3차 의존 파싱에 기반한 한국어 구문 분석

나승훈○
부산외국어대학교
nash@bufs.ac.kr

Third-order Dependency Parsing of Korean

Seung-Hoon Na○
Busan University of Foreign Studies

요 약

본 논문에서는 한국어 구문 분석을 위해 3차 의존 파싱 방법을 적용한 성능 결과를 제시한다. 3차 의존 파싱에서는 조부모 (grandparent) 노드 정보까지 참조함으로써 2차 자질의 한계를 넘어 보다 복잡하고 다양한 자질을 고려할 수 있다. 실험 결과 3차 의존 파싱은 기존의 2차 한국어 의존 파싱의 성능을 향상시켰다.

주제어: 3차 의존 파싱, 한국어 구문 분석, 그래프 기반, 기계 학습

1. 서론

최근 한국어에서 의존 파싱에 대한 연구는 기계 학습 기반 분류 모델 방법이 주된 축을 이루고 있다. 기계 학습 기반 방법은 통계 기반 방법에 비해 다양한 자질을 임의로 조합할 수 있다는 점, 분류 문제를 직접적으로 모델링함으로써 성능이 우수하다는 점 등의 강점을 지닌다.

기계 학습 기반 한국어 의존 파싱은 영어권의 연구와 마찬가지로 *크게 그래프 기반 방식*과 *전이 기반 방식*으로 나뉜다. 이 중, 그래프 기반 파싱에서 사용 자질의 *차수* (order)는 트리의 점수 계산을 위해 Factorization의 단위의 크기로, 차수가 높을수록 복잡한 자질을 고려할 수 있다. 그러나, 기존의 한국어 의존 파싱은 부모 (parent)와 두 개의 인접한 자식 (childen)노드의 정보를 이용한 *제한된 2차 모델*만을 사용하였고 3차 자질과 같은 고차 모델을 적용한 실험적 연구는 없었다.

본 논문에서는 한국어 구문 분석을 위해 *3차 의존 파싱*을 적용한 실험 결과를 제시한다. ETRI 구문분석 말뭉치에서 3차 의존 파싱을 한국어 분석에 적용한 결과 기존의 2차 의존 파싱의 최고 성능을 더욱 개선시켰다.

2. 관련 연구

의존 파싱은 크게 그래프 기반 방식 [1-7]과 전이 기반 방식 [8-10]의 두 가지 접근법으로 나뉘어 연구되고 있다. 그래프 기반 방식은 전역적 탐색 방법으로 오류가 전파되지 않은 장점이 있는 반면, 부분 의존 트리 정보를 참조할 수 없어 자질의 사용에 제한이 있다. 전이 기반 방법은 지역적 탐색으로 history 자질 등 부분 의존 트리의 구조를 이용할 수 있으나, greedy탐색 방법으로 인해 오류가 전파될 수 있는 한계가 있다. 그래프 기반과 전이 기반 방법의 통합 연구도 진행되었는데, [11]은 그래프 기반 방법과 전이 기반 방법이 서로 보완될 수 있음을 분석해내어, 다른 방법의 *파싱 결과를 추가 자질로 사용*하는 단순한 방법을 통해 약 2%의 성능 증가를 이끌어내었다. [12]는 beam-search 기반 방식으로 파싱 과정에서 두 *방법의 자질을 결합* (integration)하여 사용하여 성능을 향상시켰다.

한편, 그래프 기반 파싱에서 사용 *자질의 차수* (order)는 성능에 중요한 영향을 미치는 인자 중 하나이다. 초기의 그래프 기반 방식은 부모(parent)와 자식(child) 간의 정보를 이용한 *1차 모델* [2], 그리고 부모(parent)와 두 개의 인접한 자식 (childen)간의 정보를 이용한 *제한된 2차 모델* [1,3]이었다. 이러한 모델이 더욱 확장되어, [4-6]에서는 *조부모* (grand parent)노드와의 정보를 포함한 *3차 모델 (일반화된 2차 모델 포함)*을 이용하여 성능을 더욱 향상시켰다. [7]은 *4차 모델*까지 제안하여 추가 성능 향상을 이끌어내었다.

