인지과학, 제22권 제4호
Korean Journal of Cognitive Science
2011, Vol. 22, No. 4, 429～447.

투사성과 재탐색을 이용한 결정적 한국어 의존구조 분석의 보정기법*

박 영 민
서강대학교 컴퓨터공학과

서 정 연†
서강대학교 컴퓨터공학과
바이오융합기술협동과정

본 논문에서는 투사성의 원칙을 이용해 결정적 한국어 의존 구문분석을 보완하는 방법을 제안한다. 우리는 결정적 의존 구문분석에 투사성의 원칙을 이용하여 의존 구문분석의 오류를 찾아내고 투사성의 원칙을 만족하도록 수정하여 오류를 개선하였다. 제안하는 모델은 기존의 결정적 의존 구문분석에 비해 높은 정확률을 제공하고 결정적 의존 구문분석의 장점을 유지할 수 있다. 또한 우리는 중간 분석결과를 이용한 자질모델을 제안하였다. 우리는 실험을 통하여 제안한 의존 구문분석 모델이 기존의 의존 구문분석 모델들에 비해 성능이 향상됨을 보였다.

주제어 : 자연어처리, 의존관계, 의존 구문분석, 퍼셉트론 기계학습, 기계학습

---

* 이 논문은 2010년 '한글 및 한국어 처리 학술대회에서 발표했던 '최대신장트리를 이용한 한국어 의존구문분석'의 내용을 보완, 확장한 것임.
* 이 논문은 한국연구재단의 중견연구자 프로그램(NO.20110027537)의 지원을 받아 수행된 연구임.
† 교신저자: 서정연, 서강대학교 컴퓨터학과 / 바이오융합기술 협동과정
연구분야: 자연어처리, 대화 시스템
E-mail: seojy@sogang.ac.kr

서 론

구문분석은 문장의 문법적인 구조를 분석하여 문장 성분사이의 관계를 파악하는 기법이다. 구문분석의 과정을 통하여 목적격 관계, 주격 관계와 같은 수식관계들을 파악할 수 있으며 이 정보들을 이용하여 문장의 정확한 의미를 분석할 수 있다.

구문분석은 문장 구성성분의 조합을 찾는 구 구조 구문분석(phrase structure parsing)과 단어와 단어 또는 어절과 어절 간의 의존-지배 관계를 찾는 의존 구문분석(dependency parsing)을 중심으로 많은 연구가 이루어져왔다[1][2]. 한국어의 경우 자유로운 어순구조 때문에 구 구조 구문분석을 적용할 때 필요한 문법 규칙의 수가 크게 증가하게 된다. 뿐만 아니라 빈번한 문장 구성성분의 생략으로 인해 더 많은 문법 규칙을 요구한다. 이와 같이 통상적인 구 구조 구문분석 기법을 사용할 경우 문법 규칙의 수가 지나치게 증가하여 구문분석에 대한 애매성이 높아지고 정확한 구문분석이 어렵게 된다. 이러한 이유로 한국어를 위한 구문분석은 구 구조 구문분석 보다 의존 구문분석이 유리하다고 할 수 있다. 본 연구에서는 의존 문법을 이용하여 한국어 문장의 의존 구조를 분석한다. 한국어 의존 구문분석의 예를 (그림 1)에서 확인할 수 있다. (그림 1)에서 실선으로 된 화살표는 '의존→지배 관계'를 표현한다. 의존-지배관계에서 지배소란 의미의 중심이 되는 요소를 말하고 의존소는 지배소가 갖는 의미를 보완(수식)해주는 요소를 말한다. 예를 들어 (그림 1)에서 '기업인들이'는 '주장했다'의 주어로서 수식하고, '오늘은'은 '협상할'을 부사로서 수식한다. 그리고 트리 전체의 상위 어절인 '주장했다'는 문장의 루트(root) 노드와 의존관계에 있다.

의존 구문분석은 문장 내의 단어와 단어 또는 어절과 어절사이의 의존-지배관계를 분석하는 작업으로 주로 결정적 의존 구문분석과 비결정적 의존 구문분석의 2가지 형태로 연구가 되어 왔다[3].

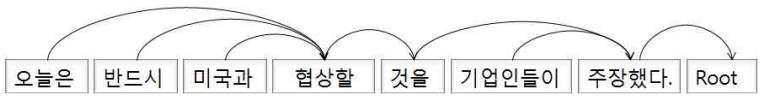

그림 1. 한국어 의존 구문분석의 예

결정적 의존 구문분석 방법은 지역적으로 최적인 결과를 연속적으로 선택해 나가는 탐욕적 알고리즘(greedy algorithm)에 기반 한 방법이다. 지역적으로 의존 관계를 결정한다는 것은 문장 전체를 탐색하지 않고 문장 일부만을 고려해 의존 관계를 결정하는 것을 말하고, 이때 결정된 결과를 최적의 해로 판단하고 분석하기 때문에 결정적 알고리즘으로 볼 수 있다. 결정적 의존 구문분석은 이전에 결정된 의존관계를 계속 유지하기 때문에 이전에 결정된 중간 분석결과(parsing history)를 사용할 수 있으며 상대적으로 자유로운 자질사용이 가능한 장점이 있다. 또한 비결정적 의존 구문분석에 비해 상대적으로 빠른 속도를 제공한다. 하지만 지역적으로 선택된 결과를 그대로 유지하기 때문에 잘못 분석된 결과가 다음 분석에 영향을 주게 되는 오류전파현상이 있고 이러한 특징 때문에 긴 문장을 분석할 때 불리하게 작용할 수 있다[3].

