Artficial Intelligence Laboratory

차트파싱 기법을 이용한 한국어 의존문법 파서 구현 방법

2018 자연언어처리 여름학교

부산대학교 전기컴퓨터공학부

권 혁 철(hckwon@pusan.ac.kr)

부산대학교 PUSAN NATIONAL UNIVERSITY

이 연구는 삼성전자 미래기술육성센터의 지원을 받아 수행된 연구임(SRFC-IT1402-03)

자연언어 처리

- 규칙에 의한 방법
  - 언어에 대한 전문적 지식이 필요(고도의 판단력)
  - 규칙화 과정에서 규칙 충돌과 규칙
- 통계적 방법
  - 학습용 말뭉치 필요
  - 자질 추출 필요
- 심층학습
  - 학습용 말뭉치 필요
  - 자질 추출 필요성 최소화

통사분석

규칙에 의한 방법

단점

- 고도의 판단력이 필요
- 규칙 간 충돌 회피가 필요
- 일관성이 필요
- 오랜 검증 시간이 필요

장점

- 말뭉치가 불필요
- 통계적, 심층학습 등 기법을 활용할 수 있음
- 기존 언어자원을 활용할 수 있음
- 언어의 변화는 아주 느리므로 한번 만들면 오래 사용 가능
- Overfitting이 없음
- Broad-coverage 접근이 가능

궁극적

Hybrid 접근 방법이 해결책?

문장 분석

왜 의존 문법인가?

- 이미 1988년부터 의존 문법을 주장함(한국어 특성을 기반으로)
- 언어는 완벽하지 않음
- 구구조문법을 이용한 방법은 영어와 같은 언어에서 출발했지만, 한계를 보임. 영어에서도 LFG, HPSG는 의존문법의 특징을 도입하고 있음
- 의미 처리를 동시에 적용할 수 있음
- 심층학습에 적합함

한계

- 문맥자유문법을 넘어서면 처리가 어려움(0.5% 미만 문서에서 보임)
  - 밥은 아주 먹기 쉽습니다.
- 불가능한 것은 아님

영상과 언어는 다르다

- 영상은 자동차가 옆에 있다고 자전거가 오토바이가 되지 않는다
  - 말하는 사람, 장소, 심지어 상식에 따라 내용이 달라진다.
  - 시간에 따라서도 달라진다.
- 영상은 생략이 없다
  - (zero) Anaphora
  - 일부 숨어 있을 수는 있어도
- 영상은 고유명사가 없다.
- 영상은 추상적 개념이 없다
  - 단, 예술 작품은 제외

이끈말이 뒤에 오게 하는 것이 도움이 되는가?

- 모형을 단순화할 수 있어서 효과적임
- 언어처리 모형이 단순할수록 문장분석 시스템을 만들기 쉬움
  - 한국어는 충분히 가능함
- SVO 언어와 다르게 SOV 언어는 조사나 어미 등 추가의 문법 요소가 있으므로 오히려 효과적임, 특히 어순이 자유로운 특징을 쉽게 해결할 수 있음
  - 영어도 HPSG는 단순화하려고 함

말뭉치, 학습자료

- 말뭉치만 있으면 학습이 가능한가?
  - 아마 장난감 수준이라면 가능할 것이다
  - 그러나 언어적 이해와 이를 이용한 모델 구성이 없다면!!

- 말뭉치는 공유할 수 있는가?
  - 정말 잘 만들어졌다면
  - 그런데 그게 쉬울까?

말뭉치, 학습자료

- 파스트리와 같은 말뭉치를 만드는 데는 고도의 언어적 지식이 필요하며, 비용도 많이 들고, 일관성을 유지하기도 쉽지 않다.
  - 문장 분석용 학습 말뭉치가 충분했다면 영어 문장분석 기술도 크게 발전했을 것이다.
- 그러나 원시말뭉치에서 비교사 학습을 하는 것도 아주 어렵다.
- 의미와 관련된 말뭉치로 가면
- 기계번역용 학습말뭉치는 예외적으로 필요에 의해 이미 많이 개발되어 있다.
- 아주 큰 지렛대를 주면 지구를 들 수 있다고 했다.
  - 말뭉치만 달라. 그러면 구현할 수 있다. 정말 그럴지도 모르지만?
  - 의미, 시간, 공간이 되면 어떻게 구축할 것인가 자체도 알 수 없다.
  - 구글도 사람과 관련된 지식은 직접 입력하고 있다.

말뭉치를 만들어 보면

- 그 과정에서 언어의 의미를 알고 새로운 방향을 알 수 있다.

- LSTM
  - Pivot 방식은 오랜 언어처리의 꿈이었다.

- 파스트리처럼 복잡한 문제는
  - 규칙에 의한 방법으로 1차 가공한 후
  - 이를 바탕으로 학습말뭉치를 만들고
  - 학습말뭉치의 일관성을 검증한 후
  - 기계학습을 이용한 시스템을 만들고
  - 규칙과 기계학습을 결합하여 새로운 시스템을 만드는 과정을 반복하여 목표를 향해 가야 함

- 그 과정에서 규칙에 의한 시스템도 고도화하지만, 기계학습 시스템의 모형도 고도화해야 함

지금으로는 거의 처리가 불가능한 문제

- 시간과 공간, 특히 문서에서 시간과 공간의 표현과 추론
- 텍스트와 상황을 이용한 문맥 분석
- Metaphor(은유)
- Speech Act
- 문서와 문서 간의 일관성
- 거짓 정보 판단
- 예측이 거의 불가능한 문맥에 따른 맞춤법 오류

차트 파싱

- Martin Kay가 개발
- 가장 초기 파싱 기법의 하나로 아주 단순함. 따라서 심층학습, 통계적 방법 등 다양한 방법과 결합이 쉬움
- Dynamic programming 기법을 활용함
- 한국어처럼 이끈말이 뒤에 오면 문장 분석이 더욱 쉬워짐
- 부산대학교는 1988년부터 한국어를 차트파싱으로 개발했음. 그러나 ???
- 2012년부터 새로 개발

현 시스템

- Broad-coverage
  - 모든 가능한 tree 생성, garden-path sentence까지 포함
  - 그 친구를 만나는 (사람, 것)
  - 밥을 먹는 (대로 토하다)
- 통사규칙과 collocation만 활용
  - 원래 목표는 1000개, collocation 10만 개였으나 ????
  - 따라서 수천, 수만 개 tree가 나올 수 있음
- 속도에 최적화되지 않음
- C++만으로 프로그램함
- Non-transparent sentence는 제외

자연언어처리를 위한 본 연구진의 연구방향

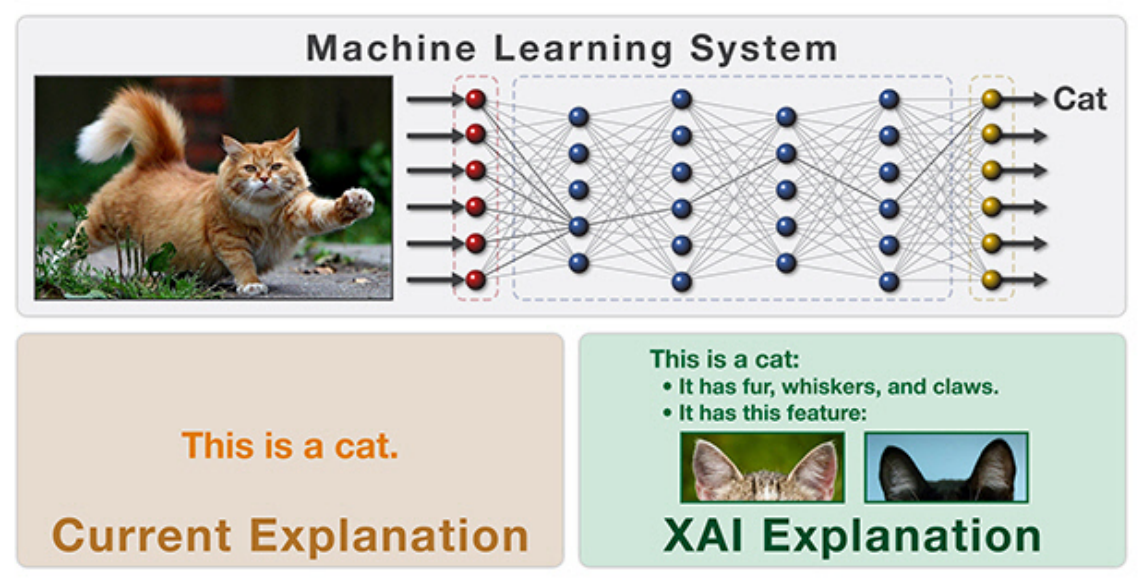
- 일상 환경(극한 언어 사용 환경)에 적용 가능한 자연언어처리 요소 기술의 개발
  - 실생활에 사용하는 문장, 인터넷 문장(댓글 등)을 처리할 수 있는 기술의 개발
  - 오탈자나 문법 오류 등을 포함한 비문의 분석도 가능한 기술의 개발
  - 최신 신문 기사, 인터넷 문장 등을 학습/분석/평가 말뭉치로 활용
- 규칙 기반 방법론과 통계 기반 방법론의 융합
  - 규칙 기반 모형과 통계 기반 모형의 융합을 통한 성능 향상
  - 언어학적 검증에 기반을 둔 규칙의 구축과 통계 기법을 통한 규칙의 일반화
- 확장성(Scalability)의 보장
  - 한국어 워드넷(Korean WordNet)을 중심으로 한 체계적인 언어자원 관리와 활용

한국어 구문분석 전략

- 생성 문법
  - 구구조문법(phrase structure grammar)
  - 의존문법(dependency grammar)
- 형태소 분석 결과의 활용
  - 품사 태깅(POS tagging)을 통해 판단한 최적의 결과만을 선택하여 활용
  - 모든 형태소분석 후보를 활용
- 구문분석 단위
  - 어절
  - 형태소
- 구문분석 방법
  - 전이 기반 파싱(Transition-Based Parsing)
  - 차트 파싱(Chart Parsing)
- 구문분석 출력 결과
  - 가장 정확하다고 판단한 1개의 구문분석 트리만 출력
  - 구문분석 트리 후보를 순위화하여 출력

특징 (1) - 규칙 기반 문장분석

- 자연스러운 지식 표현 / 처리 과정과 지식의 분리
- 빠른 처리 속도
- 학습 말뭉치가 불필요함
- 응용영역과 무관한 시스템 구현 가능
- 불완전하고 불확실한 지식(예외 현상 처리 포함)을 다룰 수 있음
- 설명 가능 인공지능(Explainable Artificial Intelligence; XAI)



(출처:DARPA)

특징 (2) – 형태소 단위의 문장분석

- 어절 단위와 비교하여 더 적은 데이터로 효과적인 모형의 개발이 가능
  - 먹다 VS 먹고/먹어서/먹으니/먹어서부터 …

- 모든 형태소 분석 결과를 활용하여 오류 전파 문제에 강건함
  - 나의 어머니와 친구이다.
    - 와/접속조사 → They are my mother and friends.
    - 와/부사격조사 → They are friends with my mother.
  - 본래
    - 본래/명사 → 본래 모습은 아무도 모른다.
    - 본래/부사 → 본래 이곳은 아무도 살지 않는다.