이러한 차수의 효과성에도 불구하고, 기존의 한국어 의존 파싱 연구에서는 [13-15], 초기의 1차 및 제한된 2차 모델에 국한되어, 고차 의존 파싱에 대한 적용 연구가 수행되지 않았다. 기계 학습 기반 한국어 의존 파싱에서 고차 자질의 효과를 논의한 연구는 [16]가 있는데 이는 전이 기반 연구에 해당된다.

3. 3차 의존 파싱에 기반한 한국어 구문 분석

3.1 Factorization

그래프 기반 의존 파싱은 주어진 문장에 대해 가능한 의존 트리마다 점수를 얻어내어, 가장 점수가 높은 트리를 결정하는 방법이다. 그래프 기반 파싱에서는 트리의 점수를 직접 정의하지 않고 factorization을 통해 부분트리의 점수의 합으로 정의한다. 이때 차수(order)란 부분트리를 구성하는 에지의 최대 개수를 일컫는다.

형식적으로, 주어진 문장 x에 대해 후보 의존 트리 y에 대한 점수를 $score(x,y)$이라 하자. Factorization은 y의 점수를 부분 트리 p의 점수들의 합으로 정의하는 과정으로, 다음 식 (1)과 같이 정리될 수 있다.

$$score(x,y) = \sum_{p \in y} score_{Type(p)}(x,p) \qquad (1)$$

여기서, $Type(p)$는 부분 트리 p의 *유형*을, $score_{Type(p)}(x,p)$는 부분 트리 p의 점수를 지칭한다. 한편, 식 (1)에서 부분 트리의 점수 $score_{Type(p)}(x,p)$는 다음과 같이 형식화 된다.

$$score_{type}(x,p) = \boldsymbol{w}_{type} \cdot \boldsymbol{f}_{type}(p) \qquad (2)$$

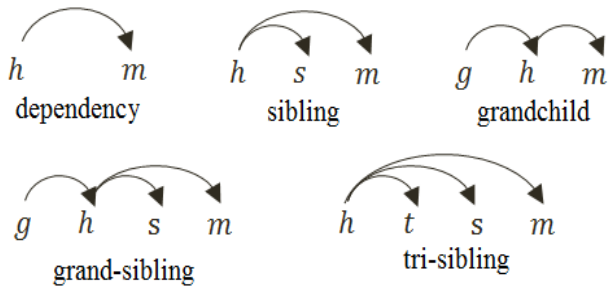
$\boldsymbol{f}_{type}(p)$는 $type$의 유형에 해당하는 부분 트리 p의 *자질 벡터*, $\boldsymbol{w}_{type}$는 대응되는 *자질 가중치 벡터*이다. 식 (1-2)처럼 부분 트리의 합에 기반한 파싱 모델을 *factored model*이라고 부른다.

부분 트리의 유형 $Type(p)$는 차수 및 구조에 따라 달라지는데, 표 1에서는 1차부터 3차까지 가능한 총 5가지의 부분 트리의 유형을 보여준다 [6].

표 1 Factored model에서 부분 트리의 유형

부분(part)의 유형	표기	차수
dependency (D)	(h,m)	1
sibling (S)	(h,s,m)	2
grandchild (GC)	(g,h,m)	2
grand-sibling (GS)	(g,h,s,m)	3
tri-sibling (TS)	(h,t,s,m)	3

표 1에서 제시된 각 부분 트리 유형의 구조는 다음과 같다.



3.2 자질 추출 (Feature extraction)

본 논문에서 1차, 2차 부분트리에 해당하는 dependency, sibling 자질은 [14,15]에 제시된 정보를 참조하여 정의하였다. Grandchild, grand-sibling을 위한 자질은 각 어절별로 내용어와 기능어의 품사 및 태그를 조합하여 어절의 기본정보를 구성한 후, 이들을 조합하여 구성하였다.

사용된 자질을 구체적으로 기술하기 위해, i번째 단어에 대한 *기본 자질*을 표 2와 같이 정의하도록 하자.

