수 있는 자연언어의 특성으로 인해 CD표현을 위한 의미분석 과정은 방대한 양의 의미정보를 활용해야 하는 과중한 부담을 안게 되었다. 뿐만 아니라, 자연언어의 모호성을 고려할 때 그 미묘한 어감을 파괴하지 않고 단일한 의미소를 이용한 표현으로 동사를 변환할 수 있느냐의 의문이 제기되면서 내부지식 표현에 관한 연구는 점차 논리(logic)나 의미 네트워크에 의한 지식표현을 중심으로 진행되게 되었다.

논리에 의한 연구는 표현력이 우수한 다차 술어논리(higher order predicate calculus)를 이용한 초기 연구로부터 시스템의 구현 가능성이 훨씬 높은 일차 술어논리(first order predicate calculus)를 이용하여 단순한 사실의 기술뿐만 아니라, 화자의 지식, 믿음의 차원에서 자연언어 문장의 의미를 분석, 표현해 내려는 시도가 이루어지고 있다.

또한 단어 상호간의 연상적인 관계를 네트워크로 받아들여 이 네트워크를 탐색하는 것에 의해 단어간의 의미적인 유사성과 관계를 도출하도록 하는 의미 네트워크 지식표현 구조가 고안되었다. 의미 네트워크의 연구는 인간 연상능력의 연구를 기초로 하며, 의미표현에 있어서 개념의 내포(intension)와 외연(extension)을 구분하여 각각 명제노드(propositional nodes)와 참조노드(referential nodes)로 표현하고 있다. 의미 네트워크는 실제의 코딩에서는 대개 논리와 유사한 형태로 표현되기 때문에 논리를 이용한 표현의 일종으로 볼 수 있지만, 일반적인 논리에 비해 훨씬 많은 제약이 가해진 형식화(formal)한 표현 방법이므로 표현력이 다소 떨어지는 반면 응용과 구현이 용이한 장점을 지니고 있다.

결론적으로 논리나 의미 네트워크에 의한 지식표현에서는 의미소의 설정에 많은 융통성을 두고 있으며 동사자체가 의미소의 역할을 하기도 한다. 이러한 융통성은 또한 응용분야에 따라 의미분석의 정도에도 융통성을 주어 보다 실용적인 응용시스템의 구현 가능성을 높이는 역할을 하였다.

3. 의미분석 방법

1980년대에 이르러 CD이론의 퇴조와 함께 자연언어 분석과정에서 구문분석의 중요성이 재고되게 되었다. 문장의 통사구조를 밝히는 구문분석을 많이 무시하였던 통합적 접근 방식대신에 정확한 통사구조를 찾아내어 이 구조를 문장의 의미분석에 이용하는 GPSG나 LFG(lexical functional grammar)와 같은 기법들이 등장하게 된 것이다.

GPSG는 많은 양의 의미정보를 처리하여 의미상으로도 완벽한 통사구조를 찾아내는 것을 의미분석의 목표로 하며, LFG는 먼저 문장의 통사구조를 밝혀내고 이 구조로부터 문장의 의미구조인 f-구조를 의미분석을 통해 얻어내는 접근방법을 취한다.

4. 응용시스템의 구성

본 절에서는 자연언어 이해 연구의 응용시스템의 한 예로 서울대 자연언어 연구실에서 실험적으로 개발된 한국어 자연언어 질의응답 시스템인 KQAS(korean question answering system)를 살펴봄으로써 구체적으로 응용시스템이 어떻게 구성되고 수행되는가를 알아보기로 한다.

KQAS는 자연언어 이해 연구추세를 한국어에 적용한 것으로서, 몇개의 문장으로 이루어진 간단한 한국어 문안내용을 이해하여 그 내용에 대한 질의응답을 한국어 자연언어를 통해 수행하는 시스템이다.

앞에서의 세가지 기준에서 보면, KQAS는 첫째, 내부지식 표현구조로 의미 네트워크의 수정·보완된 구조를 이용하고 있으며, 둘째, 의미분석을 위한 기법으로는 LFG를 이용하고, 끝으로 응용분야는 추론을 통한 질의응답 시스템이다.

