한국시뮬레이션학회 논문지
Vol. 19, No. 1, pp. 103-111 (2010. 3)

# 다단계 구단위화를 이용한 고속 한국어 의존구조 분석

오진영[1] · 차정원[1†]

# High Speed Korean Dependency Analysis Using Cascaded Chunking

Jin Young Oh · Jeong Won Cha

**ABSTRACT**

Syntactic analysis is an important step in natural language processing. However, we cannot use the syntactic analyzer in Korean for low performance and without robustness. We propose new robust, high speed and high performance Korean syntactic analyzer using CRFs. We treat a parsing problem as a labeling problem. We use a cascaded chunking for Korean parsing. We label syntactic information to each Eojeol at each step using CRFs. CRFs use part-of-speech tag and Eojeol syntactic tag features. Our experimental results using 10-fold cross validation show significant improvement in the robustness, speed and performance of long Korea sentences.

**Key words** : Korean Dependency analysis, Cascaded chunking, CRFs(Conditional Random Fields)

**요 약**

한국어 처리에서 구문분석기에 대한 요구는 많은 반면 성능의 한계와 강건함의 부족으로 인해 채택되지 못하는 것이 현실이다. 본 연구는 구문분석을 레이블링 문제로 전환하여 성능, 속도, 강건함을 모두 실현한 시스템에 대해서 설명한다. 우리는 다단계 구 단위화(Cascaded Chunking)를 통해 한국어 구문분석을 시도한다. 각 단계에서는 어절별 품사 태그와 어절 구문표지를 자질로 사용하고 CRFs(Conditional Random Fields)를 이용하여 최적의 결과를 얻는다. 58,175문장 세종 구문 코퍼스로 10-fold Cross Validation(평균 10.97어절)으로 실험한 결과 평균 86.01%의 구문 정확도를 보였다. 이 결과는 기존에 제안되었던 구문분석기와 대등하거나 우수한 성능이며 기존 구문분석기가 처리하지 못하는 장문도 처리 가능하다.

**주요어** : 한국어 의존구조 분석, 다단계 구단위화, CRFs

## 1. 서 론

구문분석은 문장안에서 문장성분들의 관계를 찾는 과정이다. 품사태깅과 더불어 구문분석은 자연어처리 분야에서 필수 단계 중 하나이다. 응용 프로그램에서 언어 분석에 대한 요구가 증가하면 할수록 구문분석에 대한 요구는 증가한다. 예를 들어 기계번역에서 처음에는 단순히 단어들의 정렬을 통해서 번역을 시도하였으나 그 성능이 만족스럽지 못하여 구문분석 정보를 이용하여 문장에서 각 문장 성분들 간의 관계 정보를 이용하여 번역을 시도하고 있다. 또한 인터넷 문서에서 유용한 정보를 추출할 경우에도 단순히 문자열 패턴을 이용하는 것이 아니라 구문분석 결과를 이용하여 좀 더 정확한 추출을 할 수 있다.

이 경우에 반드시 필요한 것이 처리 속도, 안정성, 그리고 성능이다. 품사태깅과는 달리 구문분석은 각 문장 성분들 간의 관계를 조사하기 위해서 CYK알고리즘을 사용하면 $O(n^3)$의 처리 시간이 소요된다. 문장이 길수록 속도는 더욱 느려지게 된다. 구문분석은 문장 전체에 대한 분석결과를 출력하므로 문장이 복잡해질수록 완전한 분석결과를 출력하는 것이 힘들어진다. 더욱이 인터넷 문서와 같이 비문이 많은 문장들에서는 분석을 성공하지 못하고 시스템이 죽어버리는 경우도 많이 발생한다. 성능은 모든 시스템에서 중요하지만 특히 구문분석에서는 구문분석의 오류가 응용 프로그램에 직접 영향을 미치기 때문에 더욱 중요하다.

영어권에서는 구문분석 연구가 성숙하여 구문분석 프

* 본 연구는 2009~2010년도 창원대학교 연구비에 의하여 수행하였음.
2009년 11월 4일 접수, 2010년 1월 12일 채택
[1] 창원대학교 컴퓨터공학과
주 저 자 : 오진영
교신저자 : 차정원
E-mail; jcha@changwon.ac.kr

로그램을 사용하여 다양한 연구 결과를 보이고 있다. 그러나 한국어의 경우는 다양한 연구에서 사용할 수 있는 고속, 고성능의 구문분석기가 존재하지 않는다. 따라서 영어권에 비해서 사용할 수 있는 자원이 줄어들어 연구의 깊이와 결과가 좋지 못한 상황이다.

본 연구에서는 정문과 인터넷 문서와 같이 비문이 많은 환경에서도 고속, 고성능의 결과를 보이는 구문분석기를 제안한다. 본 연구에서는 일반적인 구문분석에서 사용하는 구성 성분들 간의 결합을 통한 구문분석을 하지 않아 사용하는 메모리가 작고 속도가 매우 빠른 장점을 가지고 있다. 또한 어절 구문태그와 CRFs(Conditional Random Fields)를 사용하여 고성능의 한국어 구문분석기를 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 영어권, 일본어, 한국어에서 개발된 구문분석이 대해서 시스템들의 특징과 성능을 알아본다. 3장에서는 제안 시스템의 구조도와 특징에 대해서 기술한다. 4장에서는 다양한 실험을 통해서 시스템을 평가하고 분석하며 5장에서는 결론을 내린다.