표 2. 기본 자질의 정의

표기	정의
cm_i	i번째 단어의 최좌측 내용형태소의 문자열
ct_i	i번째 단어의 최좌측 내용형태소의 태그
fm_i	i번째 단어의 최우측 기능형태소의 문자열
ft_i	i번째 단어의 최우측 기능형태소의 태그
M_i	$cm_i fm_i$
T_i	$ct_i ft_i$
W_i	$cm_i ct_i fm_i ft_i$

여기서, *단어*란 의존 파싱을 위한 기본 단위인 *구문 노드* (syntactic node)에 대응되는 것으로, 적어도 하나의 내용형태소를 포함, 복수개의 형태소로 이루어진다. 단어의 *내용형태소*는 동사, 관형사, 부사류를, *기능형태소*는 어절의 조사, 어미류를 가리킨다. 기본 자질 중에서 *복합 자질* (compound feature)은 2개 이상의 자질들이 조합된 것으로 여기서는 단순히 concatenation을 사용하여 유도하였다. 표 2의 M_i, T_i, W_i이 복합 자질들인데 이들은 단어내의 내용형태소와 기능형태소의 문자열 또는 품사 정보의 자질들이 조합된 것이다.

최종적으로 추출된 자질들은 표 2의 기본 자질에 기반을 두며, 표 3은 각 부분 트리 유형별 대표적인 자질들을 보여준다. L은 변별성을 가질 수 있는 단어간 최대 거리를 의미하는데, 본 논문에서는 L을 6으로 고정시켰다.

표 3. 부분 트리 유형별로 추출된 자질들

유형	정의
D	$Dist_{\min(\lvert h-m\rvert,L)}$, T_hT_m, W_hW_m, W_hM_m, M_hW_m, W_hT_m, T_hW_m, M_hT_m, T_hM_m, M_h, T_h, W_h, M_d, T_d, W_d, cm_hct_h, cm_mct_m, $cm_hct_hcm_mft_m$, $cm_hcm_hcm_mct_m$, $cm_hcm_hct_m$ 등 [14,15]참고
S	$DistSib_{\min(\lvert s-m\rvert,L)}$, $T_hT_mT_s$, $T_hT_mM_s$, $T_hT_mW_s$, $M_hM_mM_s$, $T_hW_mW_s$, $T_hW_mW_s$, M_mM_s 등
GC	$DistGC_{\min(\lvert g-m\rvert,L)}$, $T_gT_hT_m$, $T_gT_hM_m$, $W_gM_hT_m$, $M_gM_hM_m$, $W_gW_hT_m$, W_gM_m, W_gW_m
GS	$T_gM_hM_mM_s$, $T_gW_hW_mW_s$, $T_gT_hM_mM_s$, $W_gW_hW_mW_s$

3.3 디코딩 (Decoding)

디코딩은 식 (1)의 정의에 따라 계산된 의존 트리 점수가 최대가 되는 트리를 찾는 과정이다. 한국어의 경우에는 *지배소 후위 원칙*이 적용되기 때문에 일반적인 projectivity제약에 더하여, 지배소가 후위에 배치되어야 함을 의미하는 *head-final제약*도 추가로 적용해야 한다. 따라서, 한국어 의존 파싱을 위한 3차 디코딩 알고리즘

은 최대 점수를 갖는 *head-final-projective 트리* (head-final제약을 추가로 만족하는 *projective* 트리) 탐색하는 과정으로, [6]의 3차 알고리즘의 단순화된 버전을 이용한다.

4. 실험 결과

3차 모델의 성능 평가를 위해, ETRI구문부착말뭉치 약 10만문장을 평가 집합으로 사용하여, 이중 90%를 학습데이터로 10%를 평가데이터로 활용하였다. 자질 가중치 학습을 위해 Averaged perceptron을 사용하였으며, 학습데이터의 반복횟수는 최대 10회로 제한하였다.

성능 평가를 위해 다음의 3가지 지표를 사용하였다.

- UAS (unlabeled attachment score): 시스템에 의해 정확하게 분석된 의존 관계의 비율로 표준 평가 지표 (standard metric)이다.
- UAS (macro): 문장별로 UAS를 구하여 이를 평균한 것으로 [15]와 비교를 하기 위해 사용한다 [1].
- Complete: 문장내 의존 관계 전체가 정확하게 분석된 비율

본 실험에서는 3.2장에서 제시된 자질 집합에 기반하여 다음의 3가지의 모델을 비교하였다.

