오류 분석을 통한 파서의 성능향상

오진영○ 차정원
창원대학교 컴퓨터공학과
psyche.ojy@gmail.com, jcha@changwon.ac.kr

Performance Improvement of Parser through Error Analysts

Jin-young Oh○ Jeong-Won Cha
Dept. of Computer Engineering, Changwon National University

요 약

본 논문에서는 무제한 텍스트 입력이 가능한 파서에서 오류분석을 통한 성능 향상을 이루고자 한다. 우선 코퍼스로부터 자동학습에 의해서 구문 분석 모델을 만들고 이를 평가하여 발생하는 오류를 분석한다. 오류를 감소시킬 수 있는 언어 특성이 반영된 자질을 추가하여 성능을 향상시키고자 한다. 세종 코퍼스를 10-fold cross validation으로 평가할 때, 한국어의 특성을 반영한 자질 추가로 1%이상의 성능 향상을 이루었다.

주제어 한국어 의존구조 분석, 다단계 구단위화, CRFs

1. 서 론

품사 태깅과 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 이용하면 좀 더 정확한 추출을 할 수 있다.

이 경우에 구문분석기에 반드시 필요한 것이 처리 속도, 안정성, 그리고 성능이다. 품사 태깅과는 달리 구문분석은 각 문장 성분들 간의 관계를 조사하기 위해서 일반적으로 $O(n^3)$의 처리 시간이 소요된다. 문장의 길이가 늘어날수록 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 종료하는 경우가 많이 발생한다. 성능은 모든 시스템에서 중요하지만 특히 구문분석에서는 구문분석의 오류가 응용 프로그램에 직접 영향을 미치기 때문에 중요하다.

영어권에서는 구문분석 연구가 성숙하여 구문분석 프로그램을 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 언어분석 도구가 줄어들어 연구의 깊이와 결과가 좋지 못한 상황이다.

본 연구에서는 무제한 텍스트 환경의 구문분석 시스템의 성능 향상에 대해 기술한다. 본 논문에서는 결과에 대해 정확한 오류분석을 통해 자질을 변경하여 성능향상을 도모하였다.

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

2. 관련 연구

영어권에서는 오래 전부터 많은 연구가 진행되었다. 최근의 연구는 CFG(Context Free Grammar)를 사용하고 통계 모델을 이용한 방법과 기계학습을 이용한 방법이 주류를 이루고 있다[1,2,3,4]. 또한 재순위화(reranking)를 통해 성능 향상시킨 방법도 제안[5]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)[6], 결정 트리(Decision Tree)[7], 최대 엔트로피 모델(Maximum

Entropy Model)[8,9], 지지 기반 기계(Support Vector Machine)[10] 등이다.

한국어에서도 다양한 시도가 있었다. 한국어 구문분석은 단일화 문법(Unification Grammar), 핵심어 중심 구 구조 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[11,12,13,14,15]. 최근에는 거의 모든 연구가 의존 문법을 기반으로 하고 있다. 또한 일본어와 마찬가지로 의존 구조의 애매성을 해소하기 위해 다양한 통계 방법과 기계학습을 이용하는 방법들이 제안되었다[16,17]. 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[16,17].

본 연구에서는 세종 구문 코퍼스[18]를 사용하여 학습 및 평가를 하며 기존 방법보다 간단하지만 효율적인 한국어 구문분석 방법의 성능 향상 방법을 제안한다. 제안된 시스템은 다단계 구 단위화를 통해 각 단계에서 지배소를 결정하는 방법을 취한다.

3. 제안 시스템

본 논문은 [19]에서 설명한 시스템과 동일한 시스템을 사용한다. 입력 문장은 먼저 품사 태깅[20] 과정을 거치고 다음에 어절 단위 구문태그를 부착하기 위한 과정이 있다. 여기서 나온 결과를 다단계 구 단위를 통해 구문분석을 한다.

본 논문에서 제안한 시스템의 구조를 그림 1에서 보인다.