## 2. 관련연구

영어권에서는 오래 전부터 많은 연구가 진행되었다. 최근의 연구는 CFG(Context Free Grammar)를 사용하고 통계 모델을 이용한 방법과 기계학습을 이용한 방법이 주류를 이루고 있다(Charniak, 1997, 2000; Dan Klein 등, 2003; Slav Petrov 등, 2007). 또한 재순위화(reranking)를 통해 성능 향상시킨 방법도 제안(Eugene Charniak 등, 2005)되었으며, 영어권에서 개발된 많은 방법들이 일본어와 한국어에 적용되었다.

어순이 비교적 자유로운 일본어에서도 의존 구조를 이용하는 방법이 많이 제안되었다. 의존 구조의 애매성을 해소하기 위해 통계적 방법 과 기계학습을 이용한 다양한 방법이 제안되었다. 예를 들어 최대 우도 추정(Maximum Likelihood Estimation)(Masakazu Fujio 등, 1998), 결정 트리(Decision Tree)(Msahiko Haruno 등, 1999), 최대 엔트로피 모델(Maximum Entropy Model)(Kiyotaka Uchimoto 등, 1999, 2000), 지지 기반 기계(Support Vector Machine)(Taku Kudo 등, 2000) 등이다.

한국어에서도 다양한 시도가 있었다. 한국어 구문분석은 단일화 문법(Unification Grammar), 핵심어 중심 구구조 문법(HPSG: Head-Driven Phrase Structure Grammar), 어휘 기능 문법(LFG: Lexical Functional Grammar), 결합 범주 문법(CCG: Combinatorial Categorial Grammar)을 이용한 시스템들이 제안되었다(Geum 등, 1998; Jung 등, 1989; Yang 등 1990; Yoon 등, 1989; Jeongwon Cha 등, 2002). 최근에는 거의 모든 연구가 의존 문법을 기반으로 하고 있다. 또한 일본어와 마찬가지로 의존 구조의 애매성을 해소하기 위해 다양한 통계 방법과 기계학습을 이용하는 방법들이 제안되었다(Hoojung Chung, 2004; Yong-Hun Lee, 2008). 초기의 한국어에 대한 연구는 학습 코퍼스의 부족으로 연구실 수준의 연구에 머물렀지만 최근에는 한국어정보베이스(Korean Language Information Base), 세종 구문 코퍼스 등이 제작되면서 대용량 코퍼스를 이용하는 연구가 활기를 띠고 있다(Hoojung Chung, 2004; Yong-Hun Lee, 2008).

본 연구에서는 세종 구문 코퍼스를 사용하여 학습 및 평가를 하며 기존에 방법보다 간단하지만 효율적인 한국어 구문분석 방법을 제안한다. 제안된 시스템은 다단계 구단위화를 통해 각 단계에서 지배소를 결정하는 방법을 취한다.

## 3. 제안 시스템

이 장에서는 문장과 의존 구조를 정의하고 제안된 시스템에서 사용하는 자질과 구문분석 방법을 설명한다. 본 연구에서는 구문분석의 기본 단위를 어절 단위로 한다. 따라서 문장은 어절들의 집합이 된다.

즉 문장은 $S=< s_1, s_2, ... s_m >$, 의존구조는 $D=< dep(1), dep(2), ..., dep(m-1) >$와 같이 표시한다. 여기서 $dep(i)=j$는 어절 $s_i$가 어절 $s_j$를 지배소로 갖는다는 것을 의미한다. 이러한 구조에서 기존의 연구에서와 같이 다음과 같은 세 가지의 제한을 $D$가 만족한다고 가정한다.

1. 한국어는 지배소 후위 언어이다.
   즉, 지배소는 피지배소보다 항상 뒤에 위치한다.
2. 교차 의존 구조는 없다.
3. 각 어절의 머리는 유일하다.

본 연구에서 제안하는 시스템은 입력 문장을 품사 태깅한다. 그 결과를 입력으로 받아서 구문태그를 부착하고, 다단계 구단위화를 통한 구문 분석을 한다. 구문태그와 다단계 구단위화는 학습을 위해 다른 자질집합을 사용하여 모델을 생성한다. 그림 1은 제안 시스템의 구조도이다.

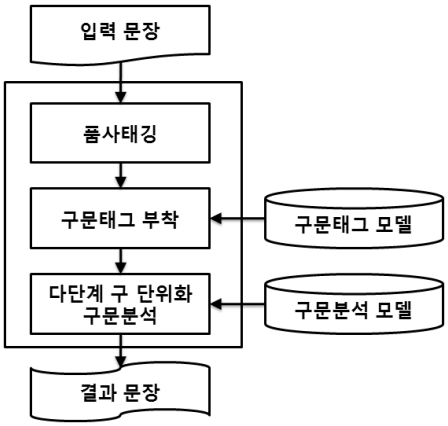


**그림 1.** 제안 시스템 구조도

### 3.1 다단계 구단위화 방법

다단계 구단위화(Cascaded Chunking) 방법은 Steven Abney(1991)에서 영어를 위해 처음 제안되었다. 이 방법은 Kudo 등(2002)과 Zhou 등(2007)에 의해 일본어에 적용되어 좋은 성능을 보였다.

