

한국어 파서에서의 지역 의존관계의 이용

류 범 모^o, 이 종 혁, 이 근 배

포항공과대학교 전자계산학과

Using Local Dependency for Dependency Parser of Korean

Pummo Ryu^o, Jong-Hyeok Lee, Geunbae Lee

Dept. of Computer Science and Engineering, POSTECH

요 약

본 논문에서는 한국어 의존관계 파서의 정확성 및 효율성을 높이기 위해 구조 내의 지역적 수식 특성을 반영할 수 있는 지역 의존관계의 사용을 제안한다. 의존문법은 자유어순 언어를 잘 설명할 수 있는 장점이 있지만, 전체 문장 구조에 관한 의존제약이 너무 미약하기 때문에 단순히 어절간 구문 의존 제약 만으로는 원하지 않는 분석 결과가 너무 많이 생성된다. 그러나 자유어순 언어라 하더라도 지역적인 구구조에는 일정한 어순 제약이 존재한다. 명사구, 용언구 등과 같은 구구조를 분석해 보면 수식어의 지배소는 반드시 그 구 안에 있다. 이러한 구조 정보에 기반을 둔 지역 의존관계 규칙을 이용하면 하나의 의존소에 대해서 지배소로 사용될 수 있는 어절의 범위를 제한하여, 원하지 않는 분석 결과를 줄일 수 있다. 한국어는 기본 문장 구조가 그대로 사용되기보다는 하나 이상의 수식 요소들이 첨가되어 보다 긴 문장 구조로 사용되는 경우가 많기 때문에, 본 논문에서 제안한 방법은 시스템 전체의 성능 및 효율을 크게 향상시킬 수 있다. 실험에서는 파싱의 첫 번째 단계에서 지역 의존관계 규칙을 사용하였을 경우 사용하지 않았을 때에 비해서 의존관계의 수가 평균 69% 정도로 줄어들었다.

1. 서론

의존문법은 어절들의 순서 관계보다 어절 사이의 의존관계를 고려하며 문장 구조를 파악하기 때문에 한국어와 같이 비교적 어순이 자유롭고 비구성적 언어의 분석에 적합하다[7]. 그러나 기존에 개발된 한국어 의존관계 파서는 한국어의 자유어순 특성을 너무 강조한 나머지, 지역적 어순 제약에 대한 고려를 거의 하지 않았다. 어순이 자유롭다는 것은 영어나 프랑스어와 같은 언어와 비교해서 자유롭다는 것이지, 절대적인 자유어순을 뜻하는 것은 아니다. 한국어에는 분명히 지역적 어순 제약이 존재하고, 그 어순 제약에 따라서 지역적인 구조를 기술할 수 있다. 이러한 구조 정보를 이용하면 불필요한 의존관계 생성을 억제할 수 있다. 명사구, 용언구의 지역적 수식 특성을 분석하면 수식어의 지배소는 반드시 그 구 안에 있다는 것을 알 수 있다. 기존에 한 어절의 지배소를 찾기 위해서 문장의 끝까지 모두 고려한 것에 비하여, 인접한

위치에서 원하는 의존관계를 찾을 수 있으면 불필요한 의존관계가 줄어들게 되므로, 문장 단위의 구문적 애매성 해소가 훨씬 쉬워진다.

본 논문에서는 기존의 2 단계 한국어 의존관계 파서 (TPDP-K:Two-Phase Dependency Parser for Korean)[10]에서 어절 간 의존관계 설정 단계를 보완하기 위하여 지역 의존관계 규칙(local dependency rule)을 제안한다. 기존의 2 단계 한국어 의존관계 파서는 첫 번째 단계에서 각 어절에 대해서 다른 어절들과 모든 가능한 구문적 의존관계를 설정하여 구문 의존관계 그래프 (syntactic dependency graph)에 표시한 후, 두 번째 단계에서 구조적 중의성을 제약 전파와 백트래킹을 혼합한 Forward Checking (FC) 알고리즘과 술어 중심의 제약 전파 알고리즘[4]을 이용하여 해소한다. 그러나 첫 번째 단계에서 바람직하지 않는 의존관계가 너무 많이 생성되기 때문에, 두 번째 단계에서 원하지 않는 파싱결과가 엄청나게 증가하게 된다. 의존관계의 수가 선형적으로 증가할 때 파싱결과가 지수배로 증가하기