비결정적 의존 구문분석은 가능한 구문트리 중 가장 높은 점수의 구문트리를 찾는 최대신장트리(Maximum spanning tree)알고리즘을 이용해 의존 구문분석을 수행하는 방법이다[4]. 비결정적 의존 구문분석은 자질사용이 제한적이지만 전역적 탐색을 하기 때문에 오류전파가 없는 장점이 있다[3].

본 논문에서는 기본적인 결정적 의존 구문분석 모델을 제안하고, 투사성의 원칙을 이용하여 오류를 찾는다. 이때 발견 된 오류를 재탐색 알고리즘을 수행하여 보정한다. 또한 본 논문에서는 분석 중간 결과를 이용한 Sibling pair자질과 Grand parent자질을 제안한다. 제안하는 알고리즘은 한국어와 같이 지배소 후위의 원칙과 투사성의 원칙이 적용되는 언어에서 기존의 모델들 보다 더 높은 성능을 기대할 수 있다.

본 논문의 구성은 먼저 기존의 의존 구문분석에 관한 연구들에 대해 살펴보고, 제안하는 의존 구문분석 알고리즘과 자질 모델을 설명한다. 그리고 제안한 방법을 검증하기 위한 실험 및 결과에 대해서 설명한다.

관련 연구

의존 구문분석의 연구는 크게 상태 전이 기반의 결정적 알고리즘과 최대신장트

리 기반의 비결정적 알고리즘의 두 가지 큰 뿌리를 두고 연구되어 왔다. 결정적 알고리즘은 지역적으로 최적인 의존구조를 최적의 해로 선택하면서 의존 구문분석을 수행하는 방법이다. 결정적 의존 구문분석 알고리즘은 Kudo에 의해 처음 제안되었다[5]. Kudo는 SVM 분류기를 이용해 뒤에 인접한 단어들과 의존관계를 결정하고 Cascade Chunking을 수행하는 방식으로 의존 구문분석을 수행하였다. Kudo의 모델은 일본어와 같은 지배소 후위의 원칙(head-final)언어를 대상으로 제안되었지만, Yamada에 의해 다른 언어에서도 사용할 수 있도록 확장되었다[6]. 또 다른 결정적 의존 구문분석 알고리즘은 Shift-Reduce 기반의 모델인 Nivre의 Maltparser가 있다[7]. Maltparser는 Left, Right, Shift, Reduce 의 4가지 동작을 취하는데 O(n)의 시간복잡도를 요구한다. 그리고 Nivre는 비투사성(non-projective) 언어의 구문분석을 위해 Reduce 대신 Swap 동작을 추가하였는데 이 모델은 최악의 경우 $O(n^2)$의 시간복잡도를 요구하지만 평균적으로 선형시간을 기대할 수 있다는 것을 실험으로 증명하였다[8]. 비결정적 의존 구문분석 알고리즘은 구문트리의 점수를 계산하고 가능한 모든 구문트리 중 가장 높은 점수의 트리를 구하는 방법으로 주로 CKY 기반의 동적 프로그래밍 알고리즘이 사용되었다. 기본적인 CKY 알고리즘을 이용한 의존 구문분석은 $O(n^5)$ 의 시간복잡도를 요구하지만 Eisner는 지배소를 중심으로 차트를 좌우로 나누어 처리해서 시간복잡도를 $O(n^3)$으로 낮추었다[9]. Eisner의 모델이 투사성 언어에 대해서만 사용 가능한 단점이 있는데 McDonald는 Chu-Liu-Edmond 최대신장트리(Maximum Spanning Tree)알고리즘을 이용해 비투사성 언어를 처리할 수 있는 방법을 제안하였다[4]. 또한 McDonald는 이 방법을 형제 쌍의 점수를 함께 계산하도록 하는 2차 최대신장트리로 확장하였다[10].

국내에서는 주로 해외의 연구들을 한국어에 적용한 연구가 많았다. 의존 구문분석에 있어서 단어간의 거리정보가 중요한 자질로서 사용된다. 우연문(2007)의 연구에서는 의존소와 지배소의 수식 거리를 좀 더 정확히 추정하기 위해 단순한 어절거리가 아닌 지배가능 경로를 고려하여 더 높은 성능을 얻어내었다[11]. 이용훈(2008)의 연구에서는 한국어 의존 구문분석을 위해 Kudo의 결정적 알고리즘을 한국어에 적합하게 수정하고 SVMs(Support Vector Machines) 분류기를 사용한 연구를 수행하였다[12]. 한국어는 지배소후위의 원칙이 있기 때문에 분석 방향에 따라 DEPEND와 SHIFT 또는 DEPEND와 REDUECE 2가지의 Action을 사용하였다. 또

한 이용훈(2010)의 연구에서는 비결정적 알고리즘인 CKY 알고리즘과 Averaged Perceptron 온라인 학습알고리즘을 이용해 구문분석을 수행하였는데 최근 연구된 한국어 의존 구문분석 중 가장 높은 어절 정확률을 보여주었다[13]. 오진영(2010)의 연구에서는 CRFs(Conditional Random Fields)를 이용해 의존관계를 추정하고 결정적 방법인 다단계 구단위화(Cascaded Chunking)를 이용한 방법으로 장문의 문장도 빠르게 처리가 가능한 의존 구문분석 연구를 수행하였다[14].