본 연구에서는 한국어의 특성에 맞게 이를 변형하여 적용한다. 그림 2는 다단계 구단위화의 과정을 보여준다. 그림 2에서 'D'는 의존소에 대한 표시이다. 각 단계에 'D' 표지를 부착할 수 있는 어절은 바로 다음 어절이 지배소일 경우이다. 1단계에서 '충실치', '생겨날', '수'에 'D' 표지가 부착되었다. 그렇지 않은 어절에는 '-' 표지를 부착한다. 그리고 다음단계로 넘어갈 때 앞의 어절 표지가 '-'이고 현재 어절의 표지가 'D'인 경우 삭제된다. 그림 2에서 1단계 '수' 어절이 삭제되지 않은 이유는 '생겨날' 어절이 의존소 표지 'D'를 가지고 있기 때문이다. 예를 들어 두 번째 단계에서는 '충실치', '생겨날' 어절이 삭제되어 '자기에 못하고는 도덕이 수 없다.'의 문장에 대해 구문분석을 다시 수행한다. 이런 과정은 한 어절이 남을 때까지 반복한다.

일반적인 구문분석 방법의 시간 복잡도가 $O(n^3)$임에 비하여 구단위화 기법은 $O(n^2)$이므로 매우 빠르다. 실제 본 연구에서 제안한 시스템은 평균 10어절의 문장을 초당 100문장 이상 분석할 수 있다. 또한 구문 요소의 결합이 아니라 레이블링 문제이므로 입력 문장에 대해서 매우 강건하다.

### 3.2 CRFs 학습 및 평가

CRFs는 조건부 확률을 최대로 하는 방향성이 없는 그래프 모델이다(J. Lafferty 등, 2001). 입력열 $X=x_1x_2\cdots x_n$, 상태열 $T=t_1t_2\cdots t_n$이 주어지고 가중치 $\Lambda=\{\lambda,...\}$가 주어졌을 때, CRFs에서는 조건 확률로 식(1)과 같이 정의된다.

| 단계 \ 문장 | 자기에 | 충실치 | 못하고는 | 도덕이 | 생겨날 | 수 | 없다 |
|---|---|---|---|---|---|---|---|
| 초기 | - | - | - | - | - | - | - |
| 1단계 | 자기에 | 충실치 | 못하고는 | 도덕이 | 생겨날 | 수 | 없다 |
| | - | D<br>삭제 | - | - | D<br>삭제 | D | - |
| 2단계 | 자기에 | | 못하고는 | 도덕이 | | 수 | 없다 |
| | D<br>삭제 | | - | - | | D<br>삭제 | - |
| 3단계 | | | 못하고는 | 도덕이 | | | 없다 |
| | | | - | D<br>삭제 | | | - |
| 4단계 | | | 못하고는 | | | | 없다 |
| | | | D<br>삭제 | | | | - |
| 5단계 | | | | | | | 없다<br>-<br>종료 |

**그림 2.** 다단계 구단위화를 이용한 구문분석의 예

$$P(T|X) = \frac{1}{Z(X)} exp\left(\sum_{i=1}^{n}\sum_{k}\lambda_k f_k(t_{i-1}, t_i, x, i)\right) \quad (1)$$

여기서 $Z(X)$는 확률 값으로 만들어 주는 정규화 값이고 $f_k(t_{i-1},t_i,x,i)$는 자질 함수이다. 또한 $\lambda_k$는 각 자질에 대한 가중치를 나타낸다. $k$는 $k$번째 자질이며, 자질 함수는 현재 시간 $i$에 대해 관측열 $x$, 상태변이 $t_{i-1}\rightarrow t_i$에 대해서 전이의 양상을 측정할 수 있다. 매개변수들은 주어진 입력 열과 이에 대응하는 상태 열에 대한 조건부 확률이 최대화하는 최대 우도(maximum likelihood)에 의해서 추정된다. 훈련 집합 $\{(t_l,x_l)\}_{l=1}^{N}$에 대해서 다음과 같은 로그 유사도(log-likelihood)를 계산한다.

식 (2)를 최대로 하도록 학습한다.

$$\begin{aligned} L(\Lambda) &= \sum_{l}\log P_{\Lambda}(t_l|x_l) \\ &= \sum_{l}\left(\sum_{i=1}^{n}\sum_{k}\lambda_k f_k(t_{i-1},t_i,x,i) - \log Z_{x_i}\right) \end{aligned} \quad (2)$$

일반적으로 CRFs는 IIS(Improved Iterative Scaling)나 GIS(Generalized Iterative Scaling)(S. Della Pietra 등, 1997)를 사용하여 학습한다.

또한 학습 데이터의 과적합(overfitting) 문제를 해결하기 위해서 가우스 사전 평활(Gaussian prior smoothing)(A. L. Berger 등, 1996)을 적용한다. 본 연구에서 CRF++[1]을 사용하였다.