특징 (3) – 차트 파싱(1/2)

- 1-way 알고리즘
  - 부분 결과를 저장하기 때문에 backtracking이 없음
  - 전이 기반 파싱에서도 이를 해결하기 위한 다양한 시도가 있으나 완전하지 않음

Input Buffer $_0$어머니 $_1$가 $_2$나오다 $_3$아 $_4$있다 $_5$

Inactive Edge Pool(chart)

1	$_0$어머니$_1$
2	$_1$가$_2$ $_0$(어머니)가$_2$
3	$_2$나오다$_3$ $_0$((어머니)가)나오다$_3$
4	$_3$아$_4$ $_0$(((어머니)가)나오다)아$_4$ $_2$(나오다)아$_4$

Active Edge Pool (Agenda)

$_4$있다 $_5$

선택 전략
= 규칙, 통계, 기계학습, 심층학습 등

특징 (3) – 차트 파싱(2/2)

- 모든 형태소 분석 결과를 이용하는 문장분석에 적합함

Input Buffer $_0$어머니$_1$가$_2$나오다$_3$아$_4$있다$_5$

$_0$어머니$_1$가$_2$나$_3$와$_4$있다$_5$

Inactive Edge Pool(chart)

1	$_0$어머니$_1$
2	$_1$가$_2$ $_0$(어머니)가$_2$
3	$_2$나오다$_3$ $_2$나$_3$ $_0$((어머니)가)나오다$_3$
4	$_3$아$_4$ $_3$와$_4$ $_0$(((어머니)가)나오다)아$_4$ $_2$나와$_4$

Active Edge Pool (Agenda)

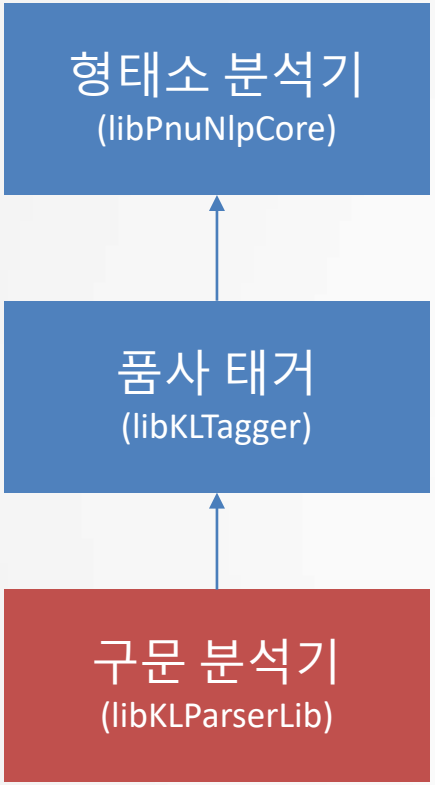
$_4$있다$_5$

모든 형태소 분석 결과를
차트에 기록하면서 분석을 수행함

형태소 분석 결과에 따라
인덱스가 달라지는 경우에도
간단한 인덱스 계산으로 가능

예) 전문의 → 전문의/명사
전문/명사 + 의/조사

부산대학교 문장 분석 시스템의 구성



- 형태소 분석 사전에 기반을 둔 형태소 분석기
- 오류 교정 기능 포함
- **제공 형태: 동적 라이브러리(Win32, Linux)**

- 형태소 분석 결과에 기반을 둔 품사 태거
- 규칙 모형과 통계 모형에 기반을 둔 알고리즘 사용
- **제공 형태: 동적 라이브러리(Win32, Linux)**

- 품사 태깅 결과에 기반을 둔 구문 분석기
- 의존문법과 지배소-의존소 관계 제약 규칙에 기반을 둔 차트 파싱 알고리즘 사용
- **제공 형태: 소스코드**

품사 태깅

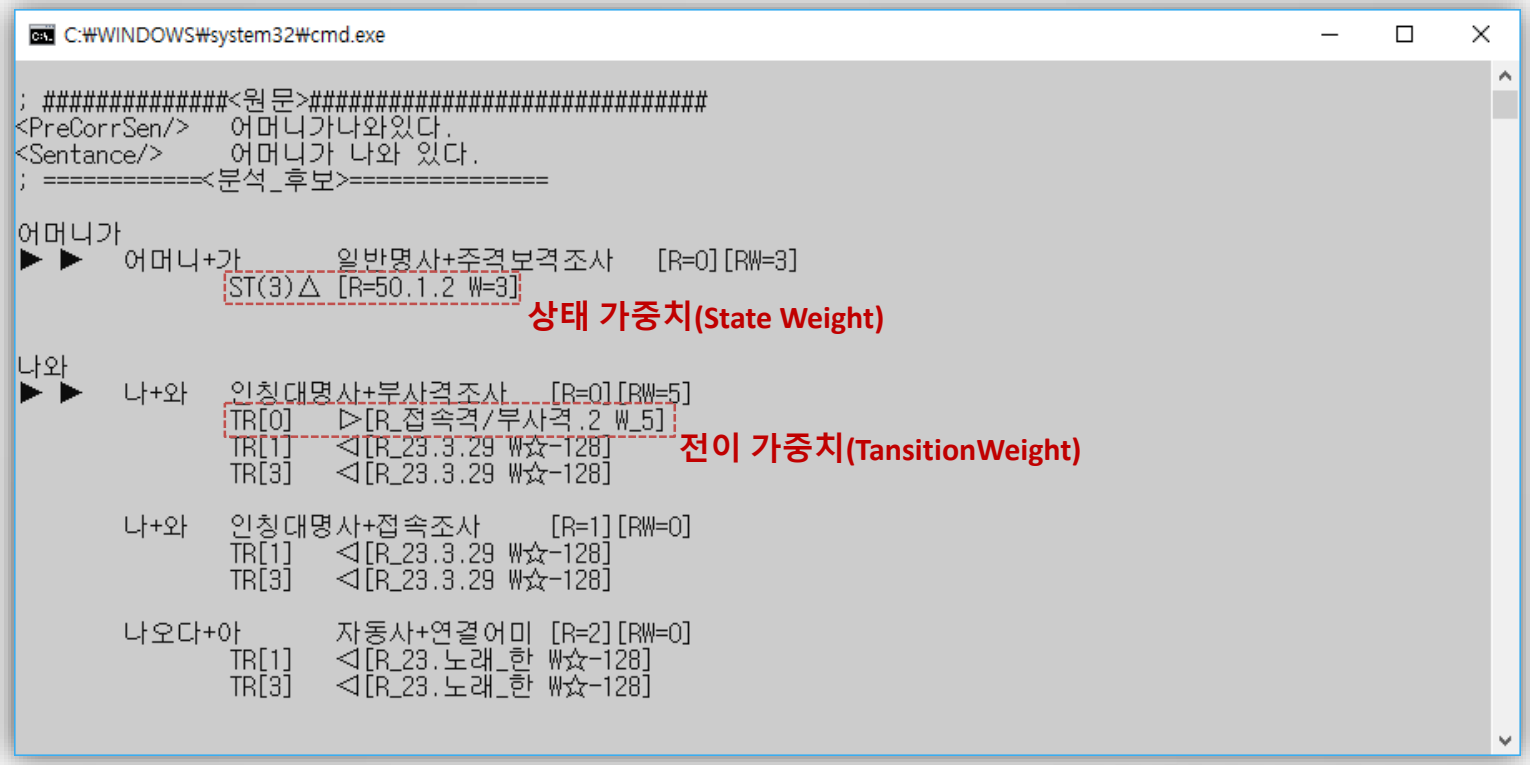
- **형태소 분석 과정에서 발생하는 품사 중의성과 어휘 중의성을 해소하는 과정**
  - **품사 중의성: 하나의 형태소가 여러 가지 품사로 분석되는 경우**
    - 예) 가는 → 가/명사 + 는/조사, 가(다)/동사 + 는/조사
  - **어휘 중의성: 하나의 단어가 다른 형태소들의 결합으로 분석되는 경우**
    - 예) 감기는 → 감기/명사 + 는/조사, 감다/동사 + 기/어미 + 는/조사

- **품사 태깅을 위한 일반적인 방법**
  - **규칙 기반 접근법**
    - 언어 정보를 생성 규칙(production rule)의 형태로 표현하고 이를 적용하여 태깅을 수행함
    - 예) [A or B] → A [C or D] : A와 B 사이에 중의성이 있을 때 C나 D가 오면 A를 선택함
  - **통계 기반 접근법**
    - 말뭉치에서 추출한 통계 정보를 이용하여 태깅 모형을 학습하여 태깅을 수행함
    - 예) HMM(Hidden Markov Model), CRF(Conditional Random Field) 등
  - **복합적 접근법**
    - 규칙 기반 접근법과 통계 기반 접근법을 결합하여 태깅을 수행함
    - 통계적 접근법을 먼저 적용한 다음, 통계적 신뢰도가 낮은 결과에 대해서만 태깅 규칙을 사용하여 규칙 기반 접근법으로 다시 태깅하는 방법 등이 있음

부산대학교 품사 태거(POS Tagger)

