온라인 학습을 이용한 한국어 의존구문분석

이용훈 ○ 이종혁

포항공과대학교 전자컴퓨터공학부 컴퓨터공학과

{yhlee95○, jhlee}@postech.ac.kr

Korean Dependency Parsing Using Online Learning

Yong-Hun Lee○ Jong-Hyeok Lee

Department of Computer Science and Engineering

Division of Electrical and Computer Engineering

Pohang University of Science and Technology

요 약

본 논문에서는 온라인 학습을 이용한 한국어 의존구문분석 방법을 제안한다. CoNLL-X에서 1위를 차지한 그래프 기반 의존구문분석 방법을 한국어에 맞게 변형하고, 한국어의 교착어적 특성을 고려해 한국어에 적합한 자질 집합을 제시하였다. 특히 의존트리의 에지(edge)를 단어와 단어간의 의존관계가 아닌 부분트리(partial tree)와 부분트리의 의존관계로 바라보기 위해 부분트리가 공유하고 있는 기능어 정보를 추가 자질로 사용하였다. 또한 한국어의 지배소 후위(head-final) 언어 특성과 투사성(projectivity)을 이용하여 Eisner(1996) 알고리즘을 사용하지 않고도 $O(n^3)$의 CYK알고리즘을 사용할 수 있었고, 이를 이용해 최적의 전역해(global optimum)를 찾을 수 있었다. 각 자질을 위한 최적의 가중치 벡터는 온라인 학습방법 중 하나인 Collins(2002)의 averaged perceptron 알고리즘을 사용함으로써 빠르게 모델을 학습할 수 있었다. 제안 모델을 국어정보베이스(KIBS) 말뭉치에 적용한 결과 어절 단위 정확률 88.42%의 높은 성능을 얻을 수 있었다.

1. 서 론

구문분석은 문장을 구성하는 단어들로부터 문장의 구조를 찾아내는 작업이라 할 수 있다. 문장의 구조는 크게 여러 단어를 더 큰 단위인 구 (phrase)로 묶어 나가는 표현 방식과 단어와 단어 간의 이진 의존관계(dependency)로 표현하는 방식으로 나눌 수 있는데, 이러한 표현방식을 사용한 구문분석을 각각 구구조 구문분석과 의존 구문분석이라 부른다.

영어와 같이 어순이 비교적 고정적인 언어에서는 구구조문법을 통해 문장을 분석해 나가는 구구조 구문분석이 대부분이었던 반면 한국어와 같이 부분 자유 어순을 가지거나 문장성분이 생략 가능한 언어에서는 의존 구문분석이 선호되어 왔다. 하지만, 구구조문법만으로 모든 구문 애매성을 해결하는 데 한계를 느낀 연구자들이 단어 자체의 정보나 단어와 단어간의 연관관계를 사용하여 구문분석 성능을 높이게 되면서 영어권에서도 점차 의존구문분석에 대한 관심이 생겨나기 시작하였다[1,2]. 그러다가 2006년도에 CoNLL이 shared task로 다국어 의존구문분석을 선택하면서 이러한 관심은 폭발적으로 증가하게 되었다[3].

최근의 의존구문분석 연구들을 살펴보면 크게 두 개의 큰 연구결과에 그 뿌리를 두고 있는데 Nivre (2003)[4]의 트랜지션 기반 의존구문분석 모델 (transition-based dependency parsing)과 McDonald (2005)[5]의 그래프 기반 의존구문분석 모델 (graph-based dependency parsing)이다.[1] Nivre(2003)[4]가 제안한 시간 복잡도 O(n)의 트랜지션 기반 의존구문분석 모델은 두 개의 대상단어의 의존여부를 결정하고 이를 이용하여 의존트리를 점차적으로 완성해나가는 방법으로서 처음 제안할 당시에는 의존관계 간의 교차가 일어나지 않는 투사적 언어(projective language)에만 적용할 수 있는 방법이었으나, 이후 비투사적 언어(non-projective language)에도 적용할 수 있도록 확장되었다[6, 7]. 이 방법은 두 대상단어와 그 단어의 지역문맥을 이용하여 결정적(deterministic)으로 두 대상단어의 의존 여부를 찾아나가는 지역적 학습 모델(locally training model)로서 부분적으로 완성된 구문분석 결과(parsing history)를 재사용하여 구문분석 성능을 향상시킬 수 있는 장점이 있다. 대표적인 연구에는 Yamada(2003)[8], Nivre(2003, 2005, 2009)[4, 6, 7], Kudo(2000, 2002)[9, 10] 등이 있다. 이에 반해 Mcdonald(2005)[5]가 제안한 그래프 기반 의존구문분석 모델은 구문분석을 모든 가능한 의존트리에서 가장 점수가 높은 의존트리를 찾는