### 3.3 자질 집합

본 연구에서는 세종 구문 코퍼스를 사용하였다. 구문분석은 형태소 단위가 아닌 띄어쓰기로 구분된 어절 단위로서 분석하게 된다. 따라서 구문태그와 의존관계에 대해서 기계 학습한 두 모델 결과에 따라 분석을 하게 되므로 각각에 맞는 자질집합 생성이 중요하다.

구문태그와 의존관계를 위해 사용한 자질은 서로 비슷하게 구성되어 있다. 자질의 기본은 형태소 분석기의 결과로서, 문장을 이루는 어절들의 모든 형태소 중에서 각 단계의 모델 생성에 있어 많은 영향을 주는 것을 선택하였다. 그림 3은 구문태그에 대한 자질 예이다.

구문태그는 5개의 자질을 사용하였으며, 자질들을 생성함에 있어서 기호에 대한 품사는 추가하지 않았다.

1은 현재 어절의 첫 번째 형태소의 품사 자질이다. 3은 마지막 형태소 앞의 품사이며, 2는 1과 3의 자질 사이에 'XSA, XSV, VCP' 중 하나의 품사가 있을 경우에 추가된다. 이것은 문장 분석에 있어 용언에 대한 영향을 고려하였을 때, 어절을 이루는 품사가 많을 경우 위의 자질이 추가되지 않는 것을 막기 위해 2번째 자질로 추가하였다. 4는 마지막 품사에 대한 자질이다. 여기에서 4번 자질은 어절을 이루는 형태소가 단일일 경우 1번 자질이 4번에도 추가된다. 어절태그를 예측하는 모델 결과를 분석해보았을 때, 어미 또는 조사가 없는 어절에 대한 오류가 가장 많았다. 이를 위해 4번 자질에 반복 추가함으로써, 오류를 줄이고자 하였다.

그리고 5번째 자질은 다음어절 첫 번째 형태소에 대한 형태소와 품사로서 이루어진 자질이다. 다음 어절의 첫 는 것으로서, 조용사(XSA, XSV)가 붙어서 형용사, 또는 동사가 되는 경우에도 5번 자질에 추가하였다. 예를 들어 다음 어절이 '입장/NNG+하/XSV+ㄴ다/EF+./SF'일 경우 이전 어절의 5번 자질에는 '입장하/VV'가 추가된다.

의존관계를 위한 코퍼스의 자질은 4개의 자질을 사용하는데, 그림 3의 3～5번 자질과 구문태그 결과를 자질로 사용한다. 구문태그 결과만을 자질로 사용하였을 경우 구문태그 자질의 오류에 대해서 구문분석 예측 확률이 낮아서 품사에 대한 자질을 추가하였다.

| 형태소 분석 | 1 | 2 | 3 | 4 | 5 | 구문태그 |
|---|---|---|---|---|---|---|
| 물론/MAG | MAG | - | - | MAG | - | AP |
| 스포츠/NNG+에/JKB | NNG | - | - | JKB | 있/VV | NP_AJT |
| 있/VV+어서/EC+도/JX | VV | - | EC | JX | - | VP |
| 이것/NP+은/JX | NP | - | - | JX | - | NP_SBJ |
| 예외/NNG+가/JKC | NNG | - | - | JKC | 아니/VCN | NP_CMP |
| 아니/VCN+다/EF+./SF | VCN | - | - | - | - | VP |

**그림 3.** 구문태그 코퍼스의 예

1) http://crfpp.sourceforge.net/

## 4. 실험 및 분석

### 4.1 실험환경

세종 코퍼스 58,175문장을 10-fold cross validation을 수행하였다. 한 문장의 평균 길이는 10.97이며, 각 실험은 1,000문장으로 평가하였다.

제안한 시스템의 성능 평가를 위해 아크-정확도와 아크-재현율을 결합한 $F_1-measure$와 문장 정확도(Exact-Matching)를 사용하였다. 평가 척도는 식 (3)과 같다. 본 연구에서는 구문 분석을 레이블링 문제로 해결하기 때문에 아크-정확도와 아크-재현율이 같다.

$$
\begin{aligned}
&\text{아크}-\text{정확도}(Arc\,\mathrm{Precision}, AP)\\
&=\frac{\text{구문 분석 파스트리에서 올바른 아크의 수}}{\text{구문 분석 파스트리에서 모든 아크의 수}},\\
&\text{아크}-\text{재현율}(Arc\,Recall, AR)\\
&=\frac{\text{구문 분석 파스트리에서 올바른 아크의 수}}{\text{정답 파스트리에서 모든 아크의 수}},\\
&F_1-measure=\frac{2\times AP\times AR}{AP+AR},\\
&Exact-Matching=\frac{\text{정확히 분석된 문장의 수}}{\text{문장의 수}}
\end{aligned}
\tag{3}
$$

### 4.2 자질 선택 실험

구문분석을 위한 학습 문서를 생성할 때, 자질의 선택은 중요하다. 특히 5번째 자질은 다단계 구단위화 방법에 따라 동적으로 생성되는 것이고, 서술어에 대한 구문분석의 영향이 크므로 5번째 자질 선택에 대해 실험을 통하여 유효성을 알아보았다.

표 1과 표 2는 5번째 자질 선택에 따른 성능이다.