때문에, 첫 번째 단계에서 바람직하지 않는 의존관계의 수를 줄이는 것이 매우 중요하다. 제안한 방법은 의존관계 설정 단계에서 명사구, 용언구 내의 수식어에 대해서는 지역 의존관계 규칙을 이용하여 지배소로 사용될 수 있는 어절의 범위를 제한하여 후보 지배소 관계를 설정하고 나머지 어절에 대해서는 일반 의존관계 규칙을 사용하여 후보 지배소를 결정한다. 한국어는 기본 문장이 그대로 사용되는 경우가 드물고, 하나 이상의 수식어가 첨가되어 보다 긴 문장 구조로 사용되는 경우가 많기 때문에, 지역 의존관계 규칙을 이용한 의존관계를 설정은 전체 시스템의 정확성 및 효율성을 크게 향상시킬 수 있다[1].

2 장에서는 관련된 연구를 살펴보고, 3 장에서는 한국어 구구조의 수식 특성에 대해서, 4 장에서는 지역 의존관계의 정의, 규칙 및 알고리즘에 대해서, 5 장에서는 2 단계 한국어 의존파서에 대해서, 6 장에서는 실험 및 결과에 대해서 이야기하고 7 장에서 결론을 맺는다.

2. 관련된 연구

의존문법은 비구성적인 언어를 분석할 수 있는 장점이 있지만 구구조문법에 비해서 너무 자유롭다는 단점이 있다. 따라서 의존문법을 사용해서 문장을 분석할 때 이러한 단점을 보완하기 위하여, 문장의 지역적인 구조에 대한 고려가 여러 연구에서 지적되고 있다[3][7].

[5]에서는 문맥자유문법 형태의 한국어 이진문법을 사용하여, 여러 개의 어절을 구절 단위로 묶은 뒤 의존문법을 이용하여 파싱을 한다. 이 방법은 간단한 문법 규칙으로 파싱 단계의 부담을 줄이고 처리하기 쉬운 결과를 생성해 내는 장점은 있지만, 최종 분석 결과를 의존 트리라고 했을 때, 이진문법을 이용해서 묶은 부분을 다시 의존트리로 변환시켜야 한다는 단점이 있다.

[2]에서는 부분적인 어절 결합을 이용하여 분석이 어려운 문장을 분석이 쉬운 문장으로 변환시킬 수 있지만, 결합의 대상이 관용어, 보조용언, 속어 수준에 머무르고 있고, 결합된 부분 안에서의 구조에 대한 고려가 없다. 구문분석이 의미분석을 위한 전 단계라고 생각했을 때, 결합된 부분 내부에 대한 분석도 반드시 필요하다.

[9]에서는 예문을 기반으로 최장 묶음을 이용한 한국어 분석기를 제안했다. 이 연구에서는 자연언어 이해보다 기계번역을 위한 구문분석을 시도했다. 큰 구성단위로 묶은 후에 목적 언어로 대응시키기 때문에, 자연스러운 번역 결과를 얻을 수는 있지만, 대량의 예문 데이터베이스의 구축이 아주 어려운 일이다.

3. 한국어 구구조의 수식 특성

한국어 구구조에서 지역적 수식 특성을 찾아볼 수 있는 부분은 크게 명사구와 용언구 두 가지로 나눌 수 있다.