[1] CoNLL-X shared task에서 McDonald의 그래프 기반 의존구문분석 모델이 1위, Nivre의 트랜지션 기반 의존구문분석 모델이 2위를 차지하였다.

MST(maximum spanning tree) 문제로 보고 구문분석을 수행하는 방법이다. 이 방법은 모든 가능한 단어 쌍의 의존가능성을 여러 문맥자질로 표현하고 이를 점수화함으로써 가장 점수가 큰 의존트리를 찾는 전역적 학습 모델(globally training model)이다. 자질의 가중치를 구하기 위해 온라인 학습기법인 MIRA 알고리즘[11]을 사용하였으며, 구문분석을 하기 위해 투사적 언어에서는 시간 복잡도 $O(n^3)$의 Eisner(1996)[12] 알고리즘을, 비투사적 언어에서는 시간 복잡도 $O(n^2)$의 CLE 알고리즘을 사용하였다. 이 방법은 추후 같은 방향의 인접한 형제노드(sibling) 정보 한 개를 추가로 이용하는 2차 MST 알고리즘으로 확장되었다[13].

지역적, 전역적 모델의 차이로 인해 트랜지션 기반 의존구문분석이 풍부한 문맥자질 사용이 가능하다는 점과 근거리 의존관계를 찾는데 강한 반면, 그래프 기반 의존구문분석은 장거리 의존관계와 문장의 헤드인 루트(root)를 찾는데 강점을 가진다[14]. 최근에는 이러한 두 모델의 장점을 살려 분류기 누적(classifier stacking) 방식의 모델이나 트랜지션 기반 의존구문분석의 지역 최적해(local optimum) 문제를 해결하기 위해 전역적 자질과 빔탐색을 이용하여 전역적 해를 찾는 연구가 진행되기도 하였다[14, 15].

한국어에 대한 대표적인 통계적 의존구문분석 방법에는 원시 말뭉치로부터 추출한 공기정보를 이용하는 방법[16]이나, 어휘 간의 의존 확률과 의존소와 지배소 간의 수식거리를 이용하는 연구 등 주로 말뭉치에서 추출한 확률을 이용하는 방법[17, 18]이 대부분이었다. 일반적으로 의존구문분석에서 어휘 의존 확률만을 사용하게 되면 비슷한 어휘 연관성을 가지는 경우 중의성을 해결하지 못하므로 수식거리를 추가적으로 사용하는데, Chung(2004)[19]은 수식거리를 자질로 사용할 때 발생하는 자료부족문제(data sparseness problem)를 해결하기 위해 의존소의 표층 문맥 정보를 사용하는 통계 모델을, 우연문(2007)[20]은 단순한 표층 문맥이 아닌 지배가능 경로 문맥을 사용한 수식거리를 사용하여 원거리 의존관계의 오류율을 낮추는 방법을 제안하였다. 최근에는 트랜지션 기반 의존구문분석 방법인 SVM을 이용한 Shift-Reduce방식의 구문분석 알고리즘[21, 22]이나 CRF를 이용한 단계적 구단위화 (cascaded chunking) 알고리즘을 이용한 한국어 의존구문분석 방법[23]이 제안되기도 하였다.

본 논문에서는 CoNLL-X에서 1위를 차지한 바 있는 McDonald의 그래프 기반 의존구문분석 방법을 한국어에 적용하는 방법을 제안한다. 특히 단어와 단어간의 의존관계인 에지(edge)의 점수를 계산할 때, 단순히 단어와 단어간의 관계로 보지 않고 부분트리(partial tree)와 부분트리의 관계로 보고 에지의 점수를 계산할 것이다. 이를 위해 두 부분트리가 공유하는 조사나 어미 등의 기능어를 비교하여 자질로 추가하여 사용하였으며, 각 자질의 가중치를 구하기 위해서는 Collins(2002)[24]의 averaged (structured) perceptron 알고리즘을 사용하였다.

