SHIFT-REDUCE 알고리즘을 이용한 한국어 자동 분석 기법

김 지은
(한국 외국어 대학교)

A Parsing Technique for Korean Using Shift-Reduce Algorithm

Kim, Jee Eun
(Hankuk University of Foreign Studies)

요 약

본 논문은 PC 환경에서 한국어 문장구조를 분석할 수 있는 분석 기법을 제시한다. 상대적으로 어순이 자유로운 언어인 한국어의 특성에 중점을 두어, 이를 효과적으로 처리할 수 있는 분석 기법으로 shift-reduce 알고리즘을 제시한다. shift-reduce 분석 기법은 구문론 및 의미론적 하위 범주화에 의한 분석을 효율적으로 실행할 수 있도록 해 주며, bottom-up과 left-right에 의한 분석 과정을 보완하여 준다.

I. 서론

최근 한국어의 자연어 처리를 위한 시스템 개발에 대한 연구가 활발하게 이루어지고 있다. 문어 혹은 구어를 대상으로 언어를 보다 체계적으로 처리할 수 있는 문법 개발과 이상적인 시스템 개발에 연구의 관심이 모아지고 있다. 이상적인 시스템이란 주어진 임무를 효과적으로 처리하되 처리된 결과가 정확해야하며, 처리 과정 또한 타당한 이론적 근거가 있어야 한다.

지금까지 자연어의 구조체계를 규명하고자 수많은 문법체계가 연구, 제시되어 왔다. 이러한 이론적 발전에도 불구하고 자연어 처리 시스템은 언어 구조를 효과적으로 구현할 수 있는 체계를 아직 확립하지 못하고 있다. 이는 개별 언어의 특성을 연구하기 보다는 언어가 갖는 보편성에 촛점을 맞추는 것에 기인한다. 또한 이러한 문법체계는 단순성을 강조하게 되기 때문에 실제 데이타를 전산처리에 적용을 시켰을 경우 많은 문제점을 야기시킨다. 따라서 언어의 효과적인 구현을 위해서는 언어학적 이론에 비춰 타당하되, 데이타를 중심으로 언어 현상을 관찰함으로써 개발되어야 한다. 여기에 '전산처리'라는 상황이 반영되어 개발되어야만 실용성 있는 문법이라 할 수 있을 것 이다. 또한 언어의 구조(형태, 구문 및 의미)를 구현하는 데 있어 효율성이 반드시 고려되어야 하는데, 전산 처리 기법의 효과적 활용으로 그 효율성을 증가시킬 수 있다.

본 논문에서는 한국어 자연어 처리를 위한 기본 문법과 분석 기법을 제시한다. 전산 처리시에 이용되는 문법을 개발하기 위하여 기존의 특정 이론 대신 ATN을 도입하여 언어 구조를 구현하고자 하였으며, ATN을 보완하고, 효율성을 높이고자 mixed-mode 및 shift-reduce 분석 기법을 이용하였다.

II. 한국어 특성의 고찰 및 분석방법

본 장에서는 한국어 분석을 위해 유형론(Typology)적 입장을 중심으로 한국어의 특성을 기술하며, 이에 따른 분석방법을 제시한다.

한국어는 SOV, 즉, 주어, 목적어, 서술어의 어순을 갖으나, 이 중 서술어를 제외한 문장 구성 요소들간의 어순이 상대적으로 자유롭다. 상대적으로 자유로운 어순은, 아래 예문 1]a - f에서 볼 수 있듯이, 조사(혹은 후치사)의 활용으로 인해 가능하다.

1] a. 영희가 철수에게 편지를 보냈다.
b. 영희가 편지를 철수에게 보냈다.
c. 철수에게 영희가 편지를 보냈다.
d. 철수에게 편지를 영희가 보냈다.
e. 편지를 영희가 철수에게 보냈다.
f. 편지를 철수에게 영희가 보냈다.

위의 예는 화용적 입장에서는 의미의 차이가 있으나 구문구조의 관점에서는 동일한 구조라 간주된다.

자유어순을 가능하게 해 주는 조사는 표층격을 결정하는데 큰 역할을 한다. 표층격은 조사와 더불어 하위범주화 정보에 의해 결정되며, 심층격을 결정하는 데 trigger 역할을 한다. 심층격은 격문법(case grammar)을 근거로 분류된 서술어의 격틀(case frame)에 의해 결정된다. 심층격은 영어와 같은 언어에서는 의미 자질과 어순에 의해 표층격과 연결되는 반면, 한국어는 의미자질과 조사에 의해 표층구조에 나타난다.