- 수작업으로 구축한 규칙에 기반을 둔 모형과 대규모 말뭉치에 추출한 통계 정보에 기반을 둔 모형, 그리고 이들을 결합한 모형으로 구성되어 있음

- 규칙과 통계에 의한 중의성 해소
  - 수작업으로 구축한 규칙 약 1,500개와 대규모 말뭉치에서 추출한 통계 정보를 바탕으로 제어



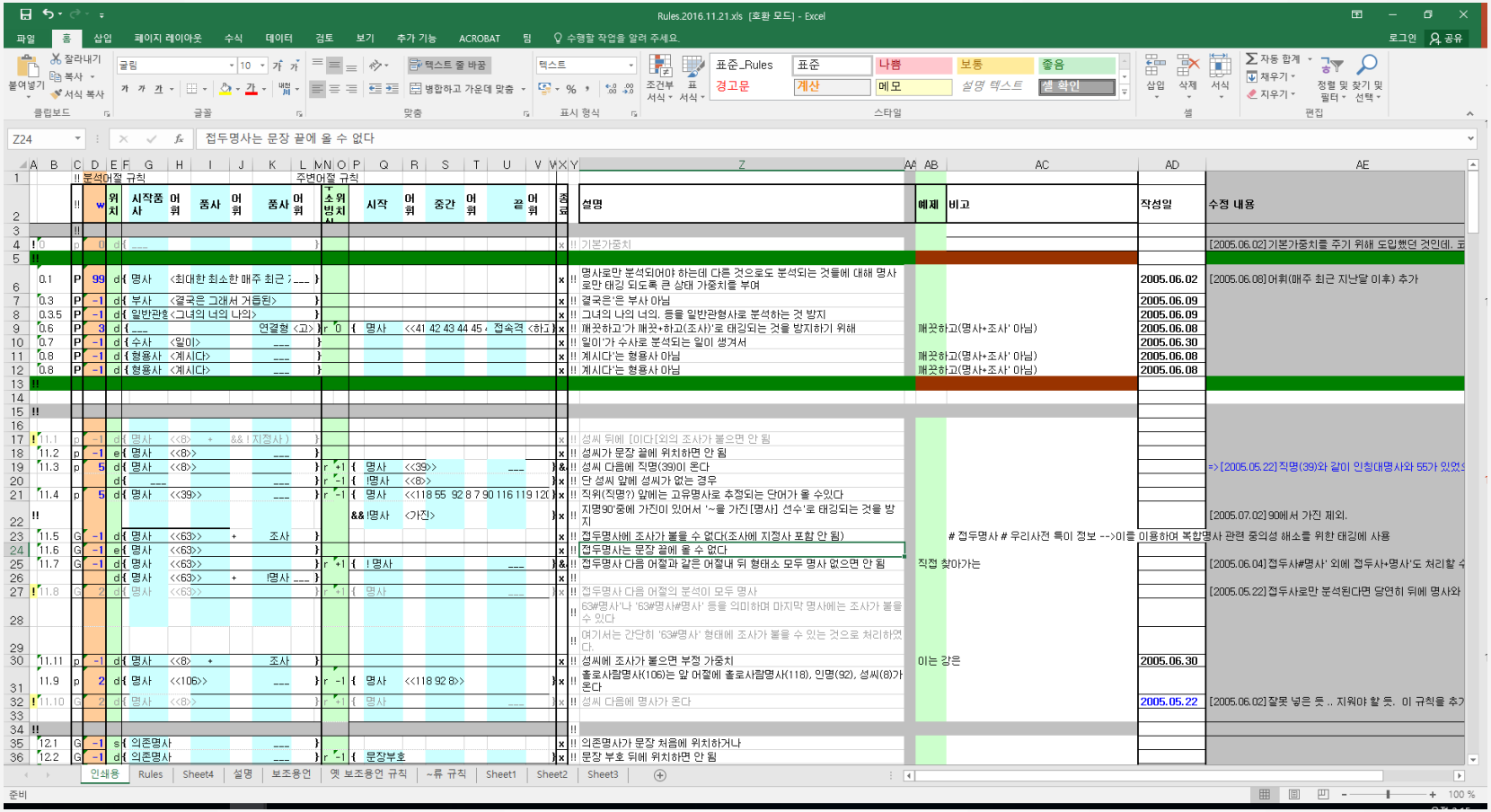
품사 태그 집합(POS Tag Set)

			부산대	세종
체언	명사			o
		일반명사	o	
		동작성명사	o	
		상태성명사	o	
	고유명사		o	o
	의존명사			o
		일반	o	
		단위성	o	
	대명사		o	o
	수사		o	o
용언	동사			o
		자동사	o	
		타동사	o	
		자타동사	o	
	형용사		o	o
	보조용언		o	o
	지정자		o	
		긍정 지정사		o
		부정 지정사		o
수식언	관형사			o
		일반관형사	o	
		수관형사	o	
	부사			
		일반부사	o	o
		접속부사	o	o
독립언	감탄사		o	o

			부산대	세종
관계언	격조사	주격	o	o
		보격	o	o
		목적격	o	o
		관형격	o	o
		부사격	o	o
		호격	o	o
		인용격	o	o
	접속조사		o	o
	보조사		o	o
의존형태	어미			
		선어말어미	o	o
		종결어미	o	o
		연결어미	o	o
	전성어미			
		명사형	o	o
		관형사형	o	o
	접두사			o
		일반접두사	o	
		수접두사	o	
	접미사			o
		일반접미사	o	
		수접미사	o	
		복수접미사	o	
	파생접미사	관형사화	o	
		동사화	o	o
		형용사화	o	o
계			41	31

규칙 기반 품사 태깅 방법(1/2)

- 문맥 정보를 활용한 품사 중의성 해소 규칙



- 최소한의 문형 정보만 활용하여 빠른 시간에 중의성을 해소하는 것이 가능함

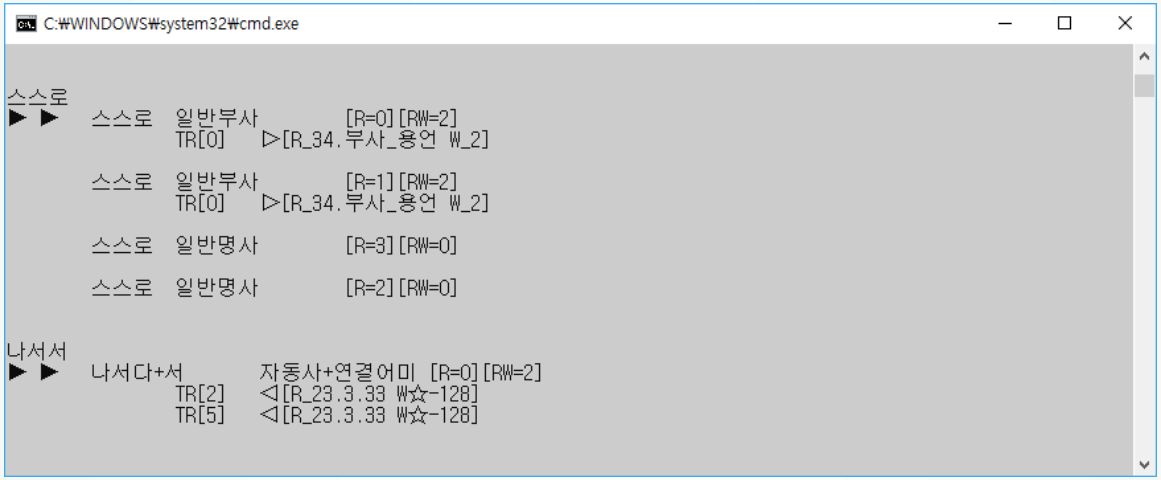
규칙 기반 품사 태깅 방법(2/2)

품사 중의성 해소 규칙

(규칙) 어절 전체가 부사나 명사로 분석될 때, 조사 결합이 없으면 부사의 가중치를 높임

→ (예문) 전 본래 돌을 좋아하여 여러 곳을 다니며 돌을 주워 모읍니다.

→ (예문) 그는 남이 싫어하는 일을 스스로 나서서 했다.



인명, 지명과 같은 미등록어의 추정

미등록어가 포함된 어절을 분석하는 기능 추가

- (예문) 그는 사람을 부춘산에게 보내, 냇가에 낚시질하는 엄자릉을 데려 오라 했다.

→ 앞에서부터 한 음절씩 사전에 등재된 고유명사로 대치하면서 분석이 되는지를 확인함

→ '홍길동+에게', '홍길동+을'로 분석이되므로 '부춘산'과 '엄자릉'을 고유명사로 판단함

통계 기반 품사 태깅 방법

- 기본 모델: 범주 패턴 기반 Hidden Makov Model(HMM)
  - Sequential labeling problem의 대표적인 해결책 중 하나인 HMM에 기반을 두고 통계적으로 모형화함
  - 어절 내 제약 조건이나 어절 간 제약조건과 같은 형태론적/통사론적 제약 조건을 반영할 수 있는 범주 패턴을 설정함

$$T(w_{1,n}) = \arg\max_{t_{1,n}} P(t_{1,n} \mid w_{1,n}) \approx \arg\max_{t_{1,n}} P(cp_{1,n} \mid ms_{1,n})$$

$$= \arg\max_{t_{1,n}} \prod_{i=1}^{n} P(cp_i, ms_i)$$

$m_{k;i}$	어절 w_k의 후보형태소열 $m_{k;*}$ 내의 i번째 형태소
$m_{k;\Omega}$	어절 w_k의 후보형태소열 $m_{k;*}$ 내의 어절핵
$m_{k;*}$	어절 w_k 내의 후보 형태소열
$t_{k;i}$	어절 w_k의 후보형태소열 $m_{k;*}$ 내의 i번째 형태소 범주
$t_{k;*}$	어절 w_k를 구성하는 후보 범주열
cp_j	범주 패턴 집합 내의 특정 범주 패턴
$cp_{j;i}$	특정 범주 패턴 cp_j 내의 i번째 범주
$CP = \{cp_1, \Lambda, cp_\beta\}$	한국어 어절 내와 어절 간에 발견되는 범주 패턴 집합
ms_i	한국어 어절 내와 어절 간에 발견되는 범주 패턴을 이루는 특정 형태소열

$$\prod_{k=1}^{n} P(cp_k, ms_k) \approx \prod_{k=1}^{n} \left\{ \prod_{j=1}^{2} P(cp_{t;j}, m_{k-2+j;\Omega}) \cdot \prod_{i=1}^{\mu} P(cp_{w;i}, m_{k,i}) \right\}$$

전처리 과정

형태소 후보 제거 및 형태소 리스트 생성

예) "이 사회의 전반적인 문제"

[이]	[사회의]	[전반적인]	[문제]
이 : 지시관형사 이 : 수관형사 이 : 명사 이 : 대명사	사회+의 : 명사+관형격조사 사+회+의 : 수관형사+단위의존명사+관형격조사 사+회의 : 수관형사+단위의존명사	전반적+이다+ㄴ : 명사+지정사+관형형전성어미	문제 : 명사

↓

[이]	[사회의]	[전반적인]	[문제]
이 : 지시관형사	사회+의 : 명사+관형격조사 사+회+의 : 수관형사+단위의존명사+관형격조사	전반적+이다+ㄴ : 명사+지정사+관형형전성어미	문제 : 명사

↓

형태소 리스트 (1*2*1*1)