2. 온라인 학습을 이용한 한국어 의존구문분석

일반적으로 한국어 의존구문분석에서 고려해야 할 특징으로는 여러 형태소가 결합된 복잡한 어절 형태, 자신의 지배소는 항상 자신의 뒤쪽에 나온다는 지배소 후위(head final) 원칙, 모든 의존관계는 서로 교차하지 않는다는 투사성(projectivity) 등이 있다.

그 중에서 한국어의 교착어적 성격은 기존 연구들에서 일반적으로 사용하고 있는 n-gram방식의 자질을 그대로 사용하지 못하게 만드는 원인이 되어 왔다. 영어나 중국어와 같은 언어들이 비교적 단어에 굴절이 거의 일어나지 않는 반면 한국어는 조사나 어미 등에 의해서 거의 모든 어절들에 굴절이 일어나기 때문에 영어와 같이 n-gram방식의 자질을 그대로 사용하게 될 경우 심각한 자료부족 문제를 야기하게 된다. 따라서 기존의 한국어, 일본어 구문분석 연구들에서는 어절과 분절을 이루는 형태소들 중에 대표내용어와 대표기능어 두개를 어절과 분절의 대표형으로 추상화하여 사용하였다[19, 21, 22].

또한 지배소 후위 원칙과 투사성을 이용하면 Eisner의 구문분석 알고리즘[12]을 사용하지 않더라도 시간복잡도 $O(n^3)$의 CYK 알고리즘을 사용할 수 있게 된다. 이에 대해서는 뒤에서 자세히 살펴보도록 한다.

2.1 구문분석 모델

본 논문에서 제안하는 구문분석 모델은 기본적으로 Eisner(1996)[12]의 edge factered model을 사용한다. 이 모델에서는 입력문장 x에 해당하는 의존구문분석 트리 y의 점수를 트리를 구성하는 모든 에지(edge) 점수 $s(i, j)$의 합으로 정의한다. 각 에지의 점수는 두 대상단어(x_i, x_j)의 문맥자질 $f(i, j)$과 가중치 w의 곱으로 표현된다.

$$S(x,y) = \sum_{(i,j)\in y} s(i,j) = \sum_{(i,j)\in y} w \cdot f(i,j)$$

제안하는 모델이 기존연구와 다른 점은 에지를 바라보는 관점이 단어와 단어간의 의존관계로 보는 것이 아니라 부분트리(partial tree) t_i와 부분트리 t_j의 의존관계로 본다는 점이다.

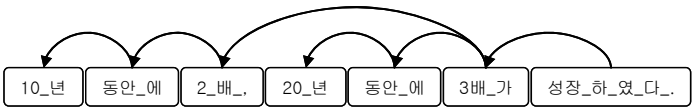

그림 1. 한국어 의존트리 예문

그림 1의 예문 “10년 동안에 2배, 20년 동안에 3배가 성장하였다.”라는 문장에서 “2배,”와 “3배가”라는 두 어절 간의 에지는 실제로는 “10년 동안에 2배,”라는 선행구와 “20년 동안에 3배가”라는 후행구가 대등적으로 나타난 것으로 볼 수 있다. 따라서 본 논문에서는 이러한 부분트리 간의 의존관계를 잡아내기 위한 한 방법으로 특별히 두 부분트리가 공유하고 있는 조사나 어미 등의 기능어 자질을 추가하여 사용하였다. 이러한 관점은 구구조문법에서 구와 구를 합쳐서 더 큰 구를 만들어 나가는 방식과도 닮아 있어 언어학적으로도 합당해 보이며, 구문분석 오류의 상당 부분을 차지하고 있는 대등구나 대등절의 오류를 줄이는데도 기여할 것으로 기대된다.

2.2 구문분석 알고리즘

일반적으로 투사적 언어의 의존구문분석을 위한 CYK 알고리즘의 시간복잡도는 $O(n^5)$으로 알려져 있다[5, 12]. 이는 CYK 삼각테이블을 구성하고 있는 두개의 차트를 합칠 때 두개의 차트가 가지고 있는 정보가 현재까지 처리된 부분트리의 시작 위치, 끝 위치, 헤드의 위치를 저장하고 있기 때문이다.