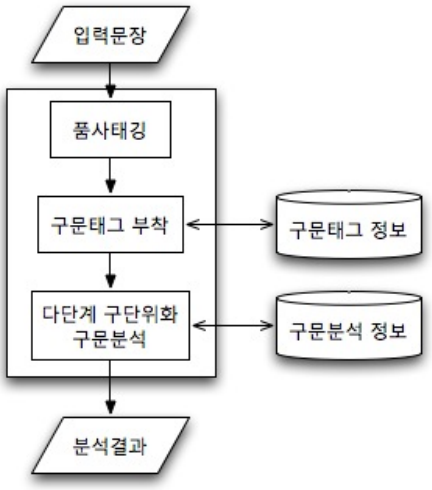

그림 1. 제안 시스템 구조도

3.1 다단계 구 단위화 방법

다단계 구 단위화(Cascaded Chunking) 방법은 [21]에서 영어를 위해 처음 제안되었다. 이 방법은 [22, 23]에 의해 일본어에 적용되어 좋은 성능을 보였다.

본 연구에서는 한국어의 특성에 맞게 이를 변형하여 적용한다. 그림 2는 다단계 구 단위화의 과정을 보여준다.

어절	1단계	2단계	3단계	4단계	5단계
자기에	-	D	X	X	X
충실치	D	X	X	X	X
못하고는	-	-	-	D	X
도덕이	-	-	D	X	X
생겨날	D	X	X	X	X
수	D	D	X	X	X
없다.	-	-	-	-	-

그림 2. 다단계 구 단위화를 이용한 구무분석의 예

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.

'X' 표시는 앞의 단계의 의존소를 제외한 어절로서, 학습코퍼스를 생성할 때 해당 단계에서 삭제된다. 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.

3.2 자질 집합

구문 분석은 형태소 단위가 아닌 띄어쓰기로 구분된 어절 단위로서 분석하게 된다. 따라서 구문태그와 의존관계에 대해서 기계학습한 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 구문 분석에 사용한 자질 예이다.

어절 번호	형태소 분석	1	2	3	4	지배 어절
1	물론/MAG	-	MAG	-	AP	6
2	스포츠/NNG +에/JKB	-	JKB	있/VV	NP_AJT	3
3	있/VV+어서 /EC+도/JX	EC	JX	-	VP	6
4	이것/NP +은/JX	-	JX	-	NP_SBJ	6
5	예외/NNG +가/JKC	-	JKC	아니 /VCN	NP_CMP	6
6	아니/VCN +다/EF+./SF	-	-	-	VP	-

그림 3. 구문분석 학습 코퍼스 예

구문분석은 4개의 자질을 사용하였으며, 자질들을 생성함에 있어서 기호에 대한 품사는 추가하지 않았다.

1은 현재 어절의 마지막 앞의 자질이고 2는 어절 마지막 자질이다. 3번째 자질은 다음어절 첫 번째 형태소에 대한 형태소와 품사로서 이루어진 자질이다. 다음 어절의 첫 번째 형태소가 'V'로 시작하는 경우 3번 자질에 추가하는 것으로서, 조용사(XSA, XSV)가 붙어서 형용사, 또는 동사가 되는 경우에도 3번 자질에 추가하였다. 예를 들어 다음 어절이 '입장/NNG+하/XSV+ㄴ다/EF+./SF'일 경우 이전 어절의 3번 자질에는 '입장하/VV'가 추가된다. 4번째 자질은 구문태그 결과를 자질로 사용한다. 구문태그 결과만을 자질로 사용하였을 경우 구문태그 자질의 오류에 대해서 구문분석 예측 확률이 낮아져서 품사에 대한 자질을 추가하였다.

4. 실험 및 분석

4.1 실험환경

세종 코퍼스 58,175문장을 10-fold cross validation을 수행하였다. 한 문장의 평균 길이는 10.97이다.