한국어 의존 구문분석 모델

투사성의 원칙을 이용한 의존 구문분석 보정

구문분석은 대상 언어의 구조적 특성을 고려하여 설계 되어야 한다. 한국어의 구조적 특성으로는 투사성의 원칙, 지배소 후위의 원칙, 지배소 유일의 원칙 등이 있으며, 이러한 특성들은 한국어 의존 구문분석기를 설계하는데 중요한 요소로 작용한다[15]. 지배소 후위의 원칙은 의존-지배관계에서 지배소가 항상 뒤에 온다는 원칙이다. 투사성의 원칙은 의존관계는 서로 교차할 수 없다는 원칙을 의미한다. 또한 지배소 유일의 원칙은 의존소에 대한 지배소는 하나만 있다는 원칙이다. 따라서 각 어절들에 대한 지배소를 하나씩 찾는 것으로 의존 구문분석을 수행할 수 있다.

예를 들어 각 단어를 그래프의 정점, 의존관계를 그래프의 간선으로 보고 최대 신장트리 알고리즘을 수행하는 McDonald[4]의 모델은 투사성의 원칙과 지배소 후위의 원칙을 고려하지 않기 때문에 한국어 의존 구문분석에 그대로 적용하기에 적합하지 않다. 예를 들어 (그림 2)과 같은 문장을 McDonald의 모델로 분석한 경우 다음과 같은 문제가 발생 할 수 있다. (그림 2)에서 '체르노빌의'의 정답 지배소는 '사태보다' 이지만 말뭉치에서 '체르노빌의'와 '원전사고는'의 의존관계가 자주 발생하여 더 높은 점수를 갖게 되면 (그림 2)처럼 잘못된 분석을 할 수 있다. 하지만 지배소 후위의 원칙을 적용하여 뒤의 어절만을 지배소로 고려하여 해결할 수 있다. 그리고 '일본의'에 대한 정답 지배소는 '원전사고는' 이지만 '일본의'와 '사태보

그림 2. 최대신장트리 모델의 오류

다' 사이의 점수가 더 높다면 잘못된 분석을 하게 된다. 이때 투사성의 원칙을 이용해 교차하지 않는 의존관계를 고려하여 문제를 해결할 수 있다. 따라서 우리는 지배소 후위의 원칙과 투사성의 원칙을 적용하여 설계한 결정적 의존 구문분석 모델을 제안한다.

첫째로 우리는 지배소 후위의 원칙을 적용한다. 의존 구문분석은 각 어절에 대한 지배소가 어떤 어절인지를 찾는 작업이라 할 수 있다. 한국어가 지배소 후위의 원칙을 따르는 특징을 이용하여 주어진 문장의 각 어절들에 대한 지배소가 뒤에 존재하는 어절들만은 지배소 후보로 선택함으로서 의존 구문분석을 수행할 수 있다. 예를 들어 (그림 2)에서 의존소 '체르노빌'에 대한 지배소 후보들은 뒤쪽에 있는 '사태보다'와 '심각하다'가 된다. 이 지배소 후보들 중 가장 높은 점수를 갖는 지배소를 선택하여 의존 구문분석을 수행할 수 있다. 이때 의존소와 지배소 후보들 사이의 의존 정도를 계산하기 위해 가중치 벡터가 필요한데 본 논문에서는 가중치를 학습하기 위한 알고리즘으로 Averaged Perceptron을 사용한다[16][17].

둘째로 우리는 지배소 후위 모델에 투사성의 원칙을 적용한다. 투사성의 원칙은 (그림 3)과 같이 의존관계는 서로 교차할 수 없다는 원칙이다. 기존의 투사성 언어를 위한 결정적 의존 구문분석 모델은 이전에 결정된 의존관계를 최적이라고 판단하고 현재 분석해야 할 의존소에 대한 지배소 후보 중에서 교차가 발생하지 않는 지배소만을 후보로 간주하여 성능을 향상 시켜왔다[18]. 이 방법은 의존 구문분석의 진행 방향에 따라 다르게 적용할 수 있다. (그림 4)는 기존의 결정적 알고리즘의 동작을 예로 든 것이다. 정방향으로 분석할 경우 (그림 4)의 왼쪽 그림과 같이