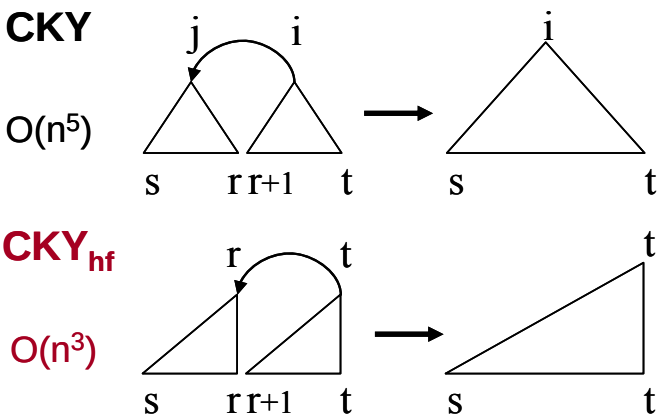

그림 2. CYK 알고리즘 도식화

Input: sentence x, sentence length n
Initialize C[i][i]=0.0 for all 1≤i≤n

for all 1≤s<n:
  for all s<t≤n:
    C[s][t]=
      $\text{argmax}_{s\le r<t}$ C[s][r] + C[r+1][t] + S(r, t)

그림 3. 지배소 후위 언어의 CYK 알고리즘

따라서 그림 2에서 보는 바와 같이 구문분석 시에 5개의 인덱스가 필요하고 $O(n^5)$의 시간복잡도를 가지게 된다. 하지만, 문장이 길어지게 되면 이러한 알고리즘은 다루기 힘든(intractable) 알고리즘이 되고 만다. 이러한 이유 때문에 Eisner(1996)[12]는 각 차트를 헤드를 중심으로 양쪽으로 나눠 처리함으로써 $O(n^3)$의 시간 복잡도를 가지는 알고리즘을 제안하였고, 이는 현재 많은 연구자들에 의해 사용되고 있다[5, 13]. 하지만 지배소 후위 언어에서는 굳이 Eisner의 알고리즘을 사용하지 않고도 $O(n^3)$의 알고리즘을 사용할 수 있는데, 이는 차트의 끝 위치가 항상 헤드이기 때문에 따로 헤드의 위치를 저장하지 않아도 되기 때문이다. 그림 1에서처럼 차트의 모양은 항상 직각삼각형을 이루게 되고 구문분석 시에 3개의 인덱스만 저장하면 된다.

그림 3는 한국어를 포함한 지배소 후위 언어에서의 구문분석 알고리즘이다. 차트 C[s][t]는 부분문자열 $x_s \cdots x_t$를 처리한 부분 의존구문트리의 점수 중에서 가장 높은 점수를 의미하며, bottom-up 방식의 동적 프로그래밍(dynamic programming)을 통해 문장 x에 대한 최적의 의존구문트리의 점수는 C[1][n]에 저장되게 된다. 차트를 업데이트할 때마다 r값을 저장해 두면, 백트랙킹(back-tracking)을 통해 최적의 의존구문트리를 구할 수 있게 된다.

2.3 문맥자질 집합(feature set)

앞서 설명하였듯이 한국어는 하나의 어절이 여러 개의 형태소로 이루어질 수 있기 때문에 자질의 선택이 매우 중요하다(그림 4 참조). 본 논문에서는 한국어나 일본어에서 주로 사용하는 하나의 어절 또는 분절을 대표 내용어와 대표 기능어로 표현하는 방식을 사용하였고, 구문분석 시에 유용하게 쓰일 수 있는 쉼표나 마침표 등의 정보를 이용하기 위해 마지막 기호 정보도 자질로 사용하였다. 그림 4는 한국어 어절을 추상화한 것으로 굵은 글씨로 표현된 것이 차례대로 마지막 내용어 형태소 (CM), 마지막 내용어 품사태그 (CT), 마지막 기능어 형태소(FM), 마지막 기능어 품사태그(FT), 마지막 기호(LS), 마지막 기호 품사태그 (ST)이다.

그림 4. 한국어 어절의 추상화

다음은 본 논문에서 제안하는 의존구문분석에서 사용한 유니그램(uigram) 자질, 바이그램(bigram) 자질, 부분트리 비교 자질들이다. 아래첨자(-1, -0, +0, +1)는 각각 대상 어절의 왼쪽 어절, 의존소 후보, 지배소 후보, 대상 어절의 오른쪽 어절을 의미하며, (+,-)는 이진값(boolean)을 의미한다.