제안한 시스템의 성능 평가를 위해 아크-정확도와 아크-재현율을 결합한 $F_1-measure$와 문장 정확도(Exact-Matching)를 사용하였다. 평가 척도는 식 (1)과 같다. 본 연구에서는 구문 분석을 레이블링 문제로 해결하기 때문에 아크-정확도와 아크-재현율이 같다.

$$\begin{aligned} &\text{아크}-\text{정확도}(Arc\,\mathrm{P}recision, AP) \\ &= \frac{\text{구문 분석 파스트리에서 올바른 아크의 수}}{\text{구문 분석 파스트리에서 모든 아크의 수}}, \\ &\text{아크}-\text{재현율}(Arc\,Recall, AR) \\ &= \frac{\text{구문 분석 파스트리에서 올바른 아크의 수}}{\text{정답 파스트리에서 모든 아크의 수}}, \end{aligned} \quad (1)$$

$$F_1-measure = \frac{2 \bullet AP \bullet AR}{AP+AR},$$

$$Exact-Matching = \frac{\text{정확히 분석된 문장의 수}}{\text{문장의 수}}$$

4.2 기본 실험

10-fold cross validation을 수행하여 기본 실험을 수행하였다. 그 결과는 표 1에 나타나 있다.

표 1. 기본 실험 결과

	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

기본 실험의 오류를 분석해 보면 다음과 같았다. 먼저 콤마(,)의 경우는 대등연결 기능(나열 기능)과 종속절 표현 등으로 다양하게 사용된다. 따라서 콤마에 따라서 지배소와 피지배소에 대한 오류가 많이 발생하였다.

두 번째는 그림 4와 같은 보조용언에 대한 오류가 많이 나타났다.

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

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.

세 번째는 조사가 없는 명사 연속 어절에 대한 오류가 많았다.

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

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.

네 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.

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

4.3 언어 자질 추가를 통한 오류 해결

콤마에 관한 오류를 해결하기 위해 자질에 콤마의 형태소를 추가하였다. 그림 3에서 '물론/MAG' 어절이 '물론/MAG+,/SF'일 경우 1번 자질은 '-', 2번 자질은 'MAG', 3번 자질은 'SF', 4번 자질은 'AF'이 된다.

표 2는 표 1에서 콤마 자질을 추가한 결과이다.

표 2. 콤마 자질 추가 실험 결과

	어절	문장
1	0.8185	0.3090
2	0.8801	0.5760
3	0.8791	0.5730
4	0.8821	0.5510
5	0.8534	0.4470
6	0.8477	0.4630
7	0.8419	0.4620
8	0.8389	0.3500
9	0.8384	0.3180
10	0.8479	0.3370
평균	0.85279	0.4386

실험 결과를 보면 콤마가 문장 구조에 영향을 주는 것을 알 수 있다. 따라서 문장 구조에 영향을 줄 수 있는 기호, 조사 등에 대한 추가적인 자질 연구가 필요하다.

그리고 보조용언 오류, 조사가 없는 연속된 명사어절의 오류를 해결하기 위해서 구문분석에서 사용하는 자질을 변경하였다. 그림 3에서 1번은 현재 어절의 마지막 앞의 형태소 자질이다. 1번 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 1번, 2번 자질에 같은 형태소가 추가된다. 1번 자질과 2번 자질이 동일할 때 동시에 사용할 것인지 하나를 사용할 것인지를 결정하기 위해서 3가지 추가 실험을 하였다.

표 3은 3가지 실험 중 1번 자질에만 형태소 자질을 추가한 실험과 2번 자질에만 추가한 실험 결과이다.

표 3. 자질 추가 실험 결과

	1번 자질에만 추가		2번 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 4는 1번과 2번 자질에 동시에 추가했을 때의 실험 결과이다.

표 4. 1번 자질과 2번 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

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.

