복합 단위 정보를 이용한 차트 파서

정한민*, 여상화, 김태완, 박동인
기계번역연구실/자연어정보처리연구부/시스템공학연구소

Chart Parser Using Compound Unit Information

Hanmin Jung, Sanghwa Yuh, Taewan Kim, Dong-In Park
Machine Translation Laboratory, Natural Language Information Processing Department, SERI
{jhm, shyuh, twkim, dipark}@seri.re.kr

요 약

본 논문은 복합 단위 정보를 이용하여 모호성을 감소시키고 자연스러운 대역어 정보를 제공할 수 있는 차트 파서를 기술한다. 복합 단위 정보를 사용하는 파싱은 태깅과 구문 분석 과정 사이에서 여러 단어들을 하나의 단위로 만들어서 형태론적/구문적 모호성과 파스 트리의 수를 감소시킨다. 우리는 Bottom-up 차트 파싱을 사용하는데, 이는 모호성 있는 태깅 결과가 많을수록 파스 트리의 생성 시간과 수의 증가를 초래하므로 복합 단위를 사용하여 파서에 대한 입력 단어의 수 및 모호성을 감소시켜 안정적인 파싱 결과를 얻을 수 있게 한다. 실험 결과는 복합 단위 정보를 사용한 차트 파싱이 차트들의 크기와 파스 트리의 수를 50%까지 감소시킴을 보여준다.

1. 서론

구문 분석기가 품사 태깅된 문장을 입력으로 받아 파싱을 수행하여 생성한 파스 트리의 수와 입력 문장의 품사 태그를 여과하고 가능한 패턴들을 묶어서 재구성한 문장을 입력으로 받아 생성한 그것과는 많은 차이가 있다. 예를 들어, "The Latin American has paid on real old debt since early last year."의 파싱 결과와 "The *Latin_American* has *paid_on_real_old_debt* since early *last_year*."의 그것과는 입력 단어의 수에 (13 개와 7 개) 있어서 뿐만 아니라 패턴의 적용으로 인한 형태론적 모호성의 수에 있어서도 차이가 날 수 밖에 없다.

우리는 복합 단위 정보를 이용하여 모호성을 감소시키고 자연스러운 대역어 정보를 제공할 수 있는 차트 파싱 기법을 소개하고자 한다. 복합 단위 정보를 사용하는 파싱은 태깅과 구문 분석 과정 사이에서 여러 단어들을 하나의 단위로 만들어서 형태론적/구문적 모호성과 파스 트리의 수를 감소시킨다. 파싱을 위해서 Bottom-up 차트 파싱을 [조혁규, 1990] 사용하는데, 이는 부분적인 파싱이 용이하며, S로 시작하는 완전한 파스 트리의 생성에 실패하는 경우에도 문장 전체에 대한 부분 파싱 결과를 얻기에 용이하기 때문이다.

기존의 차트 파싱 기법을 사용한 연구들은 [Den, 1994] [Kato, 1994] [박성숙, 1993] 파싱 자체의 개선이나 부가 정보를 이용하여 시간 복잡도의 감소를 포함하는 파싱 속도/적용 범위의 개선에만, [Yoon, 1994]은 숙어 패턴의 인식을 통한 기계번역 시스템의 속도 개선에만 관심을 가지고 있었다. 비록, [Lee, 1994]가 숙어 패턴을 차트 파싱에서의 하나의 Edge로 표현하고자 하는 시도를 하였지만, 복잡한 형태의 복합 단위 정보를 파싱에 적용하고, 가변 요소 등에 대한 부분 파싱을 제안하여 후속 모듈들과의 인터페이스를 통한 보다 자연스러운 대역어를 생성할 수 있는 방법을 제시하지는 못했다.

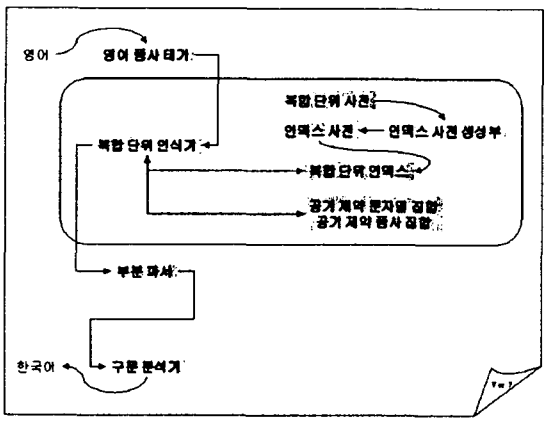
본 논문은 실험을 통하여 복합 단위 정보를 이용한 차트 파싱과 문맥에 따라 다른 대역어로 바뀌는 가변 요소들에 대한 부분 파싱에 의해 차트들의 크기와 파스 트리의 수가 약 50%까지 감소됨을 보여준다.