3.1 명사구

명사구는 하나의 체언에 한 개 이상의 수식어가 붙어 확대된 형태를 말한다. 문장에서 명사구의 역할은 가장 뒤의 체언(head)에 의해서 결정된다. 앞의 수식어는 의미적으로 한정 할 뿐, 전체 문장에서 문법적 기능에 영향을 주지는 않는다. 따라서 명사구 내에서 수식어의 지배소는 피수식어이고, 그 명사구 안에서만 결정된다. 명사구에서 수식어로 사용될 수 있는 체언 수식어구는 관형사, 체언의 절대격, 체언+관형격조사(의), 체언+적/네 접미사, 용언+관형형 전성어미 등이 있다. 하나의 체언 앞에 여러 개의 체언 수식어구가 비교적 자유로운 순서로 나타날 수 있다.

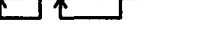
(1.1) 나는 중요한 모든 사회적 기반을 잃었다.



(1.2) 우리나라의 월드컵 개최는 자랑스러운 일이다.



(1.3) 나의 동생 영희를 보았다.

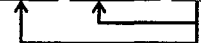


(1.1)에서는 “중요한”(용언+관형형 전성어미), “모든”(관형사), “사회적”(체언+적 접미사)이 “기반”을 수식하는 수식어로 사용되었고, (1.2)에서는 “우리나라의”(체언+관형격조사), “월드컵”(체언의 절대격)이 “개최”를 수식어로 사용되었다. (1.1)과 (1.2)에서 수식어의 지배소를 결정할 때는 각 명사구의 가장 뒤의 체언인 “기반을”과 “개최는”까지만 고려하면 된다. 또 명사구 내에서 구문 정보만을 사용하여 수식어와 피수식어를 정확히 찾는 것은 매우 어려운 일이다. (1.3)은 (1.2)와 같이 체언 “영희” 앞에 체언+관형격조사, 체언의 절대격이 수식어로 나타나지만, 의미적으로 볼 때 “나의”가 수식하는 어절이 (1.2)와 달리 바로 뒤 어절인 “동생”임을 알 수 있다. 따라서 의미분석 단계에서 두 가지 모두를 고려하여 한 가지를 선택할 수 있도록, 구문분석 단계에서는 가능한 두 가지 결과를 모두 보여 줄 수 있어야 한다.

3.2 용언구

용언구도 하나 이상의 용언 수식어구가 용언에 결합되어 이루어진다. 용언 수식어구는 부사, 용언+부사형 전성어미, 체언+부사격조사 등이 있다. 용언 수식어구의 지배소는 용언구의 마지막 어절이다.

(1.4) 나는 철수를 아침에 학교에서 만났다.



(1.5) 꽃이 활짝 피어 있는 공원을 산책했다.



(1.6) 너는 예쁘게 웃는다.



표 1. 지역 의존관계 규칙

의존관계	의존소	지배소
명사구 지배가능경로 의존관계	N, NAdn, Adn, AAdn, VAdn	N, NAdn, Adn, AAdn, VAdn
인접 의존관계	N, NAdn, Adn, AAdn, VAdn	NNom, NObj, NAdv, NAux, NCon
	NCon	N, NNom, NObj, NAdv, NAux, NCon
	Adv	VTer, VAdn, VCon, VAdv, ATer, AAdn, ACon, AAdv, Adv
	NAdv	VTer, VCon, VAdv
	NObj	VTer, VAdn, VCon, VAdn

(1.4)에서는 체언+부사격조사 “아침에”와 “학교에서”가 “만났다”를 수식한다. (1.5)에서는 부사 “활짝”이 “피어”를 수식하고, (1.6)에서는 용언+부사형 전성어미 “예쁘게”가 “웃는구나”를 수식한다. (1.5)에서 “활짝”의 지배소로 사용될 수 있는 어절은 “피어”로 확인할 수 있기 때문에 “산책했다”는 고려하지 않아도 된다.