표 1은 그림 3와 같은 방법으로 자질을 생성하였으며, 표 2는 다음 어절이 아닌 지배소 어절에 대한 자질을 생성한 결과이다.

각 성능은 10-fold cross validation을 수행하였으며 형태소분석 및 구문태그의 결과가 정답인 문서에 대해서 구문분석의 성능측정(P)하였다.

표 1과 표 2의 실험 결과로 보아 다음 어절에 대하여 5번째 자질을 생성할 때 성능이 더 높았다. 표 2는 평가 시에는 다단계 구단위화 방법에 의해 다음 어절을 지배소로 가지는 결과만 알 수 있기 때문에 학습문서와의 차이로 발생하는 오류가 많았다.

### 4.3 구문 분석 성능

본 논문에서는 구문분석 성능을 위해 세 가지의 큰 분류로 실험을 나누었다. 형태소분석 및 구문태그의 결과가 정답인 문서에 대해서 구문분석의 성능측정(P)과 형태소분석만 정답인 문서에 대한 성능측정(E+P), 형태소분석부터 구문태그 모두 시스템의 결과로서 측정한 성능평가(T+E+P) 실험으로 나누어진다. 본 논문에서는 홍진표 등(2008)의 품사 태거를 사용하였다.

또한 각 실험에 대해서 20어절 이상이 포함된 장문 문장과 20어절 미만인 문장으로 분리하여 따로 성능을 측정하였다. 먼저 표 3은 P의 성능이다.

단, 표 1의 오류를 분석하여 콤마(,)자질을 추가하였다. 콤마(,)의 경우는 대등연결 기능(나열 기능)과 종속절 표

**표 1.** 다음 어절로 5번째 자질 생성

| | 어절 | 문장 |
|---|---|---|
| 1 | 0.8151 | 0.3000 |
| 2 | 0.8743 | 0.5690 |
| 3 | 0.8731 | 0.5560 |
| 4 | 0.8800 | 0.5480 |
| 5 | 0.8517 | 0.4470 |
| 6 | 0.8433 | 0.4570 |
| 7 | 0.8410 | 0.4470 |
| 8 | 0.8352 | 0.3510 |
| 9 | 0.8394 | 0.3240 |
| 10 | 0.8446 | 0.3380 |
| 평균 | 0.8497 | 0.4337 |

**표 2.** 지배소 어절로 5번째 자질 생성

| | 어절 | 문장 |
|---|---|---|
| 1 | 0.7751 | 0.2460 |
| 2 | 0.8326 | 0.4920 |
| 3 | 0.8360 | 0.5110 |
| 4 | 0.8321 | 0.4660 |
| 5 | 0.8139 | 0.3810 |
| 6 | 0.7985 | 0.3780 |
| 7 | 0.8027 | 0.3840 |
| 8 | 0.7881 | 0.2880 |
| 9 | 0.7935 | 0.2590 |
| 10 | 0.8003 | 0.2700 |
| 평균 | 0.8073 | 0.3675 |

현 등으로 다양하게 사용하는데 표 1의 결과에서는 콤마에 따라 지배소와 피지배소에 대한 오류가 많이 발생하였다. 자질 생성의 예로 그림 3에서 '물론/MAG' 어절이 '물론/MAG+,/SP'일 경우 1번 자질은 '-', 2번 자질은 'MAG', 3번 자질은 'SP', 4번 자질은 'AF'이 된다.

표 3의 실험 결과를 표 1과 비교해보면 콤마가 문장 구조에 영향을 주는 것을 알 수 있다. 따라서 문장 구조에 영향을 줄 수 있는 기호, 조사 등에 대한 추가적인 자질 연구가 필요하다. 표 4는 표 3에 대한 E+P 성능이고, 표 5은 T+E+P의 실험 결과이다. 이 실험 또한 10-fold cross validation으로 실험하였다.

표 5 T+E+P에 관한 어절 오류를 분석을 해보았을 때, 품사 태깅 오류에 의해 구문분석 오류가 발생한다는 것을 알 수 있었다. 이것은 품사태깅 결과를 자질로 사용하기 때문에 당연한 결과이다. 따라서 품사 태깅 성능을 향상시키는 것이 당면 과제이다.

**표 3.** P 구문분석의 성능. '<20'은 20어절 미만의 성능을 나타내고 '20>'은 20어절 이상의 성능을 나타낸다. ()는 문장의 성능을 나타낸다.

| | <20 | >20 | 전체 |
|---|---|---|---|
| 1 | 0.8406<br>(0.3679) | 0.7723<br>(0) | 0.8185<br>(0.3090) |
| 2 | 0.8835<br>(0.5818) | 0.7875<br>(0) | 0.8801<br>(0.5760) |
| 3 | 0.8826<br>(0.5794) | 0.7932<br>(0) | 0.8791<br>(0.5730) |
| 4 | 0.8834<br>(0.5538) | 0.7965<br>(0) | 0.8821<br>(0.5510) |
| 5 | 0.8600<br>(0.4613) | 0.7827<br>(0) | 0.8534<br>(0.4470) |
| 6 | 0.8507<br>(0.4677) | 0.7531<br>(0) | 0.8477<br>(0.4630) |
| 7 | 0.8516<br>(0.4768) | 0.7458<br>(0) | 0.8419<br>(0.4620) |
| 8 | 0.8466<br>(0.3776) | 0.8048<br>(0) | 0.8389<br>(0.3500) |
| 9 | 0.8469<br>(0.3401) | 0.7876<br>(0) | 0.8384<br>(0.3180) |
| 10 | 0.8601<br>(0.3578) | 0.7726<br>(0.0164) | 0.8479<br>(0.3370) |
| 평균 | 0.8606<br>(0.4564) | 0.7796<br>(0.0016) | 0.8528<br>(0.4386) |