● 유니그램 자질 (의존소 후보, 지배소 후보)

FT_{-1}, EJ, CM, CT, CM/CT, FM, FT, FM/FT, CT_{+1}, LS,

Comma (+,−), TopicMarker (+,−)

● 바이그램 자질 (의존소 후보+지배소 후보)
$FT_{-0}LS_{-0}CT_{+0}$, $FM_{-0}FT_{-0}LS_{-0}CT_{+0}$, $FM_{-0}LS_{-0}FM_{+0}$, $CT_{-0}FT_{-0}LS_{-0}CT_{+0}$, $CT_{-0}FM_{-0}FT_{-0}LS_{-0}CT_{+0}$, $CM_{-0}FM_{-0}LS_{-0}CM_{+0}$, $FM_{-0}LS_{-0}CM_{+0}$, $CT_{-0}LS_{-0}CT_{+0}$, HasSameFM (+,−), HasSameFT (+,−), Distance(1,2,3,4-5,long)

● 부분트리 비교 자질
DiffChildNumber (+,0,−), HasSameFMChild (+,−), SameFMChilds, SameFMChildNumber, HasSameFTChild (+,−), SameFTChilds, SameFTChildNumber

특히 마지막의 부분트리 비교 자질로서 두개의 부분트리가 공통으로 가지고 있는 기능어형태소(FM) 정보와 기능어품사태그(FT) 정보가 사용되었다.

2.4 온라인 학습

최적의 가중치 벡터 w를 구하기 위해 본 논문에서는 온라인 학습 방법 중의 하나인 Collins(2002)[24]의 averaged (structured) perceptron 알고리즘을 사용하였다. 이 방법은 SVM이나 MIRA 알고리즘[11]이 수행하는 최적화 (optimization) 과정이 필요 없어, 쉽게 구현이 가능하고 학습 속도도 다른 알고리즘에 비해 매우 빠르다. 그림 5는 의존구문분석을 위한 온라인 학습방법인 averaged perceptron 알고리즘의 의사코드(pseudo code)이다.

$$F(x,y) = \sum_{(i,j)\in y} f(i,j)$$

```
Input: training data T = {(x_t, y_t)} t=1..T
Initialize w^0=0, v=0, i=0

for all 1≤n≤N:
   for 1≤t≤T:
      w^(i+1) = w^(i) + F(x_t, y_t) - F(x_t, y_t')
      v = v + w^(i+1)
      i = i+1
Output: w = v/(NT)
```

그림 5. averaged perceptron 알고리즘

의존트리를 구성하는 의존관계, 즉 에지(edge)의 자질이 f(i, j)의 지역 자질 벡터(local feature vector)로 표현된다면, F(x, y)는 입력문장 x와 그에 대응하는 의존트리 y의 전역 자질 벡터(global feature vector)를 의미한다. 학습데이터의 각 문장에 대해서 학습모델이 선택한 최적의 의존트리 y'가 실제 정답인 의존트리 y와 다를 경우에만 가중치 벡터(w)가 업데이트되며 이러한 과정은 미리 지정한 반복횟수(N)만큼 계속된다. 일반적으로 과적합(overfitting) 방지 목적으로 널리 쓰이고 있는 평균화(averaging)를 위해 누적벡터 v를 추가적으로 사용하였으며, 최종적으로 선택되는 가중치는 v를 NT로 나눈 평균값이 사용된다. 자세한 알고리즘 설명은 Collins(2002)[24]를 참조하기 바란다.

3. 의존구문분석 실험

3.1 실험 데이터

본 논문에서 제안하는 온라인 학습을 이용한 한국어 구문분석 성능을 평가하기 위해 95년도 국어정보베이스(KIBS[2]) 말뭉치를 지배소 후위 원칙에 따라 구구조 구문트리에서 의존트리 형태로 변환하였다. 하나의 구(phrase)가 두개 이상의 터미널노드를 가질 경우 인접한 노드를 지배소로 설정하였다. 이렇게 변환한 의존트리 말뭉치는 총 12,084문장이며 이중 랜덤하게 선택한 10,876문장(90%)을 학습데이터로 사용하였고, 나머지 1,208문장(10%)을 평가데이터로 사용하였다. 구문분석의 기본 단위는 어절이며 문장당 평균 어절수는 11.8어절이다. 표 1은 KIBS말뭉치를 CoNLL형식에 맞게 변환된 의존트리 말뭉치의 예제 문장이다.