다음은 KQAS의 문장처리 과정을 살펴보자. 사용자는 서술문과 질의문을 입력하고 시스템은 이를 구분하여 서술문의 경우 그 내용에 맞추어 의미 네트워크의 내부지식 표현구조의 내용을 갱신하며 질의문인 경우에는 추론을 통해 응답문을 생성해 출력한다. 서술문 분석과 질의문 분석과정은 구문분석 및 의미분석으로

구성되는 한국어 분석단계를 공유한다. 그러나 서술문의 경우에는 한국어 분석단계를 거쳐 분석된 문장의 의미를, 의미 네트워크 갱신단계에서 상응하는 내부지식 표현으로 나타내는 반면, 질의문의 경우에는 내부 질의문 표현단계, 추론단계, 응답문 생성단계를 거쳐 자연언어로 된 응답문을 출력하게 된다. 그림12는 개략적인 과정을 보여준다.

실제로 시스템은 그림13과 같이 7 개의 모듈로 구성되며 그들 각 단계에 대해 살펴보기로 한다.

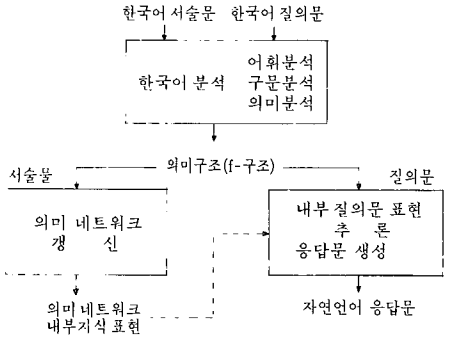


그림12. KQAS의 문장처리 과정

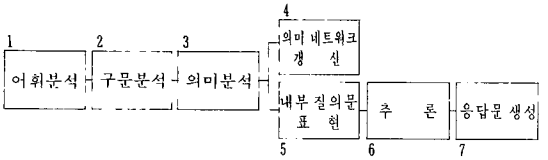


그림13. KQAS의 모듈구성

(1) 어휘분석과 구문분석

어휘분석에서는 한국어 형태소 분석이 수행되고 구문분석에서는 형태소 분석의 결과인 토큰열로부터 통사구조인 나무구조를 생성한다.

(2) 의미분석

이 단계에서는 앞에서 생성된 나무구조로부터 의미구조인 f-구조를 생성해 낸다. KQAS에서는 통사구조의 각 문법규칙에 해당하는 의미가 상향(bottom up)으로 삽입, 통합되면서 f-구조가 구성된다.

(3) 의미 네트워크 갱신

서술문의 f-구조를 입력으로 그 의미에 맞추어 내부지식 표현인 의미 네트워크의 내용을 갱신하는데, 세부적으로는 f-구조를 추출, 변형, 외연노드 삽입, 동일 외연 처리 및 의미 네트워크 생성단계로 이루어져서 다양한 개념의 종류 및 성질을 반영하여 여러가지 개념의 성질을 올바로 처리하도록 설계, 구현되었다.

(4) 내부 질의문 표현

KQAS의 질의문은 단문으로 제한하였으며 yes-no 질의문과 wh-질의문의 두가지 형태를 허용하고 있다. 질의문의 형태가 먼저 판별되고, 다음 f-구조 추출 및 변형단계를 거쳐 내부표현 구성단계로 들어간다.

(5) 추론

내부질의문 표현을 입력으로 하여, 의미 네트워크의 내부지식 표현의 내용을 이용한 추론의 결과를 응답정보로 출력한다. 응답정보는 yes-no 질의문에 대해 yes, no 혹은 maybe가 출력되며, wh-질의문에 대해서는 만족하는 외연노드의 리스트가 된다.

(6) 응답문 생성

앞에서 생성된 응답정보를 어휘분석 단계의 토큰열과 함께 입력하여 자연언어 응답문을 생성한다. 응답문은 질의문의 토큰열이 응답정보에 따라 변형된 형태이며 변형은 조건-처리 규칙(condition action rule)을 통해 다양한 유형으로 처리된다.

Ⅳ. 결 론

이상에서 살펴본 것과 같이 자연언어의 처리는 차츰 현실화 되어가고 있으며 그 응용면이 다양한 것도 짐작할 수 있을 것이다.

인간의 행위중에 자연언어를 이용한 행위는 그 범위로나 양으로 큰 비중을 차지하고 있음은 상식으로 알려져 있을 정도이다.