### 4.4 오류 분석 및 대처

표 3의 P성능에 대하여 오류를 분석해 보면, 먼저 그림 4와 같은 보조용언에 대한 오류가 많이 나타났다.

여기서 '인사/NNG+하/XSV+며/EC' 어절은 '거/NNB+이/VCP+야/EF+?/SF'를 지배소로 가져야 하는데 '기다리/VV+고/EC'를 지배소로 가졌다. 세종 코퍼스의 특성상 보조용언이 있는 구조는 가장 마지막 용언이 지배소가 되기 때문에 이러한 현상이 발생한 것으로 보인다.

두 번째는 조사가 없는 명사 연속 어절에 대한 오류가 많았다.

그림 5와 같이 명사가 연속되어 나오는 경우 수식하는 순서가 문맥에 따라서 많이 달라진다. 앞에 오는 수식어구가 첫 명사를 수식하기도 하고 마지막 명사를 수식하는 경우도 있다.

세 번째 또한 세종 코퍼스의 특성상 발생하는 오류이다. 그림 6과 같이 '수도원장/NNG+을/JKO' 어절은 '것/NNB+이/VCP+었/EP+다/EF+./SF'를 지배소로 가지는데 '임명/NNG+하/XSV+는/ETM' 어절을 지배소로 가져 생기는 오류가 많았다.

**표 4.** E+P 구문분석의 성능

| | <20 | >20 | 전체 |
|---|---|---|---|
| 1 | 0.8174<br>(0.3264) | 0.7468<br>(0) | 0.7946<br>(0.2740) |
| 2 | 0.8734<br>(0.5612) | 0.7833<br>(0) | 0.8702<br>(0.5560) |
| 3 | 0.8676<br>(0.5248) | 0.7782<br>(0) | 0.8641<br>(0.5190) |
| 4 | 0.8701<br>(0.5176) | 0.7699<br>(0) | 0.8686<br>(0.5150) |
| 5 | 0.8520<br>(0.4345) | 0.7648<br>(0) | 0.8445<br>(0.4210) |
| 6 | 0.8462<br>(0.4434) | 0.7531<br>(0) | 0.8433<br>(0.4390) |
| 7 | 0.8359<br>(0.4293) | 0.7342<br>(0) | 0.8266<br>(0.4160) |
| 8 | 0.8325<br>(0.3484) | 0.7920<br>(0) | 0.8251<br>(0.3230) |
| 9 | 0.8342<br>(0.3155) | 0.7506<br>(0) | 0.8221<br>(0.2950) |
| 10 | 0.8457<br>(0.3397) | 0.7518<br>(0.0164) | 0.8327<br>(0.3200) |
| 평균 | 0.8475<br>(0.4241) | 0.7625<br>(0.0016) | 0.8392<br>(0.4080) |

**표 5.** T+E+P 구문분석의 성능

| | <20 | >20 | 전체 |
|---|---|---|---|
| 1 | 0.7856<br>(0.2762) | 0.7166<br>(0) | 0.7633<br>(0.2320) |
| 2 | 0.8424<br>(0.5101) | 0.7542<br>(0) | 0.8392<br>(0.5050) |
| 3 | 0.8466<br>(0.4975) | 0.7331<br>(0) | 0.8421<br>(0.4920) |
| 4 | 0.8424<br>(0.4643) | 0.7611<br>(0) | 0.8412<br>(0.4620) |
| 5 | 0.8183<br>(0.3767) | 0.6905<br>(0) | 0.8073<br>(0.3650) |
| 6 | 0.8031<br>(0.3768) | 0.6862<br>(0) | 0.7995<br>(0.3730) |
| 7 | 0.7954<br>(0.3467) | 0.7135<br>(0) | 0.7880<br>(0.3360) |
| 8 | 0.7869<br>(0.2762) | 0.7240<br>(0) | 0.7754<br>(0.2560) |
| 9 | 0.8099<br>(0.2952) | 0.7180<br>(0) | 0.7966<br>(0.2760) |
| 10 | 0.8102<br>(0.3035) | 0.7251<br>(0) | 0.7984<br>(0.2850) |
| 평균 | 0.8141<br>(0.3723) | 0.7222<br>(0) | 0.8051<br>(0.3582) |

| | |
|---|---|
| 인사/NNG+하/XSV+며/EC | VP |
| 기다리/VV+고/EC | VP |
| 있/VX+는/ETM | VP_MOD |
| 거/NNB+이/VCP+야/EF+?/SF | VNP |

**그림 4.** 보조용언 오류

### 4.5 언어 자질 추가를 통한 오류 해결

보조용언 오류, 조사가 없는 연속된 명사어절의 오류를 해결하기 위해서 구문분석에서 사용하는 자질을 변경하였다. 그림 3에서 3번은 현재 어절의 마지막 앞의 형태소 자질이다. 3번 자질을 어절의 첫 번째 형태소로 변경함으로써 위의 오류들을 감소시킬 수 있었다.