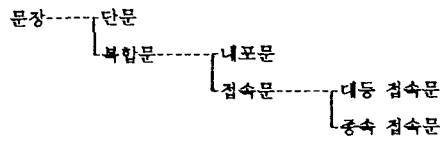
2] a. 영희가 자고 있다.
b. 영희가 철수를 좋아한다.
c. 영희가 책을 받았다.
d. 영희가 철수에게 선물을 주었다.

위의 예문에서 보면 2] a - d에 쓰인 '영희'는 모두 주어의 역할을 하나, 이에 대응되는 심층격은 각기 다르다. Object, Expriencer, Benefactive 및 Agent가 a - d에 각각 적용된다.

SOV의 어순을 가진 언어에서 대체적으로 볼 수 있듯이, 한국어는 규칙적 left-branching 언어이다. 즉, 아래 3]a - b에서와 같이, 형용사, 지시사, 소유격, 관계절 등을 포함한 모든 수식어는 반드시 피수식어 왼쪽에서 수식한다.

3] a. 저 작은 소녀의 예쁜 눈
b. 어제 만난 소녀

한국어의 복합문은 단문의 결합 형태에 따라 내포문과 접속문으로 세분된다.



내포문은 관계문 4]a & b, 보문 4]c & d, 명사형 4]e, 분열문(cleft) 4]f로 세분된다. 내포문이 모문(matrix sentence)과 같은 명사구를 문장성분으로 갖으면, Equi-NP-Deletion 규칙에 의하여 이 명사구는 내포문의 주어와 모문의 주어 혹은 목적어가 되고, 내포문의 명사구는 탈락된다.

4] a. 영희가 철수가 준 책을 읽고 있다.
b. 영희가 철수에게서 받은 책을 읽었다.
c. 영희가 철수에게 집에 가라고 말했다.
d. 영희가 열심히 공부했다고 말했다.
e. 영희가 놀러 다니기를 좋아한다.
f. 영희가 좋아하는 것은 여행이다.
g. 영희는 비가 오지 않으면 외출을 할 것이다.
h. 비가 오지 않으면, 영희는 외출을 할 것이다.
i. 영희가 철수가 좋다.

한국어의 내포문과 종속 접속문은 시작점을 나타내는 표식이 없다. 그 표식은, 문장 4]a - h에서와 같이, 내포문 혹은 접속문의 술어에서 찾을 수 있다. 문장 4]a를 보면 동일한 주격이 사용된 명사구가 존재하는 경우로, 두번째 주격인 '철수가'를 내포문의 시작점으로 간주할 수 있다. 한국어에서는, 위의 4]i에서 볼 수 있듯이, 한 문장내에 '이중 주어' 혹은 '겹주어'의 사용이 가능하며, 내포문의 주어와 구별되기 쉽지 않다. 따라서 내포문의 시작점은 사실상 존재하지 않는다고 하겠다. 내포문 뿐만 아니라 종속 접속문 역시 4]g - h에서와 같이 시작점이 없으며, 위치 또한 제약을 받지 않는다.

한국어에서는 아래 5]a - c에 나타난 것 처럼 담화 내용상 성분이 확실한 요소는 생략이 가능하다.

5] a. (영희가) 사과를 먹는다.
b. 영희가 (사과를) 먹는다.
c. (영희가) (사과를) 먹는다.

이러한 경우, 영어와 같은 언어에서는 조응사(anaphor)를 사용하나, 한국어에서는 그 흔적을 남기지 않으므로 (zero anaphora) 이러한 특성은 전산 처리시 어려움을 유발시킨다.

SOV어순을 갖는 언어에서는, 아래 예 6]a - d에서 볼 수 있듯이, 반드시 용언이 보조 용언을 선행한다.

6] a. 먹고 있다.
b. 먹어 보지 않았다.
C. 먹어 보려고 한다.
d. 먹게 했다.
e.*먹게 보다.

보조 용언이 사용될 경우, 본용언과 연결지어 주는 어말 어미가 활용이 되며 (6]a - d), 어말 어미와 보조 용언의 결합에는 6]e에서와 같이 제한이 주어진다. 시제는 보조 용언이 한개 이상 사용되었을 경우, 항상 마지막 보조 용언에 표시된다.