2. 복합 단위 정보

복합 단위는 연어, 숙어, 복합 명사들을 포함하는 문맥 상에서의 고정된 표현들을 총칭하는 복합 개념으로, 품사 태깅 과정에서 발생하는 형태론적 모호성을 일부 제거하며 자연스러운 대역어 정보를 제공하여 생성 모듈의 부담을 감소시키는 것을 목적으로 한다. [Jung *et al.*, 1997a] [Jung *et al.*, 1997b]. 또한, 문장의 길이,

복합 단위, 사전의 크기 등에 대해서 선형적인 처리 시간을 보여주는 복합 단위 검색 구조를 사용하여 복합 단위 인식기를 포함하는 전체 시스템에 시간적인 부담을 거의 주지 않는다 [정한민, 1997].

복합 단위 인식기는 품사 태깅 모듈과 구문 분석 모듈 사이에 위치하여 입력 문장으로부터 모든 가능한 복합 단위들을 발견한다. 그림 2-1은 복합 단위 인식기의 시스템 구성을 보여준다. 복합 단위 인식기는 텍스트 파일 형태의 복합 단위 사전을 메모리로 올린 형태인 복합 단위 인덱스 상에서 품사 태깅된 문장으로부터 복합 단위를 검색한다.



[그림 2-1] 복합 단위 인식기의 시스템 구성도

복합 단위 정보는 인식된 복합 단위가 가지고 있는 정보를 의미한다. 인식된 복합 단위는 대표 품사 태그, 인식 심벌, 위치 정보를 가진다. 이 정보들은 파싱 과정에서 문장 내의 다른 단어들과 복합 단위를 구별하고, 접속 정보 검사를 수행하기 위해 사용된다.

대표 품사 태그는 복합 단위를 대표하는 품사 태그이며, 복합 단위가 동사 활용 형태를 키로 가지는 경우에는 그 활용 품사 태그를 그대로 대표 품사 태그로서 사용한다 (예. 복합 단위 "abide by"가 문맥 상에서 "abides by"의 과거형으로 나타나는 경우에는, 대표 품사 태그로 VB (동사 기본형) 대신에 VBZ (동사 현재형)를 사용).

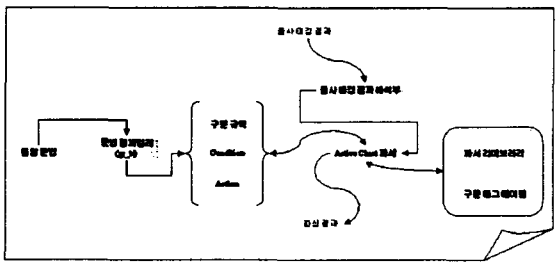
인식 심벌은 인식된 복합 단위나 그 내부의 가변 요소를 표현하기 위한 것으로, 복합 단위는 {*n | n = 1, 2, 3, ...}의 형태로, 가변 요소는 {#n | n = 0, 1, 2, ...}의 형태로 표현된다 (예: "*take for Mr. Lovery him*"은 복합 단위 인식 후에 "take (*1) for Mr. (*2:*1#1) Lovery (*2#1) him (*1#2)"의 인식 심벌들을 가짐). 이러한 표현 형식은 내포 구조 형태로 나타나는 복합 단위들을 처리할 수 있는 방법을 제공한다.

위치 정보는 복합 단위의 범위를 표현하기 위한 정보로 각 복합 단위의 키나 가변 요소의 첫번째 단어에 붙는다 (예: 문장 "I *kept the words in mind.*"내의 복합 단위 "keep #1 in mind"는 "keep"에 위치 정보 <2-6>을, #1 (the words)에 위치 정보 <3-4>를 부여).

3. 파서

본 시스템에서 사용하는 파서는 영어 품사 태거와 의미 분석기 사이에 위치하여 구문 분석을 수행한다. 파서는 구문 분석 규칙과 연동되며, 복합 단위 인식기가 추가되는 경우에는 복합 단위 인식기와 의미 분석기 사이에 위치하게 된다. 영어 품사 태거에 의해 품사 태깅된 입력 문장과 확률 정보 등을 포함하는 각 품사를 위한 정보가 파서의 입력이 되며, 파서는 이들과 구문 분석 규칙을 이용하여 영어 파싱 결과를 낸다. 이때, Active 차트 파싱 알고리즘이 사용된다.