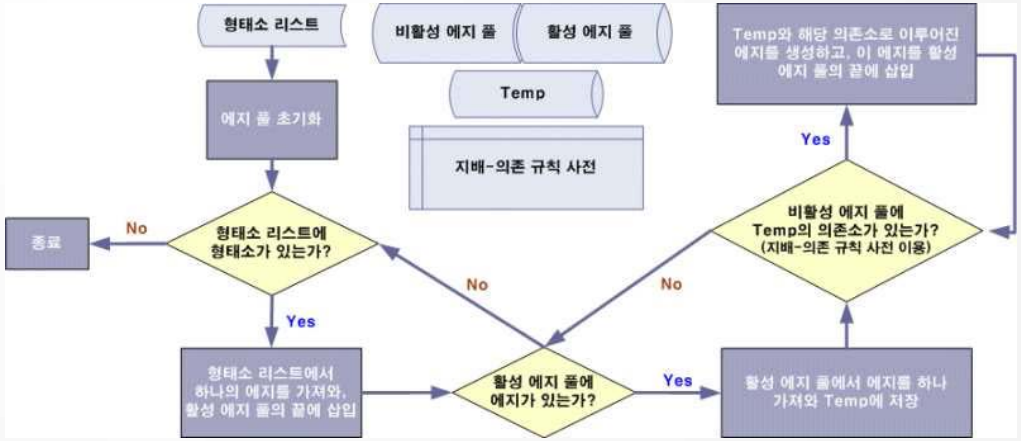
[1]	이 : 지시관형사	사회+의 : 명사+관형격조사	전반적+이다+ㄴ : 명사+지정사+관형형전성어미	문제 : 명사
[2]	이 : 지시관형사	사+회+의 : 수관형사+단위의존명사+관형격조사	전반적+이다+ㄴ : 명사+지정사+관형형전성어미	문제 : 명사

지배-의존 에지 생성

- 실제 문장 분석이 실행되는 단계
- 의존문법에 기반을 둔 규칙을 바탕으로 지배-의존 에지를 생성
  - 약 500개 가량의 수작업으로 구축된 의존 문법 규칙 존재
    - 지배-의존 연결 시 추가적으로 의존관계 패턴 사전 / 보조 용언 규칙 사전 등 사용
    - 사전에 넣을 수 없는 세부적인 규칙은 코드 상 구현
- 차트(Chart)를 자료구조로 사용하는 차트 파싱
  - Back-tracking 하지 않는 1-way 알고리즘

<표2. 의존 문법 규칙 사전의 예>

지배소	의존소	결과	인접	건너 띄기
명사	관형사	명사구	FALSE	FALSE
명사	관형사구	명사구	FALSE	FALSE
주격보격조사	명사	격조사구	TRUE	FALSE
주격보격조사	명사구	격조사구	TRUE	FALSE
목적격조사	명사	격조사구	TRUE	FALSE
목적격조사	명사구	격조사구	TRUE	FALSE
동사	명사	동사구	FALSE	FALSE
동사	격조사구	동사구	FALSE	FALSE
동사구	명사	동사구	FALSE	TRUE
동사구	명사구	동사구	FALSE	TRUE
동사구	격조사구	동사구	FALSE	TRUE
종결어미	동사	종결구	TRUE	FALSE
종결어미	동사구	종결구	TRUE	FALSE
온점	종결어미	문장	TRUE	FALSE
온점	종결구	문장	TRUE	FALSE



<그림 2. 지배-의존 에지 생성 알고리즘>

문장 분석 예

sentence 예쁜 친구의 동생과 놀며 즐겼다

Input Buffer 예쁜 친구의 동생과 놀며 즐겼다.

① ② ③ ④ ⑤

Edge Pool(Queue Chart)

Inactive Edge	Active Edge
X	①
①	X

① 예쁜 ② 친구의 ③ 동생과 ④ 놀며 ⑤ 즐겼다

sentence 예쁜 친구의 동생과 놀며 즐겼다

Input Buffer 예쁜 친구의 동생과 놀며 즐겼다.

① ② ③ ④ ⑤

Edge Pool(Queue Chart)

Inactive Edge	Active Edge
①	② | (①, ②)
① | ②	(①, ②)
① | ② | (①, ②)	X

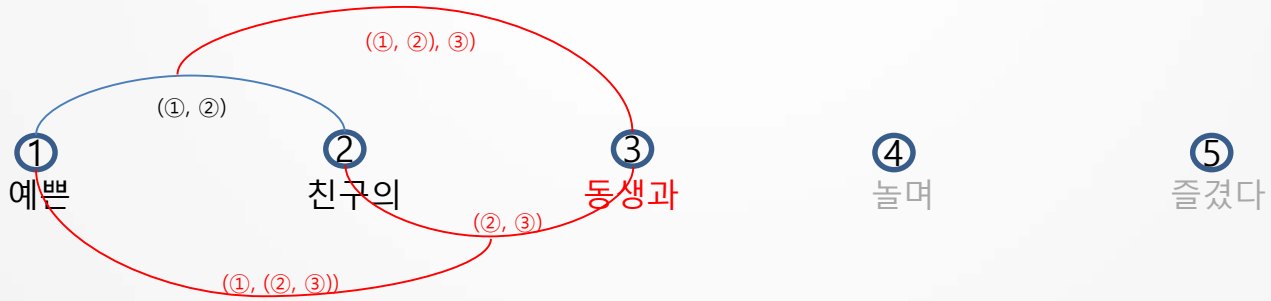
sentence 예쁜 친구의 동생과 놀며 즐겼다

Input Buffer 예쁜 친구의 동생과 놀며 즐겼다.

① ② ③ ④ ⑤

Edge Pool(Queue Chart)

Inactive Edge	Active Edge
① | ② | (①, ②)	③ | (②, ③) | ((①, ②), ③)
① | ② | (①, ②) | ③	(②, ③) | ((①, ②), ③) | (①, (②, ③))
① | ② | (①, ②) | ③ | (②, ③)	((①, ②), ③) | (①, (②, ③))
① | ② | (①, ②) | ③ | (②, ③) | ((①, ②), ③)	(①, (②, ③))
① | ② | (①, ②) | ③ | (②, ③) | ((①, ②), ③) | (①, (②, ③))	X



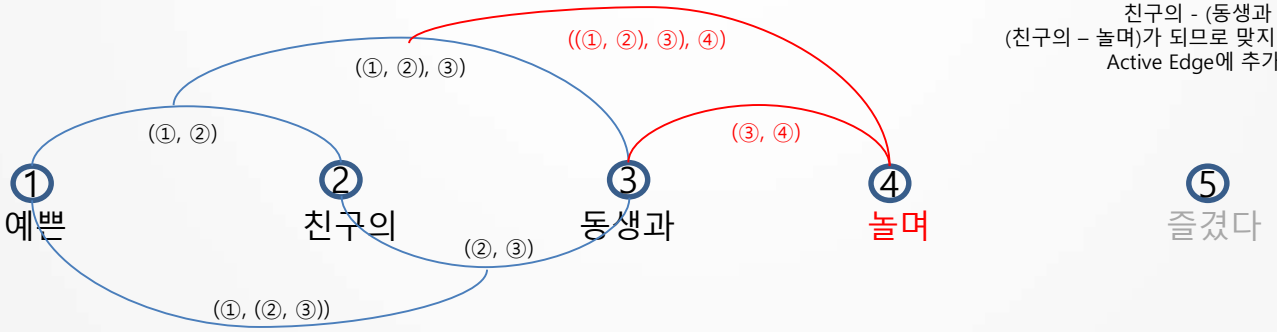
sentence 예쁜 친구의 동생과 놀며 즐겼다

Input Buffer 예쁜 친구의 동생과 놀며 즐겼다.

① ② ③ ④ ⑤

Inactive Edge	Active Edge
① | ② | (①, ②) | **③ | ((①, ②), ③) | (②, ③) | (①, (②, ③))**	④ | (③, ④) | (((①, ②), ③), ④) | ((②, ③), ④) | ((①, (②, ③)), ④) |
① | **② | (①, ②)** | ③ | ((①, ②), ③) | (②, ③) | (①, (②, ③))| ④	**(③, ④)** | (((①, ②), ③), ④) | ((②, ③), ④) | ((①, (②, ③)), ④) |
① | ② | (①, ②) | ③ | ((①, ②), ③) | (②, ③) | (①, (②, ③)) | ④ | **(③, ④)**	**(((①, ②), ③), ④)** | ((②, ③), ④) | ((①, (②, ③)), ④)

친구의 - (동생과 놀며) 는
(친구의 – 놀며)가 되므로 맞지 않은 구문이기 때문에
Active Edge에 추가하지 않음

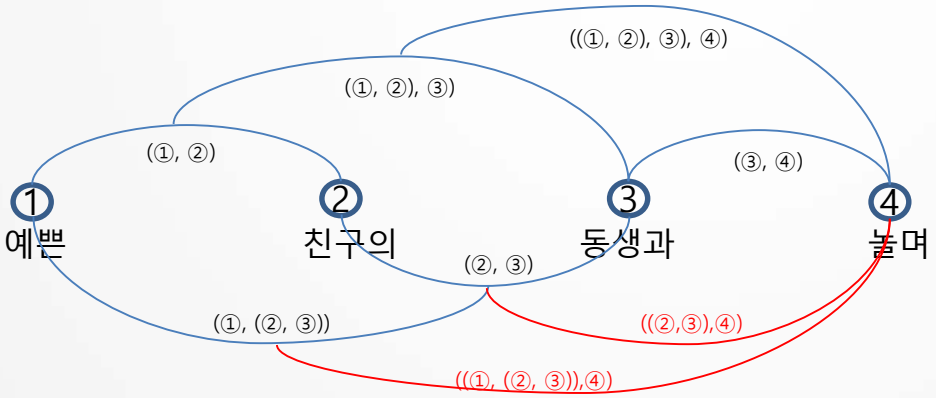


sentence 예쁜 친구의 동생과 놀며 즐겼다

Input Buffer 예쁜 친구의 동생과 놀며 즐겼다.

