고성능 비어휘정보 한국어 구문분석

오진영 ○ 차정원

창원대학교 컴퓨터공학과

psyche.ojy@gamil.com, jcha@changwon.ac.kr

Accurate Unlexicalized Korean Parsing

Jin-young Oh○ Jeong-Won Cha

Dept. of Computer Engineering, Changwon National University

요 약

본 논문에서는 어휘정보를 사용하는 한국어 구문분석 성능과 거의 비슷한 성능을 내는 비어휘정보 한국어 의존 구문분석에 대해서 설명한다. 본 논문에서는 어휘정보를 대신해서 품사정보와 어절 구문태그 정보를 사용하고 CRFs를 사용하여 레이블링 방법으로 구문분석 한다. 자질을 변경하여 어절 처음에 나타나는 용어 정보와 뒤 어절의 용언 정보를 추가하였다. 본 논문에서 제시하는 실험 결과(어절:85.73%, 문장:43.86%)는 현재 최고의 성능을 내는 어휘정보 사용 한국어 구문분석과 비슷하다. 본 논문에서 제안한 비어휘정보 구문분석 방법은 어휘정보 구문분석에 비해 모델 사이즈가 작고 처리방법이 간단하여 쉽게 다른 도메인에 적용이 가능할 것으로 기대한다.

1. 서 론

구문분석은 문장의 구조를 분석하는 것으로, 오래 전부터 연구되어 왔던 분야이다. 형태소 분석과 품사 태깅을 거쳐 구문 분석이 수행하게 되는 것이 일반적이다. 문장 구조가 밝혀지면 문장의 의미를 파악하는데 절대적인 영향을 미친다.

또한 구문 분석의 결과를 바탕으로 다양한 자연어처리 응용분야에서 활용하고자 하는 요구가 증가하고 있다. 정보추출에서는 단순히 lexico-syntactic pattern을 사용하지 않고 구문분석 결과를 사용한다면 보다 유연하게 규칙을 적용할 수 있어 성능향상에 도움이 될 것이다. 의견 마이닝에서도 구문분석 결과가 성능향상에 도움이 된다. 현재 의견 마이닝은 어휘정보, 품사정보, n-gram 정보를 사용한다[1]. 어휘간의 관계를 정확하게 알 수 있다면 의견을 파악하는데 더 유용할 것이다. 자연어 인터페이스에서도 구문분석은 문장의 의미를 명확하게 하는데 결정적인 역할을 한다.

그러나 현재까지 제안된 한국어 구문분석 방법은 어휘간의 통계 정보를 사용하여 구문 애매성을 해소하기 때문에 데이터 베이스가 커진다. 또한 어휘에 의존적이기 때문에 도메인에 의존적이다.

본 연구에서는 성능을 확보하기 위해서 어휘정보를 사용하는 방법에 대한 해결방법을 제시한다. 즉, 성능을 유지하면서 어휘정보 사용으로 인한 데이터베이스의 비대화를 막아 실용적인 구문분석 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장은 관련된 연구에 대해서 살펴보고 3장은 본 논문에서 제안하는 방법 및 시스템을 설명하며, 4장은 실험에 대해서 설명한다. 5~8장에서는 실험한 각 부분에 대해서 설명한다. 마지막으로 9장에서 결론을 맺는다.

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[2,3,4,5]. 본 논문에서는 어휘정보를 사용하지 않는 [3,5]와 같은 방법을 심도있게 분석했다. 본 논문에 또한 reranking을 통해 성능 향상시킨 방법도 제안[6]되었으며, 영어권에서 개발된 많은 방법들이 일본어와 한국어에 적용되었다. 그렇지만 한국어와 영어는 언어적 특성이 많이 달라서 그 방법을 그대로 적용할 경우 성능에 심각한 저하가 발생한다.

일본어에서는 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)[7], 결정 트리(Decision Tree)[8], 최대 엔트로피 모델(Maximum Entropy Model)[9,10], 지지 기반 기계(Support Vector Machine)[11] 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[12,13,14,15,16]. 최근에는 거의 모든 연구가 의존 문법을 기반으로 하고 있다. 또한 의존 구조의 애매성을 해소하기 위해 다양한 통계 방법과 기계학습을 이용하는 방법들이 제안되었다[17,18]. 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]. 그렇지만 성능과 속도 그리고 데이터 베이스의 크기 등의 문제로 인해 응용프로그램에서 사용할 수 있는 실용적인 방법과는 아직 거리가 있다.

3. 제안 시스템

본 논문에서는 입력 문장을 품사 태깅, 구문태그 부착 그리고 구문 분석한다. 이 과정을 그림 1과 같다[18].