4. 지역 의존관계

지역 의존관계는 인접 의존관계 규칙(adjacency dependency rule)과 지배가능경로(headable path)[6]를 이용한 명사구 지배가능경로 의존관계 규칙(noun phrase headable path dependency rule)을 사용하여 설정한다. 인접한 두 어절 사이에 명사구 지배가능경로 의존관계

1. 문장 $S = W_1 \dots W_{n-1} W_n$ 에서 W_{n-1} 에서 W_1 의 순서로 각 W_k 에 대해서 과정 2를 실행한다.
2. W_k 와 W_{k+1} 의 모든 가능한 형태소 분석결과 쌍 $(M_{k,i}, M_{k+1,j})$ 에 대해
 - 2.1 $(M_{k,i}, M_{k+1,j})$ 사이에서 명사구 지배가능경로 의존관계 규칙이 성립하면
명사구 지배가능경로 의존관계 설정
 - 2.2 $(M_{k,i}, M_{k+1,j})$ 사이에서 인접의존관계 규칙이 성립하면
인접 의존관계 설정

그림 1. 지역의존관계 규칙 알고리즘

규칙이 적용될 경우 인접한 두 어절 사이 뿐만 아니라, 뒤 어절의 후보 지배소들과 앞 어절 사이에도 의존관계 설정을 고려하여 후보 지배소로 설정한다. 인접 의존관계 규칙만 적용될 경우 뒤 어절을 앞 어절의 유일한 지배소로 설정한다. 인접 의존관계는 명사구, 용언구의 수식 구조 뿐만 아니라 명사가 접속조사에 의해서 나열되어 있는 문형이나 목적어와 타동사가 인접해 있는 경우에도 성립된다.

문장($S = W_1 \dots W_{n-1} W_n$)에서 각 어절 W_i 가 형태소분석 결과 $M_{i,0} \dots M_{i,m}$ 을 가질 때 아래와 같이 인접 의존관계와 명사구 지배가능경로 의존관계를 정의할 수 있다.

정의 1. 연속되는 두 어절 $W_i W_{i+1}$ 에서 각각의 형태소 분석 결과 쌍 $(M_{i,j}, M_{i+1,k})$ 에서 $M_{i+1,k}$ 가 $M_{i,j}$ 의 유일한 지배소라고 확인할 수 있을 때 $M_{i,j}$ 와 $M_{i+1,k}$ 사이에 의존관계를 설정하고 이를 인접 의존관계라고 한다.

정의 2. 연속되는 두 어절 $W_i W_{i+1}$ 의 각각의 형태소 분석 결과 쌍 $(M_{i,j}, M_{i+1,k})$ 사이에서 인접 의존관계를 설정할 때, $M_{i+1,k}$ 가 체언 수식어구이면 $M_{i,j}$, $M_{i+1,k}$ 사이 뿐만 아니라 $M_{i,j}$ 와 $M_{i+1,k}$ 의 후보 지배소 사이에 의존관계를 고려한다. 이 때 설정된 의존관계를 명사구 지배가능경로 의존관계라고 한다.

3 장의 예문 (1.2), (1.3)에서 두 개의 명사구가 같은 형태를 가지지만 실제 의미에 따른 수식 구조는 다르다. 이와 같이 명사구 안에서는 바로 뒤의 어절 뿐만 아니라 그 뒤의 어절도 후보 지배소로 사용될 수 있다. 모든 가능한 수식 구조를 고려해 주기 위해서는 명사구 지배가능경로 의존관계 규칙이 반드시 필요하다. 표 1은 각 구문 카테고리에 해당하는 지역 의존관계 규칙을 보여준다. 지배소로 사용된 어절이 또 다른 어절을 수식할 수 있는 어구이면, 명사구 지배가능경로 의존관계 규칙을 적용한다. 나머지 규칙은 인접 의존관계 규칙을 설명한다. 구문 카테고리에 대한 정의는 부록에 정리되어 있다. 그림 1에서는 지역 의존관계 규칙 알고리즘을 설명한다.