① ② ③ ④ ⑤

Inactive Edge	Active Edge
① | ② | (①, ②) | ③ | ((①, ②), ③) | (②, ③) | (①, (②, ③)) | ④ | (③, ④) | (((①, ②), ③), ④)	((②, ③), ④) | ((①, (②, ③)), ④)
① | ② | (①, ②) | ③ | ((①, ②), ③) | (②, ③) | (①, (②, ③)) | ④ | (③, ④) | (((①, ②), ③), ④) | ((②, ③), ④)	((①, (②, ③)), ④)
① | ② | (①, ②) | ③ | ((①, ②), ③) | (②, ③) | (①, (②, ③)) | ④ | (③, ④) | (((①, ②), ③), ④) | ((②, ③), ④) | ((①, (②, ③)), ④)	X



예쁜 – ((친구의 동생과) 놀며)) 는
((예쁜 – 놀며)가 되므로 맞지 않은 구문이기 때문
에
Active Edge에 추가하지 않음

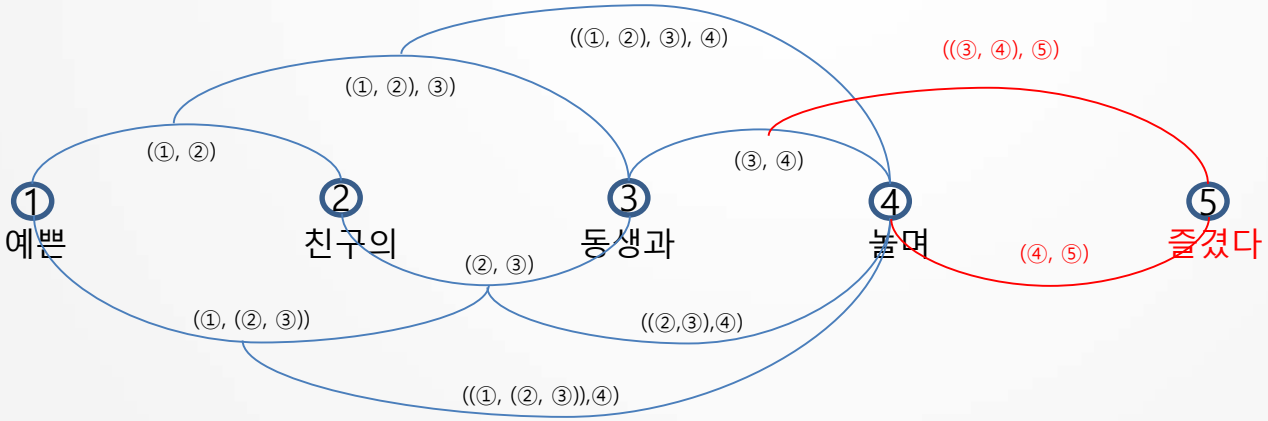
⑤
즐겼다

sentence 예쁜 친구의 동생과 놀며 즐겼다

Input Buffer 예쁜 친구의 동생과 놀며 즐겼다

① ② ③ ④ ⑤

Inactive Edge	Active Edge
① | ② | (①, ②) | ③ | ((①, ②), ③) | (②, ③) | (①, (②, ③)) | **④ | (③, ④) | (((①, ②), ③), ④) | ((①, (②, ③)), ④)**	***⑤*** | (④, ⑤) | ((③, ④), ⑤) | ((((①, ②), ③), ④), ⑤) | (((①, (②, ③)), ④), ⑤)
① | ② | (①, ②) | **③ | ((①, ②), ③) | (②, ③) | (①, (②, ③))** | ④ | (③, ④) | (((①, ②), ③), ④) | ((①, (②, ③)), ④) | ***⑤***	***(④, ⑤)*** | ((③, ④), ⑤) | ((((①, ②), ③), ④), ⑤) | (((①, (②, ③)), ④), ⑤) | (③, (④, ⑤)) | (((①, ②), ③), (④, ⑤)) | ((②, ③), (④, ⑤)) | ((①, (②, ③)), (④, ⑤))
① | **② | (①, ②)** | ③ | ((①, ②), ③) | (②, ③) | (①, (②, ③)) | ④ | (③, ④) | (((①, ②), ③), ④) | ((①, (②, ③)), ④) | ⑤ | ***(④, ⑤)***	***((③, ④), ⑤)*** | ((((①, ②), ③), ④), ⑤) | (((①, (②, ③)), ④), ⑤) | | (③, (④, ⑤)) | (((①, ②), ③), (④, ⑤)) | ((②, ③), (④, ⑤)) | ((①, (②, ③)), (④, ⑤))



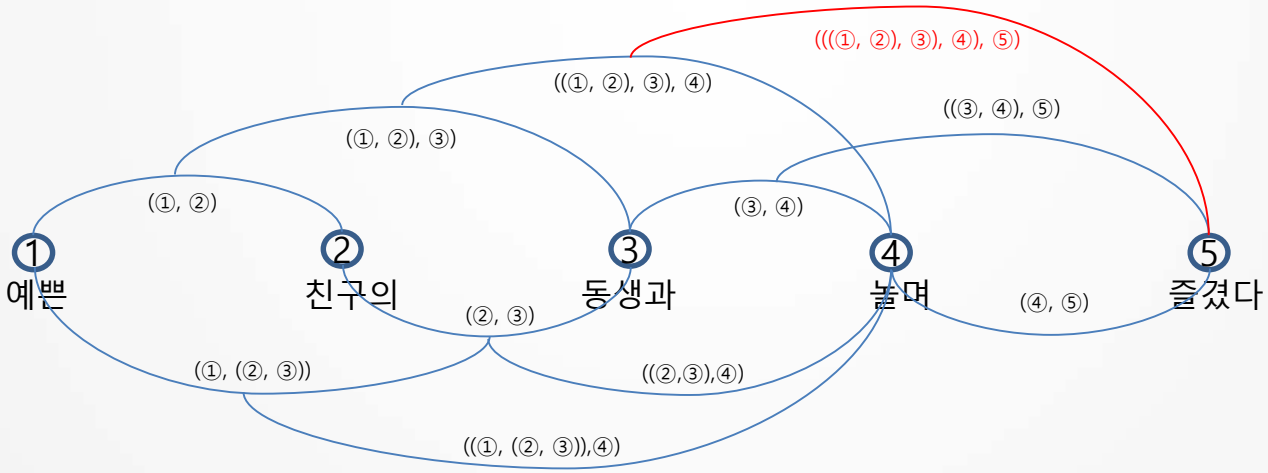
(예쁜 친구의) - (동생과 놀며) 즐겼다)) 는
(친구의 – 즐겼다) 가 되므로 맞지 않은
구문이기 때문에 Active Edge에 추가하지 않음

sentence 예쁜 친구의 동생과 놀며 즐겼다

Input Buffer 예쁜 친구의 동생과 놀며 즐겼다

① ② ③ ④ ⑤

Inactive Edge	Active Edge
① | ② | (①, ②) | ③ | ((①, ②), ③) | (②, ③) | (①, (②, ③)) | ④ | (③, ④) | | (((①, ②), ③), ④) | ((①, (②, ③)), ④) | ⑤ | (④, ⑤) | ***((③, ④), ⑤)***	***((((①, ②), ③), ④), ⑤)*** | (((①, (②, ③)), ④), ⑤) | (③, (④, ⑤)) | (((①, ②), ③), (④, ⑤)) | ((②, ③), (④, ⑤)) | ((①, (②, ③)), (④, ⑤))

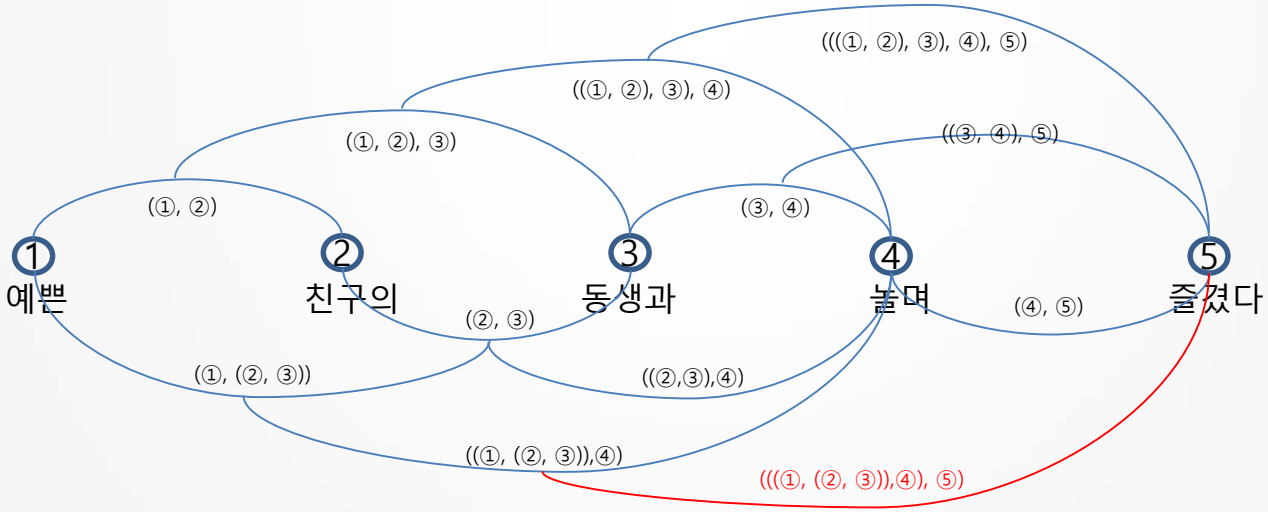


sentence 예쁜 친구의 동생과 놀며 즐겼다

Input Buffer 예쁜 친구의 동생과 놀며 즐겼다

① ② ③ ④ ⑤

Inactive Edge	Active Edge
① | ② | (①, ②) | ③ | ((①, ②), ③) | (②, ③) | (①, (②, ③)) | ④ | (③, ④) | (((①, ②), ③), ④) | ((①, (②, ③)), ④) | ⑤ | (④, ⑤) | ((③, ④), ⑤) | ***((((①, ②), ③), ④), ⑤)***	***(((①, (②, ③)), ④), ⑤)*** | (③, (④, ⑤)) | (((①, ②), ③), (④, ⑤)) | ((②, ③), (④, ⑤)) | ((①, (②, ③)), (④, ⑤))