특히 지능이 높은 문제일수록 그 표현방법은 자연언어일 수밖에 없다는 것은 부인할 수 없다.

언어의 기계번역은 이러한 개념을 확인하는데 좋은 계기가 될줄로 본다. 가장 근본적인 구성이면서 언어 전체의 구조나 내용을 관찰해 보는데 충분한 계기가 될 수 있다.

이제 기계번역이 본궤도에 오르면서 인간은 언어처리의 기술을 터득하게 되면서 자연언어 처리분야에 산재하여 있는 많은 일거리들을 본격적으로 취급하는 계기를 맞을 것으로 본다.

參 考 文 獻

[1] 강승식외, "KSHALT : 한-영 기계번역 시스템," 87년 춘계학술 발표회 논문집, 인공지능 연구회, 1987.

[2] 윤덕호외, "한국어 자연언어 처리를 위한 시맨틱 네트워크에 관한 연구," 86년 가을 학술발표 논문집(上), 한국정보과학회, 1986.
[3] David, L. Waltz, "The State of the Art in Natural Language Understanding," ed. by Barr, Avron et al., The Handbook of Artificial Intelligence, vol. 1, 1982.
[4] Johnson, Tim, Natural Language Computing: the Commercial Applications, Ovum Ltd, London, 1985.
[5] King, Margaret, "The Prospect of Machine Translation", The 7th European Conference on Antificial Intelligence, Proceedings, vol. 1, July 1981.
[6] Slocum, Jonathan, A Survey of Machine Translation: Its History, Current Status, and Future Prospects," Computational Linguistics, vol. 11, no. 1, January-March 1985.
[7] Wendy, G.L., et al., "Stategies for natural language processing", William Kaufmann, 1981. *

♣ 用 語 解 說 ♣

Concurrency Operation(Real-Time) : 병행수행연산(실시간)

실시간 처리나 일괄 처리를 각각 별도로 수행하는 것이 아니라 실행 제어 시스템의 통제하에 실시간 처리와 일괄처리를 동시에 수행한다. 실시간 처리 작업은 항상 우선 순위를 가지며, 이에 따라 자원도 분배해 준다. 실시간 처리 업무가 없으면 과학 계산이나 사무 업무처리 등의 일괄 처리가 다음 우선권을 갖는다. 이와 같은 방법으로 실습간 시스템을 구성하여 자원을 최대한 활용할 수 있고, 단위 작업당 낮은 가격으로 기억용량, 속도, 융통성, 통신 능력등의 장점을 살릴 수 있으며, 여러가지 넓은 응용 분야에서 다른 시스템 보다 많은 장점을 갖는다.

Cross Tracking(십자추적)

디스플레이상에 배열한 밝은 점에 의한 십자모양. 점과 선들을 배치하거나 곡선을 그리는데 사용된다.

Cycle Time Processor(처리기 사이클 시간)

컴퓨터는 크게 산술 및 제어장치, 입출력장치, 기억 장치로 나눌 수 있다. 산술 및 제어 장치는 프로그램 지시를 따르며, 중앙처리 장치는 계산, 정보의 이동이나 제어를 담당하고 있고, 중앙 처리 장치를 오고 가는 모든 정보는 입출력 장치를 거친다. 또한 모든 주변 장치의 동작을 제어한다. 기억 장치는 중앙처리 장치의 심장부이며 자료나 명령의 중간처리 결과를 저장한다. 기억장치의 주기 시간은 컴퓨터의 전반적인 속도를 결정하는 중요한 요소이다.

Data Manipulation(데이타 조작)

자료의 분류, 갱신, 변경, 첨가, 제거, 입출력 동작, 보고서 작성, 정렬등과 같은 자료처리 작업

DBMS : Data Base Management System(데이타 베이스 관리 시스템)

Deposit (저장)

기억 장소의 내용을 보관하거나 보조 기억장치에 기록하여 두는 것.

Encode(암호화)

(1) 각각의 문자나 전달문 내의 여러개의 문자를 나타내기 위하여 코드를 만드는 것을 말함. 주로 이진수를 사용한다. (2) 암호화시키는 방법을 아는 몇몇 개인을 제외하고는 전달문의 의미를 의도적으로 감추기 위하여 문자나 숫자로 대체하는 것.