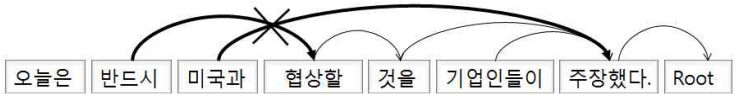

그림 3. 투사성의 원칙

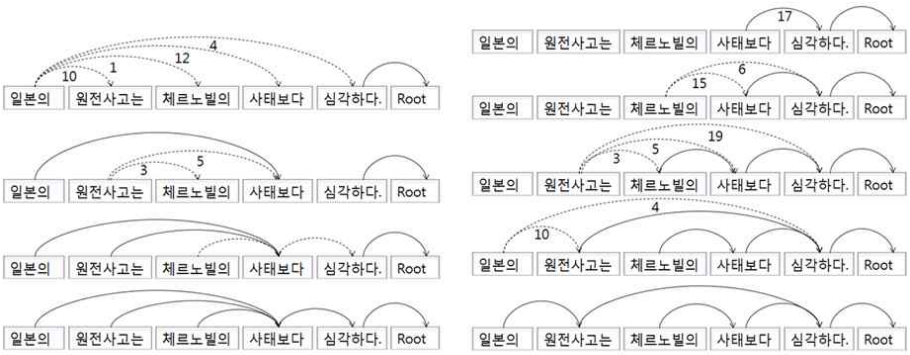

그림 4. 정방향과 역방향 분석 과정

‘일본의’의 지배소가 ‘사태보다’로 결정 되었다면 다음 의존소 ‘원전사고는’의 지배소 후보를 점선으로 연결된 ‘체르노빌의’와 ‘사태보다’ 로 제한하여 투사성을 만족하게 된다. 역방향으로 분석할 경우에는 (그림 4)에서 오른쪽 그림처럼 이전에 결정된 의존관계와 교차하지 않는 지배소 후보만을 선택하는 것으로 투사성을 만족할 수 있다. 하지만 이런 결정적 의존 구문분석 모델을 사용할 때 이전에 결정된 의존관계가 정답이 아닌 경우 다음 분석에서 정답지배소 후보를 선택할 수 없는 오류전파의 문제가 발생할 수 있다. 예를 들어 (그림 5)에서 ‘반드시’의 정답 지배소는 ‘협상할’이지만 이전에 ‘미국과’의 지배소를 ‘주장했다’로 결정하였기 때문에 ‘반드시’의 분석을 실패할 수 밖에 없게 된다. 이러한 문제를 해결하기 위해 우리는 먼저 교차를 허용해 가능한 지배소 후보의 의존관계 점수를 모두 계산하고 교차가 발생하면 오류로 판단한다. 교차가 발생하면 투사성의 원칙에 어긋나기 때문에 교차된 2 개의 의존관계 중 1 개 이상이 오류이기 때문이다. 우리는 이 오류를 보정하기 위해 재탐색 알고리즘을 수행하는 재탐색 모델을 제안한다.

재탐색 모델은 의존 구문분석을 수행하는 중에 교차하는 의존-지배관계들에 대해 다른 의존관계들을 유지한 상태에서 투사성의 원칙을 만족하도록 재탐색 과정을 수행한다. 재탐색 모델은 교차가 발생한 2 개의 의존소에 대하여 투사성의 원칙에 만족하는 의존 관계를 재 선택하기 위해 서로 교차가 발생하지 않는 모든 의존관계들만을 탐색의 범위로 정의한다. 이때 교차가 발생한 2 개의 의존소에 대해

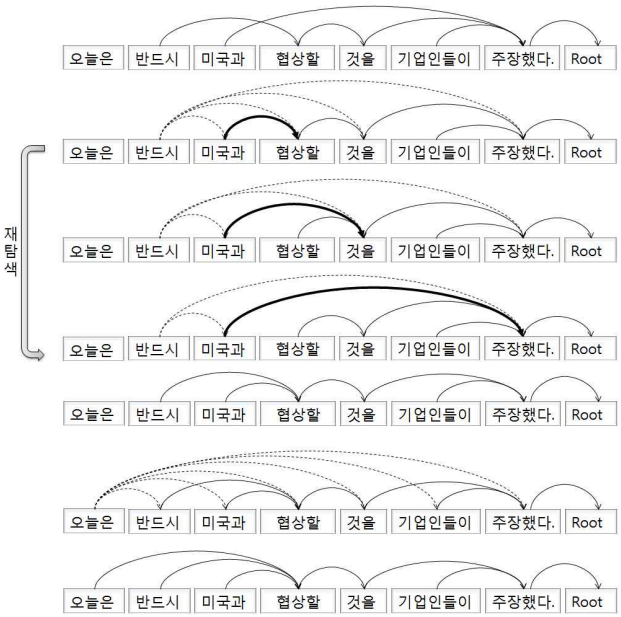

그림 5. 재탐색 모델의 분석 과정

재탐색을 수행하고 의존관계 점수의 합이 가장 높은 2개의 의존관계들이 선택된다. (그림 5)는 재탐색 모델의 분석과정의 예를 보여준다. 교차한 의존소 '반드시'와 '미국과'에 대해 재탐색을 수행하여 가장 적합한 의존관계를 재 선택하여 분석에 성공한 것을 볼 수 있다. 만약 (그림 6)과 같이 3 개 이상의 의존관계가 교차될 경우에는 교차된 의존관계들을 중 현재 분석중인 의존관계와 가장 오른쪽 의존관계만 남긴 상태로 재탐색을 수행한다. 그리고 제외했던 의존관계를 다시 만들고 교차하는 의존소가 모두 해결 될 때 까지 이 과정을 반복한다. 한 문장에서 구문분석 수행 중에 교차하는 의존 관계가 여러 번 발생할 수 있기 때문에 재탐색 알고리즘은 여러 번 호출될 수 있다.

지배소 후위 모델, 정방향 모델, 역방향 모델은 한 어절씩 지배소를 결정하면서 다음 어절의 지배소를 결정할 때 이전에 결정된 의존 관계가 변하지 않는다. 따라서 결정된 의존관계가 변하지 않기 때문에 결정적 알고리즘의 형태를 갖는 반면 재탐색 모델의 경우 재탐색 결과에 따라 의존관계가 변화할 수 있다. 따라서 재탐

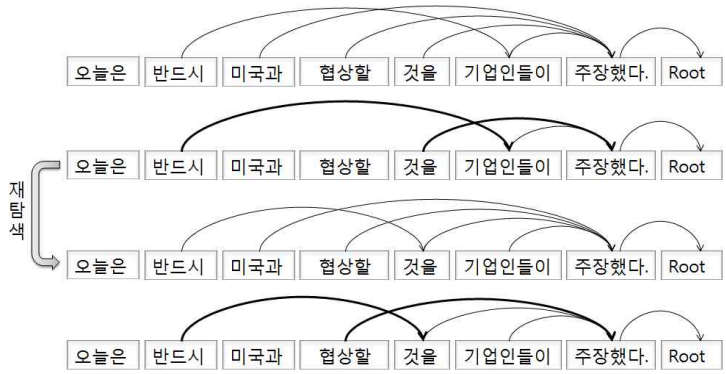

그림 6. 3 개 이상의 의존 관계가 교차된 경우

색 모델은 결정적 알고리즘에 오류전파가 발생할 가능성이 있는 위치에 부분적으로 비결정적 알고리즘을 적용하여 에러전파를 최소화함으로서 결정적 알고리즘의 단점을 극복할 수 있다.