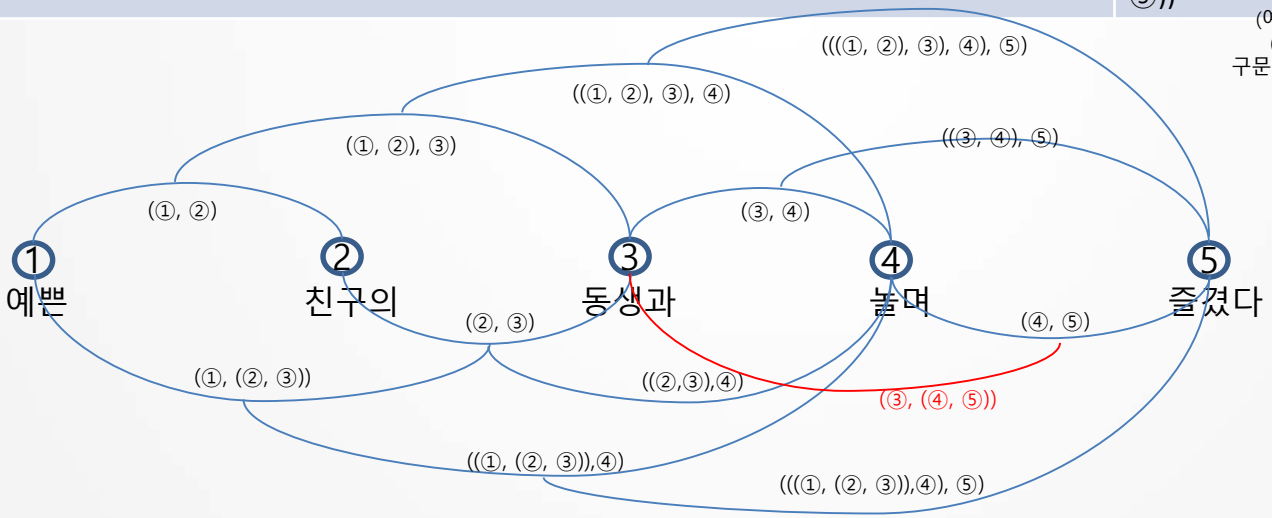
sentence 예쁜 친구의 동생과 놀며 즐겼다

Input Buffer 예쁜 친구의 동생과 놀며 즐겼다

① ② ③ ④ ⑤

Inactive Edge	Active Edge
① | **② | (①, ②)** | ③ | (((①, ②), ③) | (②, ③) | (①, (②, ③)) | ④ | (③, ④) | (((①, ②), ③), ④) | ((①, (②, ③)), ④) | ⑤ | (④, ⑤) | (((③, ④), ⑤) | ((((①, ②), ③), ④), ⑤) | ***(((①, (②, ③))), ④), ⑤)***	***(③, (④, ⑤))*** | (((①, ②), ③), (④, ⑤)) | ((②, ③), (④, ⑤)) | ((①, (②, ③)), (④, ⑤))

(예쁜 친구의) - (동생과 (놀며 즐겼다)) 는 (친구의 – 즐겼다) 가 되므로 맞지 않은 구문이기 때문에 Active Edge에 추가하지 않음

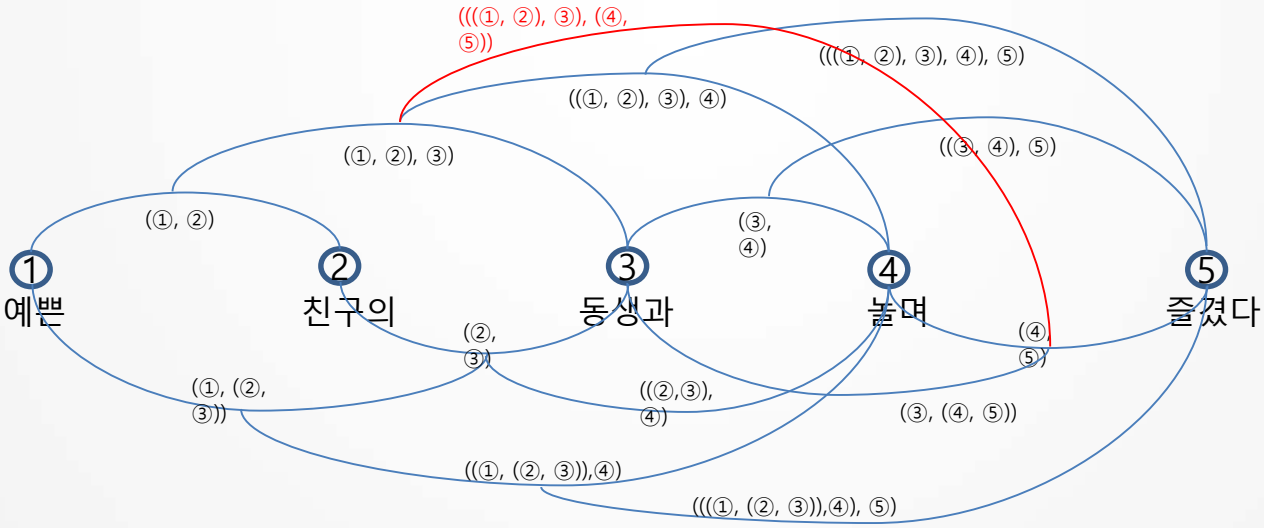


sentence 예쁜 친구의 동생과 놀며 즐겼다

Input Buffer 예쁜 친구의 동생과 놀며 즐겼다

① ② ③ ④ ⑤

Inactive Edge	Active Edge
① | ② | (①, ②) | ③ | ((①, ②), ③ | (생략. . .) | ((((①, ②), ③), ④), ⑤) | (((①, (②, ③)), ④), ⑤) | ***(③, (④, ⑤))***	***(((①, ②), ③), (④, ⑤))*** | ((②, ③), (④, ⑤)) | ((①, (②, ③)), (④, ⑤))

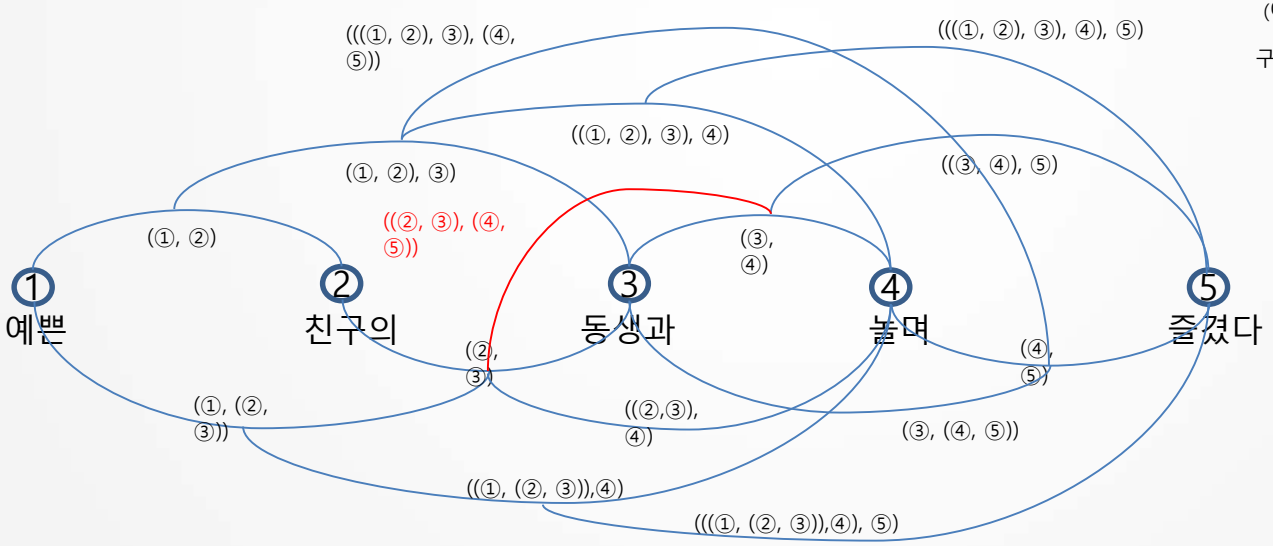


sentence 예쁜 친구의 동생과 놀며 즐겼다

Input Buffer 예쁜 친구의 동생과 놀며 즐겼다

① ② ③ ④ ⑤

Inactive Edge	Active Edge
① | ② | (①, ②) | ③ | ((①, ②), ③ | (생략. . .) | | (((①, (②, ③)), ④), ⑤) | (③, (④, ⑤)) | ***(((①, ②), ③), (④, ⑤))***	***((②, ③), (④, ⑤))*** | ((①, (②, ③)), (④, ⑤))



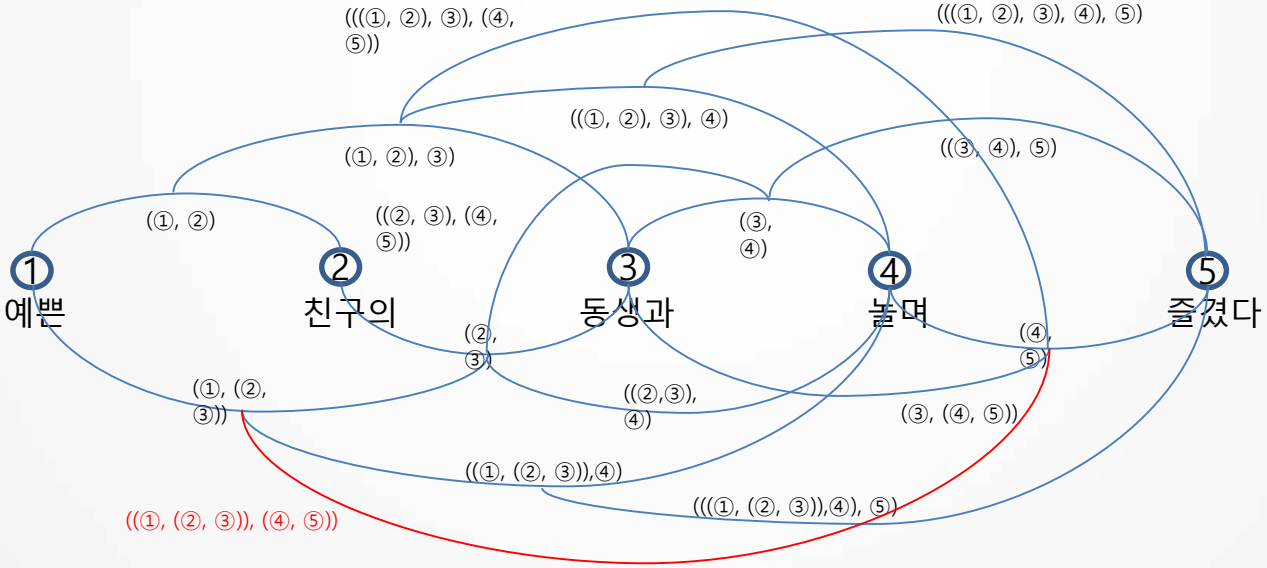
(예쁜) - (친구의 (동생과 (놀며 즐겼다))) 는 (예쁜 – 즐겼다) 가 되므로 맞지 않은 구문이기 때문에 Active Edge에 추가하지 않음

sentence 예쁜 친구의 동생과 놀며 즐겼다

Input Buffer 예쁜 친구의 동생과 놀며 즐겼다

① ② ③ ④ ⑤

Inactive Edge	Active Edge
① | ② | (①, ②) | ③ | ((①, ②), ③ | (생략. . .) | (③, (④, ⑤)) | (((①, ②), ③), (④, ⑤)) | ***((②, ③), (④, ⑤))***	***((①, (②, ③)), (④, ⑤))***