그림 2의 예문에서 굵은 직선은 인접 의존관계 규칙에 의해서, 곡선은 명사구 지배가능경로 의존관계 규칙에 의해서, “문제에”와 “대한”사이, “해결은”과 “어렵다” 사이의 가는 직선은 일반의존관계 규칙에 의해서 만들어 졌다. 일반 의존관계 규칙만 사용하면 “대한”, “해결은”, “어렵다”까지도 “우리나라”와 “청소년”

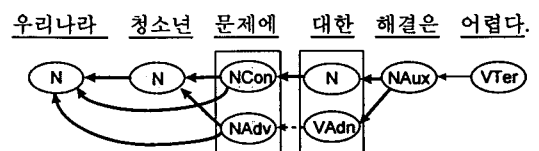


그림 2. 지역 의존관계 규칙 적용의 예

의 후보 지배소가 될 수 있지만 예문에서는 인접 의존관계 규칙과 지배가능경로 의존관계 규칙에 의해서 “문제에”까지로 범위가 제한되었다. “대한”이 명사의 절대격(N)으로 사용된 경우도 인접 의존관계 규칙에 의해서 바로 뒤 어절 “해결은”(NAux) 만을 지배소로 가진다.

5. 2 단계 한국어 의존 파서

이 절에서는 2 단계 한국어 의존관계 파서의 전체 시스템을 설명한다.

본 논문에서 제안한 파서는 그림 3에서 보는 것과 같이 2 단계로 구성되어 있다. 첫 번째 단계에서는 지역 의존관계 규칙과 일반 의존관계 규칙을 이용하여 각 어절 사이의 구문적 의존관계들을 설정하고, 각각의 경우에 문법적 기능을 부여한다. 어절 사이의 모든 가능한 구문적 의존관계가 구문 의존관계 그래프(syntactic dependency graph)에 표현된다. 초기 구문 의존관계 그래프는 두 어절 사이의 관계에 의해서만 만들어지기 때문에 많은 구조적 애매성을 가지고 있다. 두 번째 단계에서는 초기 구문 의존관계 그래프로부터 제약조건에 위반되는 의존관계들 모두 제거하여, 구문적 애매성이 해소된 구문 의존관계 트리(syntactic dependency tree)를 생성한다. 구조적 애매성을 해소하는 과정은 일치하지 않는 의존관계를 효율적으로 제거하기 위하여, 여러가지의 문장 단위의 의존관계 제약조건을 구문 의존관계 그래프에서 전파시키는 제약조건 만족문제(constraint satisfaction problem)로 볼 수 있다. 효율적인 파싱을 위하여 술어 중심의 제약 전파 알고리즘과 FC(forward checking)알고리즘 사용된다.

2 단계 한국어 의존관계 파서는 중간 결과를 하나의 구문 의존관계 그래프에 저장한 후 제약조건 전파를 통해서 모든 가능한 파서트리를 하나씩 생성한다. 구문 의존관계 그래프를 이용하면 모든 가능한 구문 의존관계를 효율적으로 하나의 합축된 그래프에 표현할