III. 구현

본 논문에서 제시된 분석 방법은 PC 환경에서 LISP에 의해 구현되었으며, 단일 문장을 대상으로 한다. 분석 과정은 어절을 기본단위로 하여 형태소 분석과 사전 검색을 통하여 얻어진 정보를 근거로 진행된다. 처리 과정에서는 위에 기술된 한국어의 특성을 고려하되, 효과적으로 처리하기 위하여 여러가지 분석기법을 적용한 알고리즘을 이용하였다. 먼저 기본이 되는 문법의 틀로서 cascaded ATN을 도입하되, ATN의 top-down 처리 방식을 top-down과 bottom-up의 분석 방법이 병행되는 mixed-mode로 전환하였다. mixed-mode 기법은 한국어의 특성 중 상대적으로 자유로운 어순이지만 서술어가 항상 문장의 후반에 온다는 점과 내포문의 시작점이 없다는 점을 보다 효율적으로 처리할 수 있도록 한다. 여기에 단순한 bottom-up 분석에서 발생할 수 있는 비효율성을 보완하고자 shift-reduce 분석 기법을 도입하였다. shift-reduce 기법은 또한 ATN의 left-to-right에 의한 처리 방식을 보완하며, 한국어의 left-branching 현상과 서술어가 문장 후반에 오는 특성을 효율적으로 처리하도록 한다. 즉, 하위 범주화 및 심층격을 결정하는 서술어가 문장후반에 위치하므로, 서술어가 처리되기 전까지 분석된 문장 구성 요소들을 shift-stack에 저장해 둠으로써 구문 구조에 대한 결정을 최적시까지 미루게 된다. 최대한의 정보가 확보될 때까지 결정을 미룸으로써 구문의 정확한 분석을 유도해 낼 수 있도록 한다.

분석 과정은 문장이 입력되면, mixed-mode 중 우선 top-down 기법이 적용됨으로써 실행된다. 한국어의 모든 문장은 서술어가 마지막에 위치한다는 규칙에 의해, 서술어에 앞서 모든 문장 구성 요소를 분석하게된다. bottom-up 기법은 서술어이외의 문장 구성 요소를 분석할 때 활용되며, 한국어의 상대적으로 자유어순을 고려하여 적용되었다. 명사류(nominals)와 부사류(adverbials)는, 아래 예 7]a - f에서 볼 수 있듯이, 명사와 조사로 구성되는 동일한 구조를 갖으며, 이들의 문장상에서의 위치도 자유롭다 (1]a - f 참고).

7] a. 학교-가/는
b. 학교-를
c. 학교-에/에서
d. 학교-로
e. 학교-에서/로부터
f. 학교-까지

위와 같은 이유로 이들에 대한 분석은 특정 구문 규칙과 상관없이 처리된다. 이들의 문장내에서의 역할은 분석이 완료된 후, 조사에 의해 결정된다.

본 시스템에 도입된 한국어의 구문 규칙은 단순화 되었으며, 그 규칙은 아래와 같다.

8] a. S --> PP Pred
b. PP --> (Det) (Adj) N Post
c. Pred --> V (Aux)

한국어 문장은 PP 즉, 후치사구(Postpositional Phrase)와 술어(predicate)로 구성된다. 후치사구는 의무적(obligatory) 요소로 명사와 후치사를 포함하여야 하며, 수의적(optional)으로 한정사(determiner)와 형용사를 수반할 수 있다. 서술어는 동사 혹은 서술 형용사와 보조용언들의 집합을 뜻한다. 이러한 규칙에 의해 서술어를 제외한 모든 문장 요소들은 단일 단계(state)에서 처리된다. 이들에 대한 분석 결과는 shift-stack에 저장되었다가 서술어의 처리가 끝나면, 다시 retrieve되어 서술어의 정보와 대조, 확인된다.