5. 결론

본 논문에서는 한국어 무제한 텍스트 입력을 처리할 수 있는 구문 분석기의 성능향상을 위해서 언어 특성을 반영한 자질에 대해서 기술하였다. 오류분석을 통한 언어자질을 추가함으로써 기 개발된 시스템의 성능향상을 이룰 수 있었다.

이 결과를 근거로 하여 추가적인 자질 연구를 할 예정이며 용언의 하위분류를 추가하여 추가적인 성능향상과 상위 응용에 대응할 수 있는 시스템을 구현할 예정이다. 또한 세종 코퍼스가 아닌 KIB(Korean Language Information Base)[24] 코퍼스를 사용하여 평가해 일반성을 확보할 예정이다.

참고문헌

[1] 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.

[2] Dan Klein and Christopher D. Manning. 2003. “Accurate Unlexicalized Parsing.”, ACL, pp. 423-430, 2003.

[3] Charniak, E. “A Maximum-Entropy-Inspired Parse.”, NAACL, pp 132-139, 2000.

[4] Slav Petrov and Dan Klein, "Improved Inference for Unlexicalized Parsing.", HLT-NAACL, pp. 404-411, 2007.

[5] Eugene Charniak and Mark Johnson. "Coarse-to-fine n-best parsing and MaxEnt discriminative reranking.", ACL, pp. 173-180, 2005.

[6] Masakazu Fujio and Yuji Matsumoto. "Japanese Dependency Structure Analysis based on Lexicalized Statistics.", EMNLP, pp. 87-96, 1998.

[7] Msahiko Haruno, Satoshi Shirai, and Yoshifumi Ooyama. "Using Decision Trees to Construct a Practical Parser.", Machine Learning, 34:131 - 149, 1999.

[8] Kiyotaka Uchimoto, Satoshi Sekine, and Hitoshi Isahara. "Japanese Dependency Structure Analysis Based on Maximum Entropy Models.", EACL, pp. 196 - 203, 1999.

[9] Kiyotaka Uchimoto, Masaki Murata, Satoshi Sekine, and Hitoshi Isahara. Dependency model using posterior context. In Procedings of Sixth International Workshop on Parsing Technologies. 2000.

[10] 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.

[11] Geum, J. C., 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.

[12] 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.

[13] Yang, J., "A study on the Korean analyzer based on HPSG.", Master's thesis, Dept. of Computer Engineering. Seoul National University, 1990.

[14] 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.

[15] Jeongwon Cha, Geunbae Lee, Jong-Hyeok Lee, Morpho-syntactic categorial modeling of Korean, computers and the humanities journal, vol 36, No. 4, page 431-453, 2002.

[16] Hoojung Chung, "Statistical Korean Dependency Parsing Model based on the surface Contextual Information", Ph.D. dissertation, 2004.

[17] Yong-Hun Lee, Jong-Hyeok Lee, "Korean Parsing using Machine Learning Techniques", KCC 2008, pp. 285-288, 2008

[18] 세종계획 21, http://www.sejong.or.kr/

[19] 오진영, 차정원, “CRFs를 이용한 강건한 한국어 의존구조 분석”, 제 20회 한글 및 한국어 정보처리 학술발표논문집, pp. 23-28, 2008

[20] 홍진표, 차정원, “어절패턴 사전을 이용한 새로운 한국어 형태소 분석기”, 한국 컴퓨터 종합 학술대회 논문집 제 35권, pp. 279-284, 2008

[21] Steven Abney, "Parsing By Chunking.", In Principle-Based Parsing. Kluwer Academic Publishers, 1991.

[22] Kudo, T., Y. Matsumoto., "Japanese Dependency Analysis using cascaded Chunking.", COLING, pp. 63-39 , 2002.

[23] Zhou, H., T. Yu, et al., "Japanese Dependency Analysis Based on SVMs and CRFs.", INTERNATIONAL JOURNAL of MATHEMATICS AND COMPUTERS IN SIMULATION, 1(3): 233-237. 2007.

[24] 국어정보베이스, http://kibs.kaist.ac.kr/