그림 3-1은 파서의 시스템 구성을 보여준다. 자체 기술 언어로 작성된 통합 문법은 문법 컴파일러를 통해 구문 규칙, Condition과 Action으로 분리된다. 품사 태깅 결과 해석부는 태깅된 품사 목록을 읽어 들여 파서를 위한 포인터로 연결된 자료 구조를 생성한다. Active 차트 파서는 품사 태깅 분석 결과와 시스템 리소스들 (구문 분석 규칙, Condition, Action, 파서 라이브러리, 구문 태그 테이블)을 이용하여 Active 와 Inactive 차트 목록을 작성하면서 차트 파싱을 수행해 나아간다. 파서에 의해 만들어진 파싱 결과는 의미 분석기 -> 변환기에 넘겨진다.

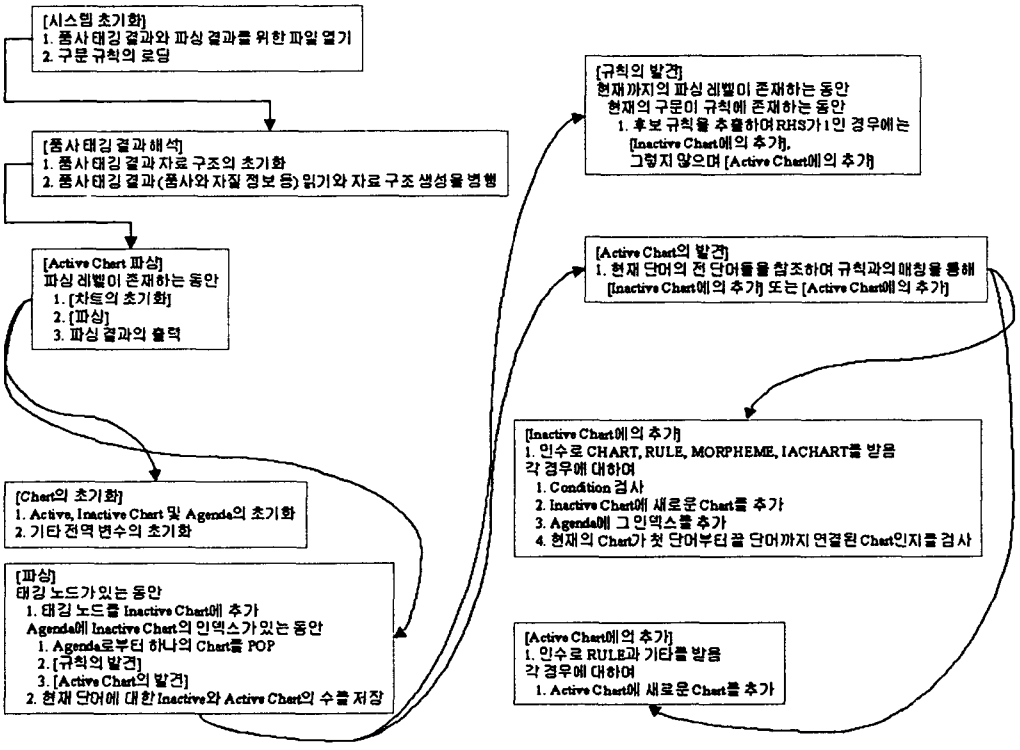


[그림 3-1] 파서의 시스템 구성도

그림 3-2는 파서의 전체 수행 알고리즘을 보여준다. 함수들 간의 제어 흐름 블록은 구문 규칙 로딩 등을 수행하는 시스템 초기화 과정, 품사 태깅 결과에 대한 자료 구조를 생성하는 품사 태깅 결과 해석 과정, Active 차트 파싱을 수행하는 Active 차트 파싱 과정, 차트의 초기화 과정, 파싱 과정, Active 차트의 발견 과정, Inactive 차트에의 추가 과정 및 Active 차트에의 추가 과정으로 구성된다.

4. 복합 단위 정보와 차트 파싱의 결합

그림 4-1은 복합 단위 인식기와 차트 파서가 결합된 시스템 구성을 보여준다. 영어 품사 태거로부터의 품사 태깅된 입력 문장 내에 복합 단위 인식기를 거치면서 복합 단위 정보가 추가된다.



[그림 3-2] 파서의 전체 수행 알고리즘

품사 태깅 결과 해석부와 Active 차트 파서를 거치면서 모호성과 파스 트리 수가 감소된 파싱 결과를 얻게 된다. 복합 단위 인식기와 차트 파서와의 결합을 위해서는 먼저 기존의 복합 단위 인식기의 자료 구조를 파서의 자료 구조와 동기화시키는 작업이 필요하다. 또한, 가변 요소 등을 포함하는 복잡한 형태의 복합 단위 처리를 위한 복합 단위 자료 구조의 정리가 필요하다.