어절을 이루는 형태소가 1개일 경우 기본 실험에서는 2번 자질에 첫 번째 형태소를 추가하였는데, 자질을 변경할 때에는 3번, 4번 자질에 같은 형태소가 추가된다. 3번 자질과 4번 자질이 동일할 때 동시에 사용할 것인지 하나를 사용할 것인지를 결정하기 위해서 3가지 추가 실험을 하였다.

| | |
|---|---|
| 있/VX+는/ETM | VP_MOD |
| 흰색/NNG | NP |
| 농구/NNG | NP |
| 반바지/NNG+를/JKO | NP_OBJ |

**그림 5.** 조사가 없는 연속된 명사어절 오류

| | |
|---|---|
| 곳/NNG+에서/JKB+는/JX | NP_AJT |
| 수도원장/NNG+을/JKO | NP_OBJ |
| 임명/NNG+하/XSV+는/ETM | VP_MOD |
| 것/NNB+이/VCP+었/EP+다/EF+./SF | VNP |

**그림 6.** 의존 명사가 용언으로 사용되어 수식을 받을 경우

**표 6.** 자질 추가 실험 결과

| | 3번 자질에만 추가 | | 4번 자질에만 추가 | |
|---|---|---|---|---|
| | 어절 | 문장 | 어절 | 문장 |
| 1 | 0.8285 | 0.3030 | 0.7862 | 0.2370 |
| 2 | 0.8877 | 0.6080 | 0.8831 | 0.5910 |
| 3 | 0.8856 | 0.5990 | 0.8785 | 0.5830 |
| 4 | 0.8964 | 0.5870 | 0.8964 | 0.5810 |
| 5 | 0.8610 | 0.4730 | 0.8531 | 0.4510 |
| 6 | 0.8469 | 0.4700 | 0.8464 | 0.4680 |
| 7 | 0.8558 | 0.4670 | 0.8486 | 0.4530 |
| 8 | 0.8463 | 0.3690 | 0.8409 | 0.3580 |
| 9 | 0.8437 | 0.3260 | 0.8491 | 0.3310 |
| 10 | 0.8487 | 0.3380 | 0.8541 | 0.3540 |
| 평균 | 0.8601 | 0.454 | 0.8537 | 0.4407 |

표 6은 3가지 실험 중 3번 자질에만 형태소 자질을 추가한 실험과 4번 자질에만 추가한 실험 결과이고, 표 7은 3번과 4번 자질에 동시에 추가했을 때의 실험 결과이다.

실험 결과를 분석하면 어절의 첫 형태소, 즉 의미 형태소를 추가하는 것이 성능 향상에 도움이 되었다. 이것은 구문 태그가 의미 형태소와 조사에 의해서 결정이 되지만 특별한 경우에는 여전히 의미 형태소가 문장 구조에 영향을 미친다는 것을 보여준다. 예를 들어 '사랑/NNG+하/XSV+는/ETM'는 구문태그는 VP_MOD이지만 수식을 받을 수 있다. 이러한 경우 VP_MOD라는 구문태그만으로는 구조를 결정하는데 부족하다.

**표 7.** 3번 자질과 4번 자질을 동시에 사용한 실험

| | 어절 | 문장 |
|---|---|---|
| 1 | 0.8285 | 0.3030 |
| 2 | 0.8880 | 0.6030 |
| 3 | 0.8837 | 0.5840 |
| 4 | 0.8933 | 0.5900 |
| 5 | 0.8555 | 0.4640 |
| 6 | 0.8509 | 0.4780 |
| 7 | 0.8496 | 0.4490 |
| 8 | 0.8450 | 0.3650 |
| 9 | 0.8454 | 0.3330 |
| 10 | 0.8526 | 0.3540 |
| 평균 | 0.8593 | 0.4523 |

## 5. 결 론

논문에서는 한국어 의존구조 분석을 위해 기계학습을 이용하였다. 어절태그를 사용하여 문장의 요소를 분리 하였으며, 다단계 구단위화(Cascaded Chunking) 방법을 사용하여 강건함을 획득할 수 있었다. 실험결과로 보았을 때, 어휘를 사용하지 않은 자질 학습으로 모델의 사이즈가 작으며, 고속으로 처리할 수 있었다.

또한 이전 논문에서 처리하기 힘든 장문에 대한 분석이 가능하며, 이 또한 분석 처리속도가 빠르게 이루어짐을 확인하였으며, 오류분석을 통한 언어자질을 추가함으로써 기 개발된 시스템의 성능향상을 이룰 수 있었다.

현재 개발된 한국어 구문분석은 KIB(Korean Language Information Base) 코퍼스를 사용하였기 때문에 직접적인 분석을 할 수는 없지만, 85.62~87.46%의 성능을 나타내고 있다. 본 논문은 세종코퍼스를 사용하여 86.01%를 보이고 있다.