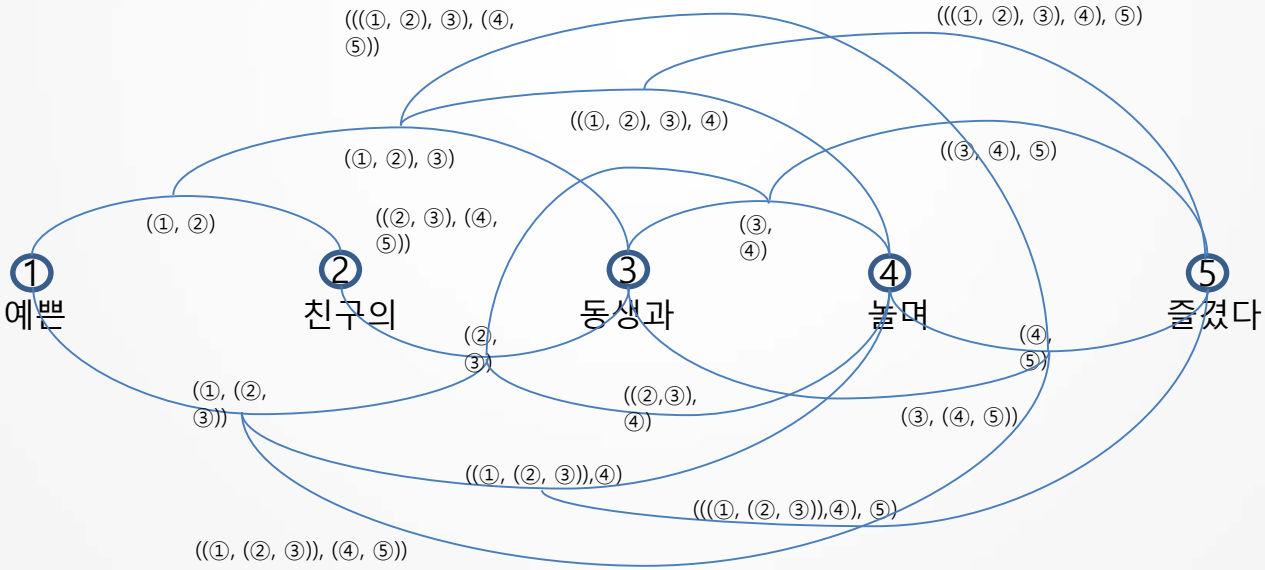


sentence 예쁜 친구의 동생과 놀며 즐겼다

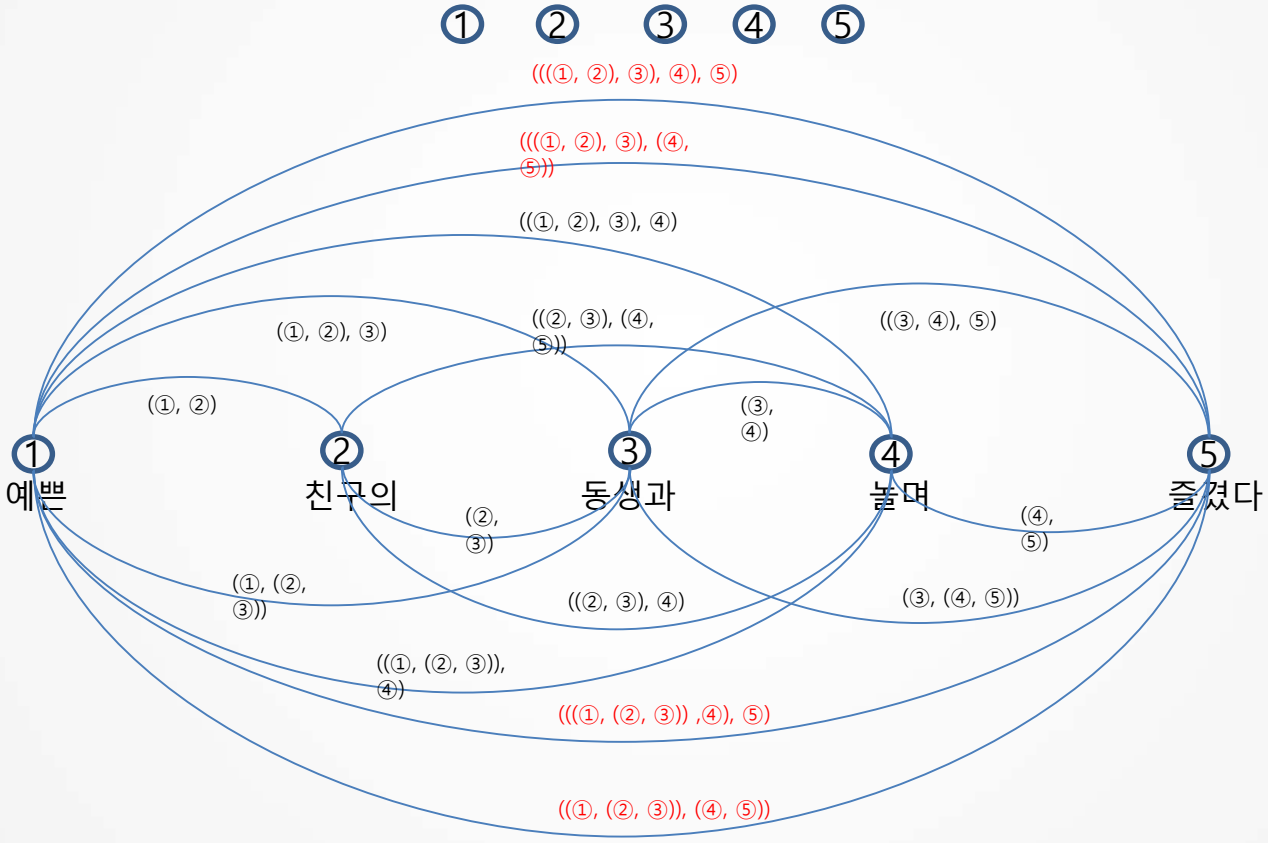
Input Buffer 예쁜 친구의 동생과 놀며 즐겼다

① ② ③ ④ ⑤

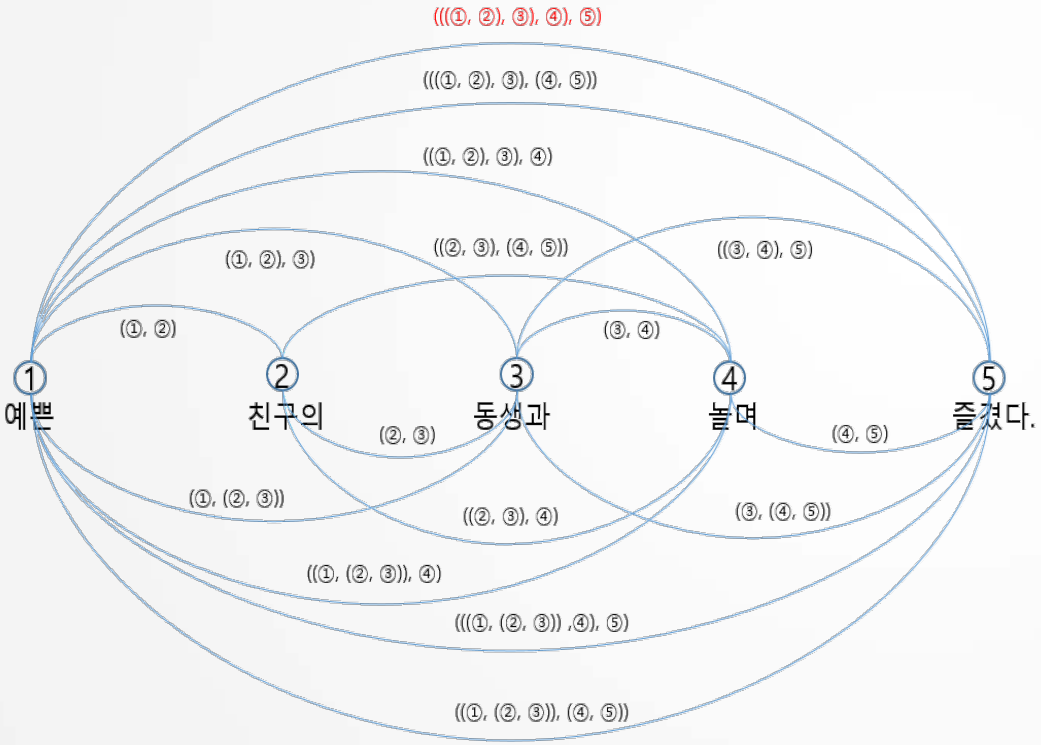
Inactive Edge	Active Edge
① | ② | (①, ②) | ③ | ((①, ②), ③ | (생략. . .) | (③, (④, ⑤)) | (((①, ②), ③), (④, ⑤)) | ((②, ③), (④, ⑤)) | ((①, (②, ③)), (④, ⑤))	X