- 1st-order: D의 자질 사용
- 2st-order: D, S의 자질 조합 사용
- 3rd-order: D, S, GC, GS의 자질 사용

세 가지 방법을 비교한 결과가 표 4에 제시되어 있다.

표 4. 차수에 따른 성능 비교

	UAS	UAS (Macro)	Complete
[15]	N/A	88.06*	N/A
1st-order	88.06	88.13	12.87
2nd-order	88.24	88.31	12.88
3rd-order	**88.49**	**88.55**	**14.16**

표 4에서 보다시피 확장된 2차 자질 (G) 및 3차 자질 (GS)을 통해 성능이 더욱 향상되었다. 덧붙여, 표 4에서는 동일 집합에서 현재까지 최고의 성능을 보여주는 [15]의 결과도 함께 인용하였다. 여기서 [15]는 2차 파싱 방법으로 D,S에 해당하는 자질만을 주로 고려한 방식이다. 결과적으로, 3차 의존 파싱 방법은 동일 테스트 집합에서 알려진 기존의 최고 성능 [15]를 더욱 개선시킬 수 있었다.

5. 결론

본 논문에서는 3차 의존 파싱을 한국어 구문 분석에 적용한 결과를 제시하였다. 실험 결과, 동일 집합에서 알려진 기존의 최고 성능을 더욱 개선시켰다. 향후, tri-sibling자질을 추가로 이용하여 3차 의존 파싱에 대한 확장 실험을 진행할 계획이다.

1) [15]의 UAS는 Macro지표로 Micro지표가 아니다. 실험에서 보다시피 Macro지표가 Micro지표보다 성능 수치가 다소 높다.

참고문헌

[1] McDonald, R., & Pereira, F. (2006). Online Learning of Approximate Dependency Parsing Algorithms. EACL, 81-88.

[2] Mcdonald, R., Pereira, F., Ribarov, K., & Hajic, J. (2005). Non-projective Dependency Parsing using Spanning Tree Algorithms. EMNLP.

[3] McDonald, R., Lerman, K., & Pereira, F. (2006). Multilingual dependency analysis with a two-stage discriminative parser. CoNLL

[4] Carreras X., Collins M., Koo T., "TAG, Dynamic Programming, and the Perceptron for Efficient, Feature-rich Parsing," EMNLP-CoNLL '07, 957-961

[5] Carreras X. 2007. Experiments with a higher-order projective dependency parser. In Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, pages 957--961, Prague, Czech Republic, June. Association for Computational Linguistics.

[6] Koo T., Collins M. Efficient third-order dependency parsers, ACL 2010, pp. 1-11

[7] Ma. X, Zhao H. Fourth-order dependency parsing, COLING '12785-796

[8] Nivre. (2003). An Efficient Algorithm for Projective Dependency Parsing IWPT. IWPT.

[9] Nivre, J. (2008). Algorithms for Deterministic Incremental Dependency Parsing. Computational Linguistics, (May 2007).

[10] Nivre, J. (2011). Non-projective dependency parsing in expected linear time. ACL '09

[11] Mcdonald, R., & Nivre, J. (2011). Analyzing and Integrating Dependency Parsers. Computational Linguistics.

[12] Zhang Y. & Clark S. (2008) A Tale of Two Parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search, EMNLP '08, 562-571

[13] 이용훈, 이종혁, "온라인 학습을 이용한 한국어 구문 분석," 한국컴퓨터종합학술대회논문집, 37(1), 299-303, 2010

[14] Lee Y.-H., Jin M., Lee J.-H.: Graph-Based Dependency Parsing Using Dynamic Features in Korean. Int. J. Comput. Proc. Oriental Lang. 23(2): 185-199 (2011)

[15] 임수종, 김영태, 나동열, "자질 가중치의 기계 학습에 기반한 한국어 의존 파싱," 정보과학회논문지: 소프트웨어 및 응용, 38(4), 600-608, 2011

[16] 박영민, 서정연, "투사성과 재탐색을 이용한 결정적 한국어 의존구조 분석의 보정기법," 인지과학, 22(4), 429-447, 2011