또한 이 결과를 근거로 하여 추가적인 자질 연구를 할 예정이며 용언의 하위분류를 추가하여 추가적인 성능향상과 상위 응용에 대응할 수 있는 시스템을 구현할 예정이다.

## 참 고 문 헌


1. 홍진표, 차정원, "어절패턴 사전을 이용한 새로운 한국어 형태소 분석기," 한국정보과학회 학술발표논문집, 35(1(C)), pp. 279-284, 2008년.
2. A.L. Berger, V.J. Della Pietra, and S.A. Della Pietra, "A maximum entropy approach to natural language processing," Computational Linguistics, vol. 22, no. 1, pp. 39-71, 1996.
3. Charniak, E., "Statistical parsing with a context-free grammar and word statistics," In Proceedings of the Fourteenth National Conference on Artificial Intelligence. Menlo Park, AAAI Press/MIT, pp. 598-603, 1997.
4. Charniak, E., "A Maximum-Entropy-Inspired Parse," In Proceedings of NAACL-2000, pp. 132-139, 2000.
5. Dan Klein and Christopher D. Manning., "Accurate Unlexicalized Parsing," ACL 2003, pp. 423-430, 2003.
6. Eugene Charniak and Mark Johnson, "Coarse-to-fine n-best parsing and MaxEnt discriminative reranking," In ACL 2005, pp. 173-180, 2005.
7. Geum, J. C. and G. Kim, "Implementation of HPSG parsig mechanism for Korean syntactic structure analysis," In Proceedings of the Spring Conference of Korea Information Science Society, pp. 139-142, 1998.
8. Hoojung Chung, Statistical Korean Dependency Parsing Model based on the surface Contextual Information, Ph.D. dissertation, 2004.
9. J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," In Proceedings. 18th International Conference on Machine Learning, pp. 282-289, 2001.
10. Jeongwon Cha, Geunbae Lee, and Jong-Hyeok Lee, "Morpho-syntactic categorial modeling of Korean," Computers and the Humanitie Journal, vol 36, no. 4, pp. 431-453, 2002.
11. Jung, H.-S., J.-H. Kim, J.-S. Lee, S.-Y. Chun, and M.-J, "Park Design of Korean-English machine translation system (KoEng)," In Proceedings of the 1st Workshop of Machine Translation, pp. 87-96, 1989.
12. Kiyotaka Uchimoto, Masaki Murata, Satoshi Sekine, and Hitoshi Isahara, "Dependency model using posterior context," In Procedings of Sixth International Workshop on Parsing Technologies, pp. 321-322, 2000.
13. Kiyotaka Uchimoto, Satoshi Sekine, and Hitoshi Isahara, "Japanese Dependency Structure Analysis Based on Maximum Entropy Models," In Proceedings of the EACL, pp. 196-203, 1999.
14. Kudo, T. and Y. Matsumoto, "Japanese Dependency Analysis using cascaded Chunking," In Proceedings of the CoNLL-2003, pp. 63-69, 2002.
15. Masakazu Fujio and Yuji Matsumoto, "Japanese Dependency Structure Analysis based on Lexicalized Statistics," In Proceedings of EMNLP '98, pp. 87-96, 1998.
16. Msahiko Haruno, Satoshi Shirai, and Yoshifumi Ooyama, "Using Decision Trees to Construct a Practical Parser,"

Machine Learning, 34:131-149, 1999.
17. S. Della Pietra, V. Della Pietra, and J. Lafferty, “Inducing features of random fields,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 4, pp. 380-393, 1997.
18. Slav Petrov and Dan Klein, “Improved Inference for Unlexicalized Parsing,” In proceedings of HLT-NAACL 2007, pp. 404-411, 2007.
19. Steven Abney, “Parsing By Chunking,” In Principle-Based Parsing. Kluwer Academic Publishers, 1991.
20. Taku Kudo and Yuji Matsumoto, “Japanese Dependency Structure Analysis based on Support Vector Machines,” In Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 18-25, 2000.
21. Yang, J, A study on the Korean analyzer based on HPSG, Master’s thesis, Dept. of Computer Engineering. Seoul National University, 1990.
22. Yong-Hun Lee and Jong-Hyeok Lee, “Korean Parsing using Machine Learning Techniques,” KCC 2008, pp. 285-288, 2008.
23. Yoon, D. H. and Y. T. Kim, “Analysis techniques for Korean sentence based on Lexical Functional Grammar,” In Proceedings of the International Parsing Workshop ’89, pp. 369-78, 1989.
24. Zhou, H., T. Yu, et al, “Japanese Dependency Analysis Based on SVMs and CRFs,” International Journal of Mathematics and Computersin Simulation, 1(3): 233-237, 2007.

**오 진 영** (psyche.ojy@gmail.com)

2008 창원대학교 컴퓨터공학과 학사
2010 창원대학교 컴퓨터공학과 석사
2010 창원대학교 컴퓨터공학과 박사과정

관심분야 : 구문분석, 개체명 인식, 자연어처리

**차 정 원** (jcha@changwon.ac.kr)

2002 포항공과대학교 공학박사
2003 USC/ISI 박사후 과정
2004～현재 창원대학교 조교수

관심분야 : 기계학습, 자연어처리, 정보검색