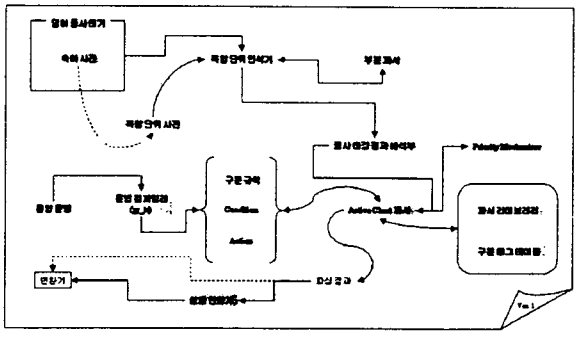
또한, 가변 요소를 포함하는 복합 단위를 차트 파싱에서 이용하기 위해서는 가변 요소에 대한 별도의 처리 모듈이 필요하다. 우리는 가변 요소에 대하여 부분적으로 파싱을 수행하여 그 결과를 따로 저장하며, 파스 트리에서는 이 결과에 대한 링크를 가지는 방법을 사용한다. 이런 방법은 가변 요소에 대한 부분 파싱 결과의 대역어를 따로 얻을 수 있으며, 이 대역어를 다시 복합 단위 대역어의 해당 위치에 삽입하여 자연스러운 번역 결과를 얻을 수 있도록 해준다 [부록 1].

우리는 Penn Treebank의 [Marcus *et al.*, 1993] WSJ에서 수동적으로 추출한 1268 문장을 복합 단위 인식을 위한 실험 대상으로 하였다. 이 문장들에는 1194 개의 복합 단위들이 포함되어 있으며, 평균 단어 수는 15.33 개이다. 문장 당 평균 1.29 개의 복합 단위가 있으며, 발견된 1636 개의 복합 단위들 중 56.26%가 두 개 이상의 명사로 구성된 복합 명사이며, 29.58%가 동사구이며, 14.16%가 기타 복합 단위들이다.

복합 단위 인식기만의 성능은 Recall이 97.65%, Precision이 98.52%이다. 부분 파서 (복합 단위 인식의 성능을 향상시키고자 가변 요소들에 대해 구문적 검증을 수행하는 모듈)의 도입은 Precision을 99.69%까지 높여준다.

복합 단위 정보를 이용하여 차트 파싱을 수행하는 실험은 5 개의 샘플 문장들을 대상으로 하여, Inactive 차트의 수를 53.7%로, Active 차트의 수를 61.1%로, 파스 트리의 수를 50%로 감소되는 결과를 얻었다 (각 문장 당 복합 단위 1 또는 2 개 포함; 부록 2). 부록 1은 복합 단위 및 가변 요소에 대한 링크를 포함하는 파스 트리의 한 노드와 해당 가변 요소 노드를, 부록 2는 복합 단위 정보를 이용하지

않는 경우와 하는 경우에의 Inactive 차트, Active 차트 및 파스 트리 수의 변화 예를 보여준다.



[그림 4-1] 복합 단위 인식기와 차트 파서가 결합된 시스템 구성도

5.결론

우리는 Bottom-up 차트 파싱에 복합 단위 정보를 사용하여 파서에 대한 입력 단어의 수 및 모호성을 감소시켜 안정적인 파싱 결과를 얻을 수 있음을 실험을 통해 보여주었다. 복합 단위 정보는 품사 태깅된 입력 문장으로부터 추출된 복합 단위가 가지고 있는 정보로 대표 품사 태그, 인식 심벌, 위치 정보로 나누어진다. 이 정보들을 이용하여 품사 태깅된 결과를 여과/결합함으로써 파서는 시간적/공간적으로 안정된 모습을 가질 수 있게 된다.

실험 결과는 복합 단위 정보를 이용하는 경우에 파서가 생성하는 Inactive 차트의 수가 53.7%로, Active 차트의 수가 61.1%로, 파스 트리의 수가 50%까지 감소됨을 보여준다.

앞으로의 연구 범위에는 보다 많은 문장들을 대상으로 하여 복합 단위 정보를 이용한 파싱의 효율성을 증명하는 것과, 가변 요소들을 위한 부분 파싱 결과의 후속 모듈에서의 실제적인 이용이 포함될 예정이다.

참고 문헌

[Den, 1994] Y. Den. Generalized Chart Algorithm: An Efficient Procedure for Cost-Based Abduction. In *Proceedings of ACL*, 1994.