보조 용언을 포함한 모든 서술어도 단일 단계에서 처리된다. 서술어의 어말 어미가 '아/어', '게', '지', '고', '려고' 등일 경우는 보조용언의 수반을 의미하고, 어미가 문장의 완료를 나타내는 형태소일 경우 서술어의 분석을 마치게 된다. 이 단계에서 사전에 수록된 서술어의 하위 범주화 정보와 실제로 문장에 쓰인 논항들이 비교되어 단일화(unification)에 의해 처리된다. 현 시스템의 단일화 과정에서는 통사론 규칙이 적용되며, 이를 근거로 하여 비문을 가려내게 된다. 문장상의 논항이 생략되지 않은 경우, 논항이 적법한 자질을 갖고 있는지, 조사의 활용이 제대로 되었는지 등이 확인된다. 확인이 성공적으로 끝나게 되면, 처리되어 저장된 정보를 바탕으로 문장 요소들의 역할이 각각 결정되며, 일단 입력문은 통사론적 입장에서 정문이라 볼 수 있다.

다음 단계에서는 논항의 심층격이 결정된다. 각 논항은 술어의 사전정보에 의해 격틀과 대조, 확인되고, 선별적 제한(Selectional Restrictions)을 통해 테스트가 되어 의미론상

의 적법 여부를 가리게 된다. 의미상으로도 정문으로 판정이 될 경우 모든 분석 과정이 완료되며, 처리의 결과로서 parse tree가 생성된다. 본 시스템에 의해 분석 한 단문의 Parse tree는 다음과 같다.

eg) 나는 오늘 점심을 먹지 않았다.

(subj 나_는_ilinching)
(adv 오늘_adv)
(obj 점심_을_umsik)
(mainv 먹_지_basicact)
(finalv 않_았다_proc-dscrp-aux)
(s-subj 나_는_ilinching)
(s-adv 오늘_adv)
(s-pred (먹_지_basicact 않_았다_proc-dscrp-aux))
(s-tense past)
(s-mood decl)
(agt 나_는_ilinching)
(thm 점심_을_umsik)

IV. 결론

본 논문에서는 한국어의 전산처리를 위한 기본 문법 및 분석 방법에 대하여 기술하였다. 한국어에 대한 이론적 배경에 대해 기술하였으며, 효과적인 전산 처리를 위하여 고려해야 할 문제들을 제시하였다. 구현의 문제에 있어서는 cascaded ATN을 기본틀로 하되 한국어의 특성을 고려하여 mixed-mode 및 shift-reduce 분석 기법을 이용하였다.

본 시스템을 실용화 될 수 있는 것으로 발전시키기 위해서는 다음의 몇가지 사항을 연구, 첨가시켜야 할 것이다. 우선 처리 범위가 단일 문장에 한정되므로 그 범위를 단화(discourse) 수준으로 확대시켜야 한다. 처리 대상이 담화가 될 경우, 현 시스템을 이용한 분석의 결과로서 생성되는 parse tree로부터 기억된 정보를 이용하여 조응관계를 밝힘으로써 문맥상의 흐름을 파악할 수 있다. 본 연구에서는 대등 접속문은 다루어지지 않았으나, 문법에 대한 약간의 수정으로 처리될 수 있다. 또한 기본 문법의 틀에서 벗어나 고유의 생성 규칙을 갖는 관형적 표현에 관한 연구도 첨가되어야 할 것이다.

[주]

본 논문은 Ph.D. 학위 논문(1993)의 일부로 개발된 시스템에 관한 기술로 학위 논문의 일부를 요약, 수정한 것이다. 이 시스템은 로만자로 표기된 한글 text를 분석하도록 개발되었으나, 본 논문에서는 편의상 예문을 한글로 표기하였다.

참고문헌

Cook, Walter A. 1989. Case Grammar Theory. Washington D.C.: Georgetown University Press.

Greenberg, Joseph H. 1961. Some Universals of Grammar with Particular Reference to the Order of Meaningful Elements. In Universals of Language, ed. by Joseph Greenberg. Cambridge, MA:MIT Press.

Kim, Jee Eun. 1993. Semantic Subcategorization of Korean for Natural Language Processing. Unpublished Ph.D. dissertation. Gerogetown University: Washington, D.C.

Loritz, Donald. 1990. Using Artifitial Intelligence to Teach English to Deaf People. Georgetown University, School of Language and Linguistics: Washington, D.C. Language Research Laboratories Report #RX2500-950F.

Shieber, Stuart M. 1983. Sentence Disambiguation by Shift-Reduce Parsing Technique. In Proceedings of the 21st Annual Meeting of the Association for Computational Lingustics.

Woods, William A. 1980. Cascaded ATN Grammars. In American Journal of Computational Linguistics, 6 (January - March): 1-12.

Yi, Hanso! H.B. 1989. Korean Grammar. New York: Oxford University Press.