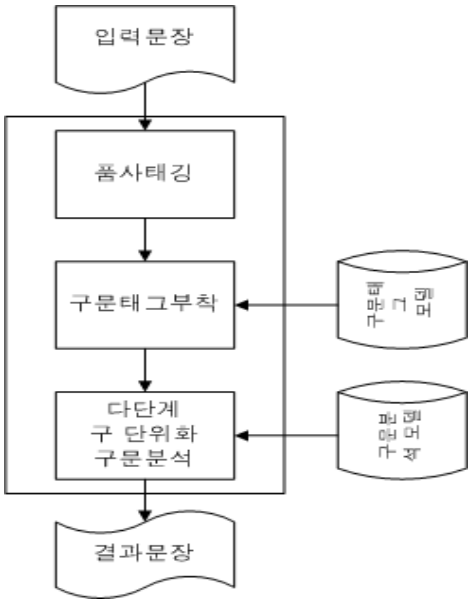

그림 1. 시스템 구조도

3.1 다단계 구 단위화 방법

본 논문에서는 다단계 구 단위화를 사용하여 구문 분석한다. 다단계 구 단위화(Cascaded Chunking) 방법은 [19]에서 영어를 위해 처음 제안되었다. 이 방법은 [20, 21]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 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.

다단계 일반적인 구문분석 방법의 시간 복잡도가 $O(n^3)$임에 비하여 구 단위화 기법은 $O(n^2)$이므로 매우 빠르다. 실제 본 연구에서 제안한 시스템은 평균 15어절의 문장을 초당 100문장 이상 분석할 수 있다. 또한 구문 요소의 결합이 아니라 레이블링 문제이므로 입력 문장에 대해서 매우 강건하다.

4. 실험

실험에서 사용된 코퍼스는 어절태그를 학습하기 위해서 9,889문장(90,112어절)을 사용하였고, 의존관계를 위해 1,120,777어절(96,412문장)을 사용하였다. 평가 코퍼스는 20,685어절(1,430문장)을 사용하였고, 한 문장당 평균 14.59어절로서 구성되어 있다.

실험을 위해서 5-fold cross validation을 실시하여 그 값을 평균하여 평가하였다.

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 동일한 어절에 대해서 지배소 표시가 정확한지를 검사한다. 따라서 아크-정확도와 아크-재현율이 같다.

$$\text{아크정확도(Arc Precision, AP)} = \frac{\text{구문분석에서정확한아크의수}}{\text{구문분석에서모든아크의수}}$$
$$\text{아크재현율(Arc Recall, AR)} = \frac{\text{구문분석에서정확한아크의수}}{\text{정답구문분석에서모든아크의수}} \quad (1)$$

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

$$Exact - Matching = \frac{\text{정확하게분석된문장수}}{\text{문장수}}$$

	모델크기	어절	문장	차이
LEX	52MB	0.8554	0.4330	
UNLEX 1	4.1MB	0.8319	0.4024	-0.0235
UNLEX 2	5.7MB	0.8562	0.4356	+0.0008
UNLEX 3	5.6MB	0.8573	0.4386	+0.0019
UNLEX 4	5.6MB	0.8561	0.4348	+0.0007

그림 2. 모델은 CRFs 학습된 모델의 크기를 나타낸다. 어절과 문장은 각각 어절 단위와 문장 단위 F_1-measure를 나타낸다. 차이는 각 결과를 LEX와 비교한 값이다.

5. 어휘정보를 이용한 방법

본 연구에서는 세종 구문 코퍼스[22]를 사용하였다. 구문 분석은 형태소 단위가 아닌 띄어쓰기로 구분된 어절 단위로서 분석한다.

그림 2에서 LEX는 어휘정보를 사용하여 실험한 결과이다. 이 경우에 사용한 자질 정보는 그림 3과 같다.

형태소 분석	1	2	3	구문태그
물론/MAG	MAG	-	-	AP
스포츠/NNG +에/JKB	NNG	JKB	있/VV	NP_AJT
있/VV+어서/EC+도/JX	VV	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. 자질 정보.

구문 분석에는 4개의 자질을 사용하였으며, 자질들을 생성함에 있어서 기호에 대한 품사는 추가하지 않았다. 1은 현재 어절의 첫 번째 형태소의 품사 자질이다. 2는 마지막 품사에 대한 자질이다.

그리고 3번째 자질은 다음어절 첫 번째 형태소에 대한 형태소와 품사로서 이루어진 자질이다. 다음 어절의 첫 번째 형태소가 'V'로 시작하는 경우 5번 자질에 추가하였으며, 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.