[Jung *et al.*, 1997a] H. Jung *et al.* Compound Unit Recognition for Efficient English-Korean Translation. In *Proceedings of ACH-ALLC*, 1997.

[Jung *et al.*, 1997b] H. Jung *et al.* Multilingual Approach with Compound Unit. In *Proceedings of DIALOGUE*, 1997.

[Kato, 1994] T. Kato. Yet Another Chart-Based Technique for Parsing Ill-Formed Input. In *Proceedings of The 4th International Conference on Applied Natural Language Processing*, 1994.

[Lee, 1994] H. Lee. Recognition of Korean-English Bilingual Idioms using Idiom Dispersion Characteristics. Ph.D. Diss., Seoul National University (Korean), 1994.

[Marcus *et al.*, 1993] M. Marcus, B. Santorini, and M. Marcinkiewicz. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics* 19, 1993.

[Yoon, 1994] S. Yoon. Efficient Parser to Find Bilingual Idiomatic Expressions for English-Korean Machine Translation. In *Proceedings of International Conference on Computer Processing of Oriental Languages*, 1994.

[박성숙, 1993] 박성숙 외 5인. 이진 결합 중심의 한국어 Chart Parser. In *Proceedings of Hangul and Korean Information Processing Conference* (Korean), 1993.

[정한민, 1997] 정한민 외 2인. 異質 노드를 가진 트라이 상에서의 복합 단위 검색. In *Proceedings of The 24th KISS Spring Conference* (Korean), 1997.

[조혁규, 1990] 조혁규 & 권혁철. 단일화와 차트를 이용한 한국어 구문 분석 시스템의 구현. *Journal of the KISS* 17, No. 4 (Korean), 1990.

부록

부록 1: 복합 단위 및 가변 요소에 대한 링크를 포함하는 파스 트리의 한 노드 및 해당 가변 요소 노드

```
(VERB ((CU T)(CU_KEY PAY)
(LEX PAID-ON-REAL-OLD-DEBT)(DICTYPE 1)
(SEM_MK2 CA)(TOKEN_NO 5)(FORM PAST EN)
(ROOT PAY-ON-#1-DEBT)(CAT VERB)(UID 3 2)
(LDOCE_TYPE D1 T1 V3 I0 I3)(V_TYPE I0 I3 T1)
(SITU ACTIV)(VOICE A B)(S_CASE1 SUBJ)
(S_CASE2 COMP DOBJ)(D_CASE1 AGT)
(D_CASE2 MGO REC)(C_LEX2 FOR)(S_FORM2 1 3)
(SEM_MK1 CA CAH)(OBLIG1 1)
(LEX PAID-ON-REAL-OLD_DEBT)
(#1 ../EKMT_RESULTS/#1_PARSE)
(CU_K_LEX1 #1 빚을 갚)))

(AP ((g_level 0)(A_TYPE B E))
 (ADP ((g_level 0)(SEM_CAT MODAL_APPROX)(SMHEAD
REAL)(SUBCAT ADJT))
 (ADV_ ((g_level 0)(SEM_CAT MODAL_APPROX)(SUBCAT
ADJT)(ROOT REAL)(LEX REAL))
  (ADV ((LEX REAL)(DICTYPE 1)(TOKEN_NO 7)(ROOT
REAL)(CAT ADV)(UID 1)(SUBCAT ADJT)(POSITION
```

POST)(SEM_CAT MODAL_APPROX)))))
(AP ((g_level 0)(A_TYPE B E))
(ADJ_ ((g_level 0)(A_TYPE B E))
(ADJ ((LEX OLD)(DICTYPE 1)(S_FORM1 1)(TOKEN_NO 8)(ROOT OLD)(CAT ADJ)(UID 2 1)(A_TYPE B E)(SEM_CAT CENT_AGETE)(THE_ADJ PEOP)(S_CASE1 SUBJ)(D_CASE1 CHD)(SEM_MK1 CAS CI CAH)(OBLIG1 1))))))

부록 2: 복합 단위 정보를 이용하지 않는 경우와 하는 경우에의 파싱 결과의 변화 예

[1]
...Level 0 ***ia_no: 481 a_no: 4450 Parse Trees: 14***

[2]
...Level 0 ***ia_no: 245 a_no: 2213 Parse Trees: 7***

Partial Parsing for CU ...
...Level 0 ia_no: 11 a_no: 79
...Level 1 ia_no: 11 a_no: 141
...Level 2 ia_no: 11 a_no: 142
...Level 3 ia_no: 11 a_no: 142