표1. 한국어 의존트리 말뭉치 예제

1	10_년	nno_nbu	2	DEP
2	동안_에	ncn_jca	3	DEP
3	2_배_,	nnc_nbu_sp	6	DEP
4	20_년	nno_nbu	5	DEP
5	동안_에	ncn_jca	6	DEP
6	3_배_가	nnc_nbu_jcs	7	DEP
7	성장_하_였_다_.	npca_xsv_ep_ef_sf	0	ROOT

3.2 실험결과 및 분석

2장에서 제안한 지배소 후위 언어를 위한 CYK 알고리즘과 한국어 의존구문분석을 위한 자질집합을 이용하여 실험을 실시하였다. 모든 실험은 Intel Core 2 CPU (2.13GHz), 4G 메모리의 리눅스 시스템에서 실행되었으며, 다른 시스템과의 비교를 위해 Shift-Reduce 방식과 Cascaded Chunking 방식의 트랜지션 기반 의존구문분석 알고리즘은 자바로 재구현하여 실험하였다. 공정한 비교를 위해 본 논문의 제안방법과 똑같은 자질 집합을 사용하였으며, SVM의 커널은 linear kernel을 사용하였다. 표 2는 최종 구문분석 성능 표이며, 위쪽 3개의 성능을 제외한 나머지는 해당 논문에서 인용한 성능이다.

[2] http://kibs.kaist.ac.kr/kibs

	정확률	
	어절	문장
Proposed method	**88.42**	**35.13**
Forward transition- (SVM)	86.10	29.28
Cascaded chunking- (CRF)	85.98	30.93
Forward transition- (이용훈 2008b)	**88.25**	**36.42**
Cascaded chunking- (오진영 2008)	87.30	31.95
Backward transition- (이용훈 2008b)	87.07	32.70
Lexicalized prob. (Chung 2004)	86.74	34.05
Unlexicalized prob. (Chung 2004)	85.62	32.25

표 2. 최종 의존구문분석 성능 비교

표 2에서 알 수 있듯이 제안모델은 어절 단위 정확률(accuracy)이 88.42%이고, 문장 단위 정확률(complete)이 35.13%이었다. 이용훈(2008b)[22]의 순방향 트랜지션 기반 의존구문분석 시스템보다 문장 단위 정확률은 약간 떨어진 반면, 어절 단위 정확률은 소폭 상승하였다. 하지만 똑같은 자질집합을 사용할 경우 2% 이상의 성능차이를 보임을 알 수 있다. 무엇보다도 제안 모델의 장점은 빠른 학습속도이다. 온라인학습(online learning)을 이용할 경우 SVM이나 CRF 등의 배치학습(batch learning)을 사용할 때보다 10배 가까이 빠른 학습속도를 보였으며, 비교적 적은 자질집합 (816,383)으로 최고의 성능을 보여주었다.

4. 결론

본 논문에서는 한국어 의존구문분석 시에 고려해야 할 점들을 알아보고 한국어에 적합한 문맥자질 집합을 제안하였다. 또한 한국어의 지배소 후위 특성과 투사성을 이용하여 Eisner(1996)[12] 알고리즘을 사용하지 않고도 $O(n^3)$의 CYK알고리즘이 가능함을 보였다.

온라인 학습을 통해 얻어진 최적의 가중치 벡터를 이용하여 구문분석을 수행한 결과 빠른 학습 속도와 비교적 적은 자질집합에도 불구하고 어절 단위 정확률 88.42%의 성능을 얻을 수 있었다. 특히 한국어나 일본어 등의 교착어에 발달한 조사나 어미 등의 기능어 정보는 의존관계를 단어와 단어간의 관계로 보지 않고 부분트리와 부분트리의 의존관계로 바라볼 수 있는 가능성을 제시하였다. 이는 올바른 구와 구의 의존관계, 절과 절의 의존관계를 찾는데 유용할 것으로 기대된다.

추후에 부분트리의 유사성을 표현할 수 있는 좀더 정교한 방법을 고안하거나 세종사전과 같은 고품질의 격틀사전을 이용한다면 불필요한 결과의 생성을 억제하고 성능을 높일 수 있으리라 예상된다.