6. 조사정보 제거

그림 2에서 UNLEX 1은 그림 3의 자질 2번을 제거한 결과이다. 구문태그를 결정한 때 이미 조사 정보를 사용하기 때문에 그 정보가 구문태그에 포함되었을 것이라고 가정하고 제거하였다. 그러나 실험 결과에서 보듯이 성능이 저하되었다. 이것은 구문태그로 합쳐진 정보보다 좀 더 세분화된 조사/어미 정보가 구문 분석에 도움이 된다는 것을 말한다. 즉, 동일한 구문태그를 가지더라도 조사/어미 품사 정보에 따라 구문 분석이 달라짐을 의미한다.

7. 어휘정보 제거

그림 2에서 UNLEX 2는 그림 3의 자질 3번에서 어휘정보를 제거한 결과이다. 3번 자질은 다음 어절의 첫 형태소가 동사/형용사류일 경우에 해당 정보를 추가하는 것이다. LEX의 결과에서 보듯이 이 자질의 어휘정보가 모델을 크게 한다. 그런데 같은 품사의 특정 어휘정보가 특정 문장 성분을 제약할 것이라는 가정하에 어휘정보를 사용하였지만 실험 결과에서 보듯이 어휘정보는 구문 분석 성능 향상에 도움이 되지 않았다. 이것은 어휘가 중요한 것이 아니라 그 어휘의 품사가 구문 분석에 중요함을 나타내는 것이다.

8. 자질 변경

그림 2에서 UNLEX 3는 그림 3의 1번 자질을 변경한 것이다. 결과 분석을 통해서 우리는 다음과 같은 경우가 에러 발생이 많이 됨을 발견하였다.

가) 설명/NNG+ 하/XSV+ 는/ETM VP_MOD

이 어절은 명사와 조용사가 결합하여 용언이 되는 것인데 자질 1번이 어절의 첫 형태소만을 사용하기 때문에 어절의 구문태그가 VP가 되는 것을 설명하는데 방해가 된다. 따라서 이 자질을 생성할 때 NNG|XR + XSV|XSA가[1] 될 경우 VV|VA로 되도록 하였다.

그림 2에서 UNLEX 4는 그림 3의 3번 자질이 기존에는 VA와 VV를 구분하였는데 이것을 모두 VV로 통일한 실험이다. 즉 동사와 형용사의 구별없이 용언임을 나타내도록 자질을 구성한 것이다.

결과를 분석해보면 그림 3의 1번 자질에 용언 정보를

[1] '|'는 둘 다 되는 경우를 말한다.

반영한 것이 성능이 높았으며, 3번 자질에서는 동사와 형용사를 구별한 것이 성능이 좋았다. 이것은 동사와 형용사의 구문 구조가 서로 다르게 나타난다는 것을 의미한다.

9. 결론

본 논문에서는 기 제안된 다단계 구단위화를 이용한 한국어 의존구조 분석 방법에서 어휘정보를 제거하여 도메인 의존성을 낮춘 시스템을 제안하였다.

어휘정보는 거의 모든 언어처리 시스템에서 성능향상을 위해서 사용하고 있는데 이것은 시스템의 메모리 사용을 크게 하고 알고리즘을 복잡하게 만든다. 또한 도메인에 의존성이 높은 시스템이 되게 만든다. 본 논문에서는 어휘정보를 제거하였음에도 불구하고 성능이 오히려 어휘정보를 사용했을 때보다도 우수한 실험 결과를 보임으로써 시스템을 도메인 독립이 되도록 구현할 수 있게 하였다.

향후에는 동일한 품사나 구문태그에 대해서 다른 구문 분석 결과를 나타내는 기능어나 기호 등의 정보를 자질에 추가하여 성능을 향상시킬 예정이다.

참고문헌

[1] Mainqing Hu and Bing Liu. 2004. “Mining and Summarizing Customer Reviews.” In Proceedings of KDD. Seattle, Washington, USA, pp 168-177.

[2] 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.

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

[4] Charniak, E. “A Maximum-Entropy-Inspired Parse.” in Proceedings of NAACL-2000, pp 132--139.

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

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

[7] Masakazu Fujio and Yuji Matsumoto. "Japanese Dependency Structure Analysis based on Lexicalized Statistics." In Proceedings of EMNLP '98, 1998, pp. 87-96.

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

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

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

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

[19] 오진영,차정원, “다단계 구단위화를 이용한 고속 한국어 의존구조 분석”, 시뮬레이션 학회 논문집, 2010.

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

[21] Kudo, T. and Y. Matsumoto. "Japanese Dependency Analysis using cascaded Chunking." coling02. 2002.

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