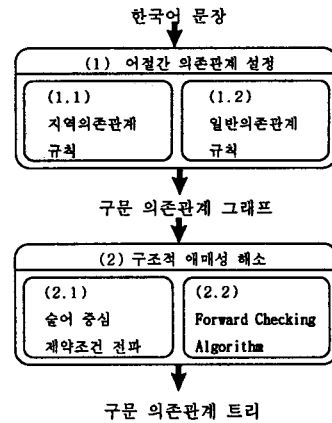


그림 3. 2 단계 한국어 의존 파서의 전체 구성

수 있고, 구문적 애매성이 있는 위치들을 쉽게 발견할 수 있어서 그 위치들을 중심으로 효율적으로 애매성을 해소해 나갈 수 있다[8]. [6]에서 제안한 “지배가능경로를 이용한 오른쪽 우선 구문 분석”은 지배가능경로를 이용하여 생성되는 의존트리의 수를 줄일 수 있고, 의존관계의 검사를 위해서 지배가능경로 상의 문장 성분만을 조사하면 되는 장점이 있지만, 모든 가능한 중간 과정의 의존트리를 동시에 저장하고 있어야 하는 단점이 있다. 특히 한국어와 같이 구문적 애매성이 많은 문장을 분석할 때 가능한 파서트리가 수천 개까지 생성될 수 있는데, 이 경우 중간 과정의 파서트리를 모두 저장하고 있는 것은 기억장치 이용면에서 매우 비효율적이다. 그러나 2 단계 의존파서는 파싱 과정에서 하나의 구문 의존관계 그래프만 유지하고 있기 때문에 기억장치 이용면에서 아주 효율적이다.

표 2. 지역 의존관계 규칙을 적용했을 때 의존관계 수의 변화

문장의 길이 (어절)	문장 수	평균 의존관계 수		(A)/(B) (%)
		지역의존관계 적용(A)	지역의존관계 적용안함(B)	
3~4	12	3.58	3.83	93.5
5~6	34	7.68	9.41	81.6
7~8	55	12.76	16.34	78.1
9~10	42	17.76	24.52	72.4
11~12	34	19.59	29.62	66.1
13~14	12	32.42	52.00	62.3
15~16	8	34.75	55.38	62.7
17 이상	3	66.00	128.67	51.3
계	200	16.41	23.78	69.0

6. 실험

본 논문에서 제안한, 지역 의존관계를 이용한 2 단계 한국어 의존파서는 Sun sparc 10 workstation 에서 C 언어로 구현되었다. STEP 2000 국어정보처리기술 개발 과제인 통합 국어정보베이스 말뭉치 중 임의로 200 문장을 선택하여 실험하였다. 실험에 사용된 문장은 3 어절에서 21 어절로 구성되어 있고 한 어절 당 형태소 분석 결과는 평균 1.11 개이다. 실험에서는 지역 의존관계 규칙을 사용하였을 때와 사용하지 않았을 때, 파싱의 첫 번째 단계에서 생기는 의존관계의 수를 비교하였다.

표 2 의 실험 결과를 보면 지역 의존관계 규칙을 사용했을 경우 사용하지 않았을 때에 비해서 평균 69% 정도로 의존관계의 수가 줄어들었다. 짧은 문장에 대해서는 의존관계의 수에서 차이가 많이 나지 않지만, 문장이 길어질수록 의존관계의 수가 급격히 줄어드는 것을 볼 수 있다. 이 때 없어진 의존관계들은 모두 원하지 않는 분석 결과를 만들어 낼 가능성이 있는, 구문적으로 바람직하지 않는 의존관계이다. 의존관계의 수가 선형적으로 증가할 때 파싱 결과의 수는 지수적으로 증가하기 때문에, 문장의 길이가 길어질수록 제안한 방법이 좋은 효율을 보인다는 것을 알 수 있다.

7. 결론

본 논문에서는 지역 의존관계 규칙을 이용한 2 단계 한국어 의존관계 파서를 제안하였다. 첫 번째 단계에서 지역 의존관계 규칙을 사용하여 불필요한 의존관계의 생성을 억제함으로써 두 번째 단계의 부담을 줄이고 전체적인 효율을 높였다. 특히 긴 문장의 파싱에서 불필요한 의존관계의 수를 많이 줄일 수 있어서 좋은 효율을 보여주었다. 그러나 제안한 시스템은 실제로 사용되기 위해서 몇 가지 해결해야 할 문제점들을 가지고 있다. 지역 의존관계를 사용하여 불필요한 의존관계를 많이 제거하였지만, 여전히 많은 구문적 애매성이 존재한다. 이 문제는 구문 단계에서 여러가지 제약조건을 가하여 어느 정도 해결할 수는 있지만, 완전히 해결하기 위해서는 어떤 형태론든지 의미 정보를 사용하여야 한다. 그러나 정확한 의미 정보를 획득하고 이용하는 과정은 아주 어려운 일이다. 앞으로 대량의 말뭉치에서 자동 추출한 의미분류체와 동사의 격정보를 사용하여 보다 정확하게 한국어를 분석할 계획이다.