감사의 글

본 논문은 2010년도 두뇌한국21사업, 포항공과대학교 정보통신연구소 자체 학술연구과제(선도과제), 그리고 한국과학재단 기초연구사업(No. 2010-0012662)의 지원으로 수행되었습니다.

참고 문헌

[1] M. Collins, “Head-Driven Statistical Models for Natural Language Parsing”, Ph.D thesis, University of Pennsylvania, 1999

[2] E. Charniak, “A Maximum-Entropy-Inspired Parser”, In Proc. of NAACL, pp.132-139, 2000

[3] S. Buchholz, E. Marsi, “CoNLL-X Shared Task on Multilingual Dependency Parsing”, In Proc. CoNLL, pp.149-164, 2006

[4] J. Nivre, “An Efficient Algorithm for Projective Dependency Parsing”, In Proc. of IWPT, pp.149-160, 2003

[5] R. McDonald, K. Crammar, F. Pereira, “Online Large-margin Training of Dependency Parsers”, In Proc. of ACL, pp.91-98, 2005

[6] J. Nivre, J. Nilsson, “Pseudo-Projective Dependency Parsing”, In Proc. of ACL, pp.99-106, 2005

[7] J. Nivre, “Non-Projective Dependency Parsing in Expected Linear Time”, In Proc. ACL-IJCNLP, pp.351-359, 2009

[8] H. Yamada, Y. Matsumoto, “Statistical Dependency Analysis with Support Vector Machines”, In Proc. of IWPT, pp.195-206, 2003

[9] T. Kudo, Y. 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

[10] T. Kudo, Y. Matsumoto, “Japanese Dependency Analysis using Cascaded Chunking”, In Proc. of CoNLL, pp.63-69, 2002

[11] K. Crammer, Y. Singer, “Ultraconservative Online Algorithms for Multiclass Problems”, Journal of Machine Learning Research, pp.951-991, 2003

[12] J. Eisner, “Three New Probabilistic Models for Dependency Parsing: An Exploration”, In Proc. of CONLING, pp.340-345, 1996

[13] R. McDonald, F. Pereira, “Online Learning of Approximate Dependency Parsing Algorithms”, In Proc. of EACL, pp.81-88, 2006

[14] J. Nivre, R. McDonald, “Integrating Graph-Based and Transition-Based Dependency Parsers”, In Proc. of ACL, pp.950-958, 2008

[15] A. Martins, D. Das, N. Smith, E. Xing, “Stacking Dependency Parsers”, In Proc. of EMNLP, pp.157-166, 2008
[16] 윤준태, “공기 관계 기반 어휘 연관도를 이용한 한국어 구문 분석”, 연세대학교 박사학위 논문, 1997
[17] 김학수, 서정연, “어휘 의존 정보에 기반한 한국어 통계적 구문 분석기”, 97년도 정보과학회 인공지능 연구회 춘계 발표 논문집, pp. 74-90, 1997
[18] 김형근, “확률 의존 문법을 이용한 한국어 분석”, KAIST 석사학위 논문, 1994
[19] H.J. Chung, “Statistical Korean Dependency Parsing Model based on the Surface Contextual Information”, Ph.D. thesis, Korea University, 2004
[20] 우연문, 송영인, 박소영, 임해창, “지배가능 경로 문맥을 이용한 의존 구문 분석의 수식 거리 모델”, 정보과학회논문지: 소프트웨어 및 응용 제34권 제2호, 2007
[21] 이용훈, 이종혁, “기계학습 기법을 이용한 한국어 구문분석”, In Proc. of KISS, pp.285-288, 2008a
[22] 이용훈, 이종혁, “SVM을 이용한 결정적 한국어 의존 구문분석”, 한국어정보학 제10권 제2호, pp.7-14, 2008b
[23] 오진영, 차정원, “CRFs를 이용한 강건한 한국어 의존구조 분석”, 제20회 한글 및 한국어 정보처리 학술대회 논문집, pp 23-28, 2008
[24] M. Collins, “Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms”, In Proc. of EMNLP, pp.1-8, 2002
[25] M. Collins, B. Roark, “Incremental Parsing with the Perceptron Algorithm”, In Proc. of ACL, pp.111-118, 2004