재탐색 알고리즘의 의사코드는 (그림 7)에서 확인할 수 있다. (그림 7)에서 L은 교차된 의존소 중 왼쪽의 어절, R은 오른쪽의 어절을 의미한다. 2번째 줄과 4번째 줄의 반복문은 의존소 L과 R에 대해 교차를 만들지 않는 모든 의존관계를 탐색함

```
교차된 의존소 (L, R)
max = maxL = maxR = -∞
for( hr = all headable path of R )
  head[R] = hr;
  for( hl = all headable path of L )
    if( hl.score + hr.score > max )
      maxL = hl; maxR = hr;
      max = hl.score+hr.score;
head[L] = maxL;
head[R] = maxR;
```

그림 7. 의존관계 재탐색 알고리즘

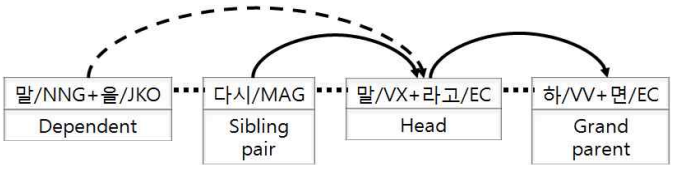

그림 8. 자질 추출의 예

을 의미한다. 여기서 hl은 현재탐색중인 L의 의존관계, hr은 현재탐색중인 R의 의존관계를 의미하고 여기서 score는 그 의존관계의 점수를 의미한다. 3번째 줄은 의존관계를 탐색할 때 R의 의존관계에 따라 L의 가능한 의존관계가 변화하기 때문에 hr을 기록하는 과정이다. 본 논문에서는 score 계산을 위해 Averaged Perceptron 알고리즘에 의해 추정된 점수를 사용한다. head는 각 어절에 대한 지배소를 기록하는 배열을 의미한다.

기존의 상태전이 기반 결정적 의존 구문분석 모델인 Nivre의 모델[7], Kudo의 모델[5]은 가장 가까운 단어를 우선으로 하고 의존관계로 판단되면 그대로 결정하기 때문에 의존관계가 멀리 있는 경우에는 높은 성능을 제공하지 못하는 경우가 많다. 본 논문에서 제안하는 재탐색 모델은 가능한 모든 지배소 후보들을 고려하기 때문에 이러한 단점을 해결할 수 있다.

앞서 설명한 모델들 중 지배소 후위모델, 정방향 모델, 역방향 모델은 $O(n^2)$의 시간 복잡도를 요구한다. 하지만 제안하는 재탐색 모델의 경우 시간복잡도가 $O(n^2)$이기 때문에 재탐색 알고리즘의 호출횟수가 m일 때 재탐색 모델의 전체 시간복잡도는 $O(m*n^2)$이며 최악의 경우 $O(n^4)$을 요구하게 된다.

자질 모델

한국어의 한 어절은 하나 이상의 형태소로 이루어져 있다. 일반적으로 각 어절의 가장 마지막에 위치하는 내용어 또는 기능어의 역할을 하는 형태소가 구문분석에 중요한 정보로 사용되는 경우가 많다[13]. 따라서 우리는 의존소와 지배소의 마지막 내용어와 기능어 형태소들을 자질로서 사용한다. 추가로 의존소와 지배소를 기준으로 거리, 주변 형태소, 포함된 기호, 동일한 형태소 사용 여부 등을 자질로

사용한다. 또한 우리는 의존 구문분석 중간결과(parsing history)로 grand parent정보와 sibling pair정보를 자질로 추출한다. 의존 구문분석 중간결과는 결정적 의존 구문분석에서 사용할 수 있는 정보이다[3]. 본 연구에서 제안하는 재탐색 모델은 결정적 의존 구문분석의 결과를 보정하는 모델이기 때문에 의존 구문분석 중간결과를 이용할 수 있다. 자질을 추출하는 예는 (그림 8)을 통해 확인할 수 있다. (그림 8)에서 사각형은 어절의 형태소/품사를 의미한다. 점선으로 된 화살표는 현재 분석 대상인 의존 관계이고 실선으로 된 화살표는 이전에 결정된 중간결과로서의 의존 관계이다. Dependent 어절(말/NNG+을/JKO)은 현재 분석중인 의존소이고 Head 어절(말/VX+라고/EC)은 의존소에 대한 지배소 후보들 중 하나이다. Sibling pair 어절(다시/MAG)은 현재 까지 분석된 결과들 중 Head 어절을 지배소로 갖는 어절이다. 그리고 Grand parent 어절(하/VV+면/EC)은 이전에 분석된 Head 어절의 지배소 어절을 의미한다. 의존소(Dependent)와 지배소(Head)를 기준으로 추출한 자질의 종류와 분석 중간결과를 이용한 자질의 종류는 부록에서 확인할 수 있다(부록1, 2 참조). 예를 들어 (그림 8)에서 기본자질의 9번 자질(fpos,smor/cpos)을 추출한다면 Dependent의 기능어 품사, 특수문자 형태소/ Head의 내용어 품사를 추출하여 “9,[JKO],None/[NNP]"의 형태로 추출하게 된다. 중간분석 결과 자질 중 5번 자질(fmor/-/fmor)의 경우에는 마찬가지로 ”S5,을[JKO]/-/면[EC]“와 같은 형태로 추출된다.