참고 문헌

- [1] 강은국, "조선어 문형연구," 박이정출판사, pp.59-72, 1995.
- [2] 김창제, 정천영, 김영훈, 서영훈, "부분적인 어절 결합을 이용한 효율적인 한국어 구문 분석기," 한국정보과학회 가을 학술발표논문집, 제 22 권 2 호, pp. 597-600, 1995.
- [3] 나동열, "한국어 파싱에 대한 고찰," 정보과학회지 제 12 권 제 8 호, pp.33-46, 1994.
- [4] 류범모, 이태승, 이종혁, 이근배, "술어 중심 재의 전파를 이용한 2-단계 한국어 의존 파서," 한국정보과학회 봄 학술발표논문집, 제 23 권 1 호, pp. 923-926, 1996.
- [5] 윤덕호, 김영택, "다단계 여과 및 탐색을 이용한 의존문법에 기반을 둔 한국어 분석 알고리즘," 한국정보과학회 논문지, 제 19 권 6 호, pp. 614-623, 1992.
- [6] C.H. Kim, J.H. Kim, J.G. Seo, G.C. Kim, "A Right-to-Left Chart Parsing for Dependency Grammar using Headable Paths," in Proceedings of the 1994 International Conference on Computer Processing of Oriental Languages, pp.175-180, 1994.
- [7] Covington, M. A., "A Dependency Parser for Variable-Word-Order Languages," Research Report AI-1990-01, Artificial Intelligence Programs, Univ. of Georgia, 1990.
- [8] Jungyun Seo, Robert F. Simmons, "Syntactic Graphs: A Representation for the Union of All Ambiguous Parse Trees," Computational Linguistics, Vol 15, Num 1, pp. 19-32, 1989.
- [9] S.K. Park, C.M. Jeong, J.M. Jo, S.J. Lee, "An Effective Korean Syntactic Analyzer Using Longest Grouping," Natural Language Processing Pacific Rim Symposium '95, Vol.1, pp. 348-352, 1995.

[부록] 구문 카테고리 분류

1. N(Noun) : 명사
2. NAdn(Noun + Adnomial Particle) : 명사 + 관형격조사
3. NObj(Noun + Objective Particle) : 명사 + 목적격조사
4. NNom(Noun + Nominative Particle) : 명사 + 주격조사
5. NAdv(Noun + Adverbial Particle) : 명사 + 부사격조사
6. NCon(Noun + Conjunctive Particle) : 명사 + 접속조사
7. NAux(Noun + Auxiliary Particle) : 명사 + 보조사
8. Adn(Adnomial) : 관형사
9. Adv(Adverb) : 부사
10. VTer(Verb+Terminative Ending) : 동사 + 종결어미
11. VAdn(Verb+Abnomial Ending) : 동사 + 관형형어미
12. VCon(Verb+Conjunctive Ending) : 동사 + 연결어미
13. VAdv(Verb+Adverbial Ending) : 동사 + 부사형전성어미
14. ATer(Adjective+Terminative Ending) : 형용사 + 종결어미
15. AAdn(Adjective+Abnomial Ending) : 형용사 + 관형형전성어미
16. ACon(Adverb+Conjunctive Ending) : 형용사 + 연결어미
17. AAdv(Adverb+Adverbial Ending) : 형용사 + 부사형전성어미