본 논문에서는 각 지배소 후보들의 점수를 계산하기 위해서 Collins의 Averaged Perceptron 알고리즘을 사용한다[16][17]. 주어진 문장 X를 $X = x_1 x_2 x_3 \ldots x_n$와 같이 표현한다면(이때 x_i 는 문장 X의 i 번째 어절) 의존소 x_i와 지배소 x_j 사이의 점수를 식 (1)과 같이 계산할 수 있다.

$$score(x_i, x_j) = w \cdot f_{x_i x_j} \tag{1}$$

이때 $f_{x_i x_j}$ 자질벡터를 의미하고 w는 가중치벡터를 의미한다. 자질벡터는 의존소 x_i와 지배소 x_j에 대해서 자질 모델에 의해 추출된 자질들로 구성되며 추출된 자질의 경우 1, 추출되지 않은 경우 0의 값을 갖는다. 가중치벡터 w는 Averaged Perceptron 학습에 의해 정해진다.

실 험

본 논문에서 제안한 의존 구문분석 모델을 평가하기 위해 ETRI 구문구조부착 말뭉치(2005)를 사용하였다[19]. 이 말뭉치는 복문이 포함된 20 어절이상의 장문을 구 묶음한 문장으로 구성되어 있고, 각 문장은 형태소와 구문구조 정보가 부착되어있다. 사용된 문장들은 문장 당 평균 17.3 개의 구 묶음으로 구성되었다. 말뭉치의 문장들은 지배소 후위의 원칙, 투사성의 원칙, 지배소 유일의 원칙을 모두 만족하는 문장들로 구성되어있다. 우리는 학습데이터를 4000개에서 20000개까지 실험하였으며 평가를 위해 3000개의 문장을 사용하였다. 또한 의존 관계의 점수를 계산하기 위해 Collins의 Averaged Perceptron 알고리즘을 사용하였고 학습 반복 횟수는 실험적으로 결정하였다(최대 10회)[16][17]. 모델의 성능 평가를 위한 척도로 아크 정확률이 사용되었으며 평가 척도에 대한 수식은 식 (2)와 같다.

$$\text{아크 정확률} = \frac{\text{정답트리와 같은 아크의 수}}{\text{정답트리에서 아크의 수}} \tag{2}$$

제안하는 재탐색 모델의 성능을 평가하기 위해 일반적인 의존 구문분석 모델인 지배소 후위 모델, 투사성을 유지하는 정방향 모델, 역방향 모델을 구현하여 비교하였다. 또한 최근 발표된 한국어 의존 구문분석기 중 가장 높은 성능을 보여준 이용훈(2010)의 CKY 모델[13]을 구현하여 비교하였다. 모두 동일한 데이터를 사용하였고 실험결과는 (그림 9)에서 확인할 수 있다.

실험 결과 정방향 모델과 지배소 후위모델이 거의 같은 성능을 보여주었다. 투사성을 만족하기 위해 제한을 적용하더라도 오류전파의 영향이 크기 때문에 때로는 오히려 낮은 성능을 보여주는 것으로 분석된다. 반면 역방향 모델은 더 높은 성능을 보여 주었는데 정방향에 비해 오류전파가 많이 줄어든 것을 예상할 수 있다. 재탐색 모델의 경우 비결정적 알고리즘을 사용하는 CKY 모델과 거의 유사한 성능을 제공하는 것을 알 수 있다. CKY 모델이 오류전파가 없는 것을 고려했을 때 제안한 재탐색 모델이 상당부분의 오류전파를 해결한 것을 알 수 있는 실험 결과이다. 실험에서 의존관계가 교차하여 의존관계 재탐색 알고리즘이 수행된 횟수

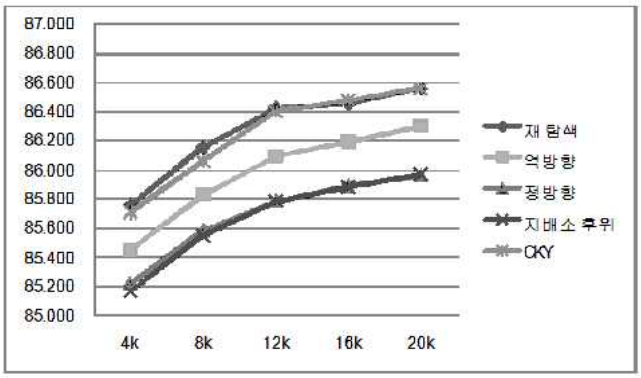

그림 9. 학습문장 수에 따른 아크 정확률 비교

는 1 문장 당 평균 0.27 회였다.

우리는 여기서 재탐색 알고리즘의 오류전파 보정성능을 명확히 평가하기 위해 재탐색 알고리즘을 호출한 문장들에 대해서만 분석을 수행하였다. 학습을 위해 8000개의 문장이 사용되었고 평가를 위해 3000개의 문장 중 재탐색 알고리즘이 호출된 문장 697개 만이 사용되었다. 재탐색 알고리즘이 수행되지 않은 문장들에 대해서는 지배소 후위, 정방향, 역방향, 재탐색 모델 모두 같은 결과를 제공하기 때문에 나머지 2303개의 문장에서는 같은 성능을 제공한다. 697개의 문장 분석 중 810회의 재탐색 알고리즘이 수행되었다. 이 문장들 중 398개의 문장에서는 성능이 향상되었고, 115개의 문장에서는 성능이 하락하였으며 나머지 문장에서는 성능이 그대로 유지되었다. 결과적으로 상황에 따라 재탐색 알고리즘을 수행하면 성능이 하락하기도 하지만 대부분의 경우 성능을 향상시키는 것으로 볼 수 있다. 697개 문장에 대한 아크 정확률은 (표 1)에서 확인할 수 있다.

우리가 제안한 모델들은 결정적 알고리즘이기 때문에 중간결과(parsing history)자질을 사용할 수 있다. 자질모델의 성능을 평가하기 위해 각 모델별로 기본 자질과

표 1. 재탐색 알고리즘 보정 효과

분석모델	지배소 후위	정방향	역방향	재탐색
아크 정확률	81.68	81.91	82.83	**84.28**

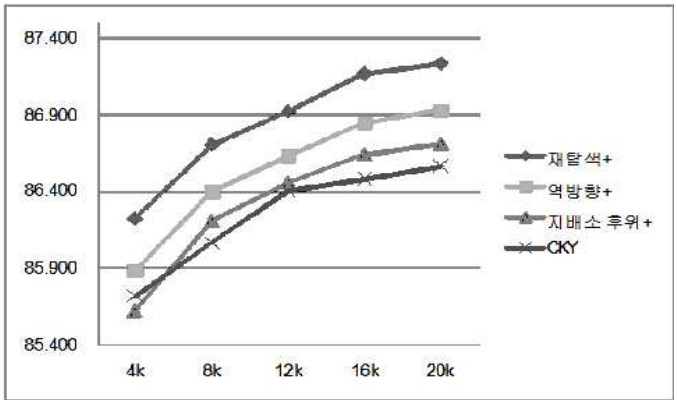

그림 10. 중간결과 자질 추가 후 성능

분석 중간결과자질을 추가하여 성능평가를 수행하였다. 정방향 모델은 grand parent 자질을 사용할 수 없으므로 제외하였다. 실험결과는 (그림 10)과 같다. 실험 결과 모든 모델에서 중간결과 자질을 추가했을 때 성능 향상이 있었다. 우리가 제안한 sibling pair 와 grand parent 자질들이 성능향상에 기여한 것을 알 수 있다. 이러한 실험 결과는 우리가 제안한 재탐색 모델이 결정적 모델과 비결정적 모델의 장점을 모두 제공함을 보여준다. 또한 한 문장 당 평균 0.27회 재탐색 알고리즘을 수행하기 때문에 실질적으로 $O(n^3)$의 시간복잡도를 요구하는 CKY 모델에 비해 유리하다고 할 수 있다.

결 론

본 논문에서는 투사성의 원칙을 이용한 보정기법을 적용하여 결정적 한국어 의존 구문분석 모델을 제안하였다. 우리가 제안하는 의존 구문분석 모델은 의존관계를 추정한 뒤 교차가 발생할 경우 투사성의 원칙에 맞도록 의존관계를 재탐색한다. 제안하는 탐색 알고리즘은 기존의 결정적 알고리즘 모델들의 단점인 오류전파를 감소시키는 효과가 있을 뿐 아니라 결정적 알고리즘의 형태를 갖고 있기 때문

에 분석 중간결과 정보를 자질로 사용할 수 있다. 우리는 분석 중간결과를 이용해 grand parent 와 sibling pair자질을 사용하였고 제안하는 모델이 비결정적 의존 구문 분석 모델보다 뛰어난 성능을 제공함을 실험으로 증명하였다. 제안하는 모델은 한국어 또는 일본어와 같은 지배소 후위의 원칙, 투사성의 원칙이 적용되는 언어에서 유용하게 사용할 수 있을 것이다. 차후 우리는 더 효과적인 자질을 연구하고, 우리의 의존 구문분석 모델을 한국어 이외의 언어에도 적용 가능한 구문분석 알고리즘으로 확장할 계획이다.

참고문헌

[1] Michael A. Covington, "A dependency parser for variable-word-order languages," Research Reprot AI-1990-01, University of Georgia, 1990.

[2] 나동렬, "한국어 파싱에 대한 고찰," 정보과학회지, 제12권, 제8호, pp.33-46, 1994.

[3] Ryan McDonald, Joakim Nivre, "Analyzing and Integrating Dependency Parsers," *Computational Linguistics*, Volume 37, Number 1, 2011.

[4] Ryan McDonald, Feranado Pereira, Kiril Ribarov Jan Hajič "Non-projective dependency parsing using spanning tree algorithms," In *Proceedings of the EMNLP-CoNLL*, pp.915-932, 2005.

[5] Taku Kudo, Yuji Matsumoto, "Japanese dependency analysis using cascaded chunking," In CoNLL, pp.18-25, 2000

[6] Hiroyasu Yamada, Yuji Matsumoto, "Statistical dependency analysis with support vector machines," In *Proceedings of IWPT*, pp.195-206, 2003

[7] Joakim Nivre, Johan Hall, Jens Nillson, Atanas Chanev, Gűlsen Eryigit, Sandra Kűbler, Svetoslav Marinov, Erwin Marsi, "MaltParser: A language-independent system for data-driven dependency parsing," In *Proceedings of the Natural Language Engineering*, pp.95-135, 2007.

[8] Joakim Nivre, "Non-projective dependency parsing in expected linear time," In

Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pp.351-359, 2009.

[9] Jason M. Eisner, "Three new probabilistic models for dependency parsing: An exploration," In *Proceedings of the COLING*, Vol.1, pp.340-345, 1996.

[10] Ryan McDonald, Feranado Pereira, "Online Learning of Approximate Dependency Parsing Algorithms," In *Proceedings of the EACL*, pp.81-88, 2006.

[11] 우연문, 송영인, 박소영, 임해창, "지배가능 경로 문맥을 이용한 의존 구문 분석 수식 거리 모델," 정보과학회 논문지: 소프트웨어 및 응용, 제34권, 제2호, pp.140-149, 2007.

[12] 이용훈, 이종혁, "SVM을 이용한 결정적 한국어 의존 구문분석," 한국어정보학 제10권, 제2호, pp.7-14, 2008.

[13] 이용훈, 이종혁, "온라인 학습을 이용한 한국어 의존 구문분석," 한국컴퓨터종합 학술발표논문집, 제37권, 제1호(C), pp.299-304, 2010.

[14] 오진영, 차정원, "다단계 구단위화를 이용한 고속 한국어 의존구조 분석," 한국시뮬레이션학회 논문지, 제19권, 제1호, pp.103-111, 2010.

[15] Changhyun Kim, Jae-hoon Kim, Junyeon Seo, Gil Chang Kim, "A Right-to-Left Chart Parsing for Dependency Grammar using Headable Paths," In *Proceedings of the International Conference on Computer Processing of Oriental Language*, pp.175-180, 1994.

[16] Michael Collins, "Discriminative training method for hidden markov models: Theory and experiments with perceptron algorithms," In *Proceedings of the EMNLP*, pp.1-8, 2002.

[17] Michael Collins, Brian Roark, "Incremental parsing with the perceptron algorithm," In *Proceedings of the ACL*, pp.111-118, 2004.

[18] Masakazu Iwatate, Masayuki Asahara, Yuji Matsumoto, "Japanese dependency parsing using a tournament model," In *Proceedings of the COLING*, pp.361-368, 2008.

[19] ETRI, http://slrdb.etri.re.kr/, "구문구조부착 말뭉치", 2005.

1 차원고접수 : 2011. 10. 24
2 차원고접수 : 2011. 12. 9
최종게재승인 : 2011. 12. 19

(*Abstract*)

Correction Method for Korean Dependency Parsing using Projectivity and Re-searching

Youngmin Park[1] Jungyun Seo[2]

[1]Department of Computer Science and Engineering, Sogang University

[2]Department of Computer Science and Interdisciplinary Program of Integrated Biotechnology, Sogang University

In this paper, we propose a modified deterministic Korean dependency parser using a projectivity. The modified parser is improved by finding errors, such as cross dependency, from the original parsing results and correcting them according to the projectivity and head-final principles. Our parser also uses parsing history information in addition to rich features, which only a deterministic algorithm can use. Results on the modified parser for ETRI(2005) corpus, that consists of complex sentences, show that our parser outperforms other parsers.

Key words : dependency parsing, natural language processing, dependency relation, perceptron machine learning, NLP, machine translation

부록 1. 기본 자질 집합

	Dep		Head
1	어절		어절
2	fpos		fpos
3	fmor		fmor
4	$fpos_{-1}$		$fpos_{-1}$
5	cpos		cpos
6	cmor		cmor
7	$cpos_{+1}$		$cpos_{+1}$
8	smor		smor
9	fpos, smor		cpos
10	fmor, fpos, smor		cpos
11	fmor, smor		fmor
12	cpos, fpos, smor		cpos
13	cpos, fmor, fpos, smor		cpos
14	fmor, smor		cmor
15	cpos, smor		cpos
16	컴마기호$_{(1,0)}$		컴마기호$_{(1,0)}$
17	열림기호$_{(1,0)}$		열림기호$_{(1,0)}$
22	거리 (1,2,3,4-5,6이상)		
18	cpos	거리 (1,2,3,4-5,6이상)	cpos
19	cmor	거리 (1,2,3,4-5,6이상)	cmor
20	fpos	거리 (1,2,3,4-5,6이상)	fpos
21	fmor	거리 (1,2,3,4-5,6이상)	fmor
23	의존소와 지배소가 동일한 $fpos_{(1,0)}$		
24	의존소와 지배소가 동일한 $cpos_{(1,0)}$		
25	의존소와 지배소가 동일한 $fmor_{(1,0)}$		
26	의존소와 지배소가 동일한 $cmor_{(1,0)}$		

접두어 f : 기능어 c : 기능어 s : 특수문자

접미어 pos : 품사 mor : 어휘

(1, 0) : (있음, 없음)

부록 2. Parsing History 자질 집합

	Dep	Head	grand parent	Dep 와 grand parent 거리
1	fpos	fpos	fpos	-
2	cpos	cpos	cpos	-
3	fpos	-	fpos	-
4	cpos	-	cpos	-
5	fmor	-	fmor	-
6	cmor	-	cmor	-
7	fpos	fpos	fpos	(1,2,3,4-5,6이상)
8	cpos	cpos	cpos	(1,2,3,4-5,6이상)
	Dep	Head	sibling pair	-
9	fpos	fpos	fpos	-
10	cpos	cpos	cpos	-
11	fmor	-	fmor	-
12	cmor	-	cmor	-

접두어 f : 기능어 c : 기능어 s : 특수문자

접미어 pos : 품사 mor : 어휘