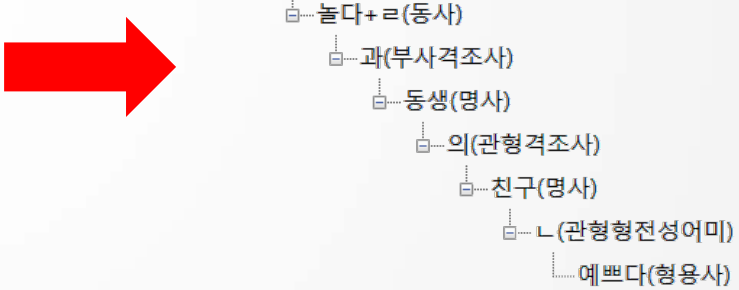


Input Buffer 예쁜 친구의 동생과 놀며 즐겼다

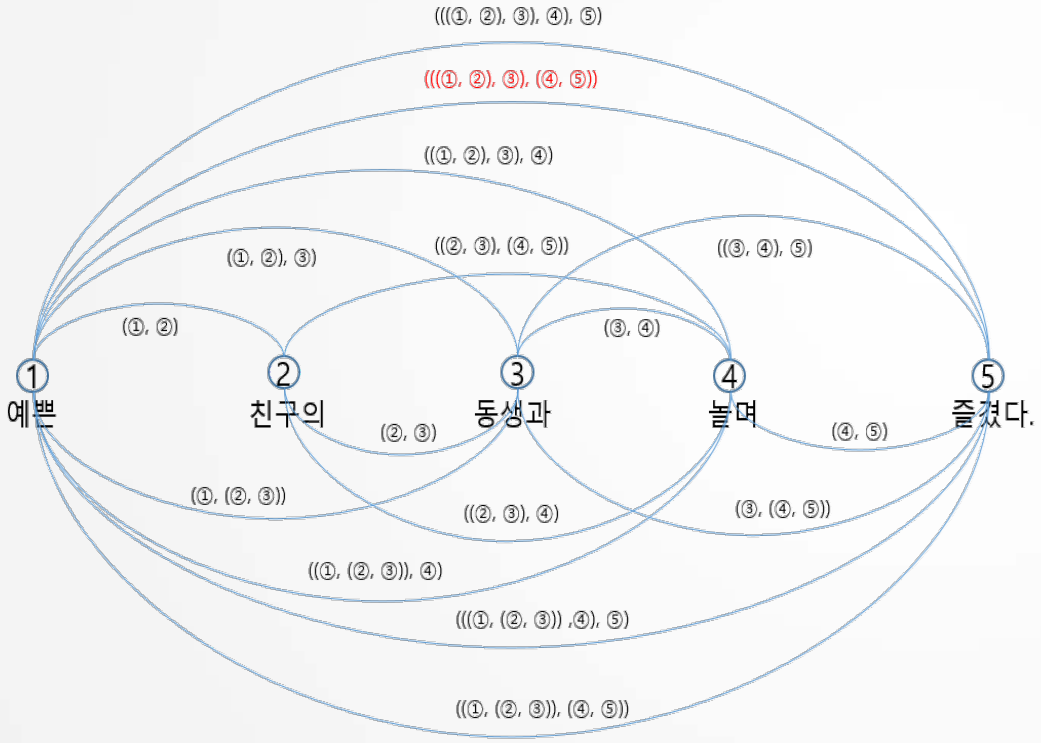


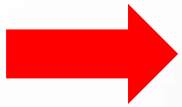
후보 파스 트리 1


(((①, ②), ③), ④), ⑤)
(((①, ②), ③), (④, ⑤))
((①, ②), ③), ④)
(①, ②), ③)
((②, ③), (④, ⑤))
((③, ④), ⑤)
(①, ②)
(③, ④)
1
예쁜
2
친구의
3
동생과
4
놀며
5
즐겼다.
(②, ③)
(④, ⑤)
(①, (②, ③))
((②, ③), ④)
(③, (④, ⑤))
((①, (②, ③)), ④)
(((①, (②, ③)) ,④), ⑤)
((①, (②, ③)), (④, ⑤))


ROOT
.(온점)
다(종결어미)
즐기다+었(타동사)
며(연결어미)
놀다+ㄹ(동사)
과(부사격조사)
동생(명사)
의(관형격조사)
친구(명사)
ㄴ(관형형전성어미)
예쁘다(형용사)

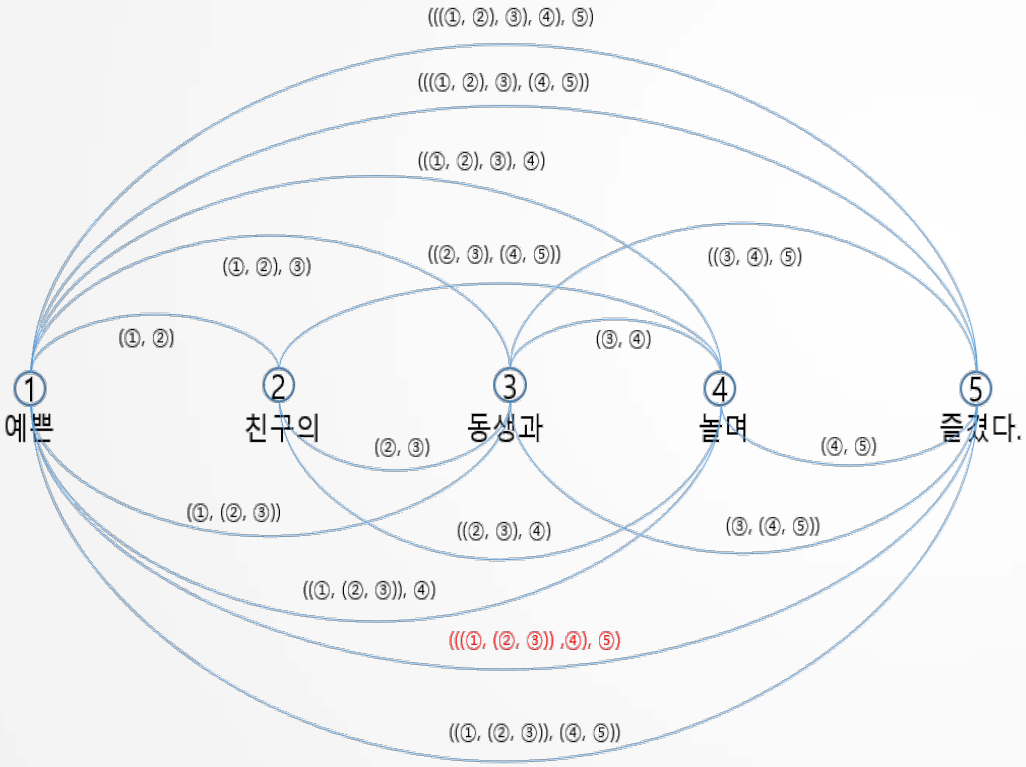
후보 파스 트리 2

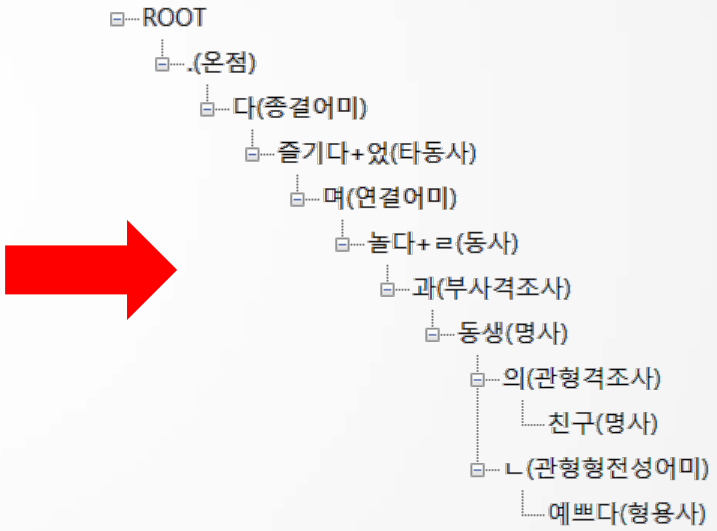

(((①, ②), ③), ④), ⑤)
(((①, ②), ③), (④, ⑤))
((①, ②), ③), ④)
(①, ②), ③)
((②, ③), (④, ⑤))
((③, ④), ⑤)
(①, ②)
(③, ④)
①
예쁜
②
친구의
③
동생과
④
놀며
⑤
즐겼다.
(②, ③)
(④, ⑤)
(①, (②, ③))
((②, ③), ④)
(③, (④, ⑤))
((①, (②, ③)), ④)
(((①, (②, ③)) ,④), ⑤)
((①, (②, ③)), (④, ⑤))



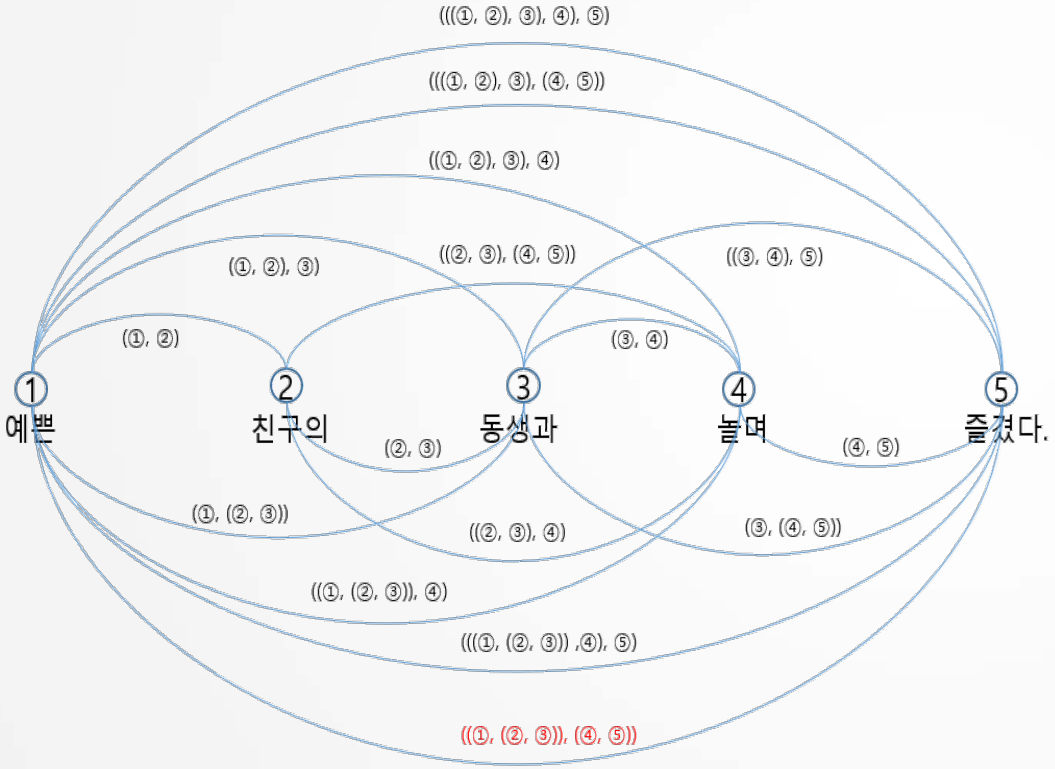

ROOT
.(온점)
다(종결어미)
즐기다+었(타동사)
며(연결어미)
놀다+ㄹ(동사)
과(부사격조사)
동생(명사)
의(관형격조사)
친구(명사)
ㄴ(관형형전성어미)
예쁘다(형용사)

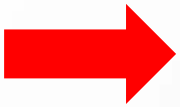
후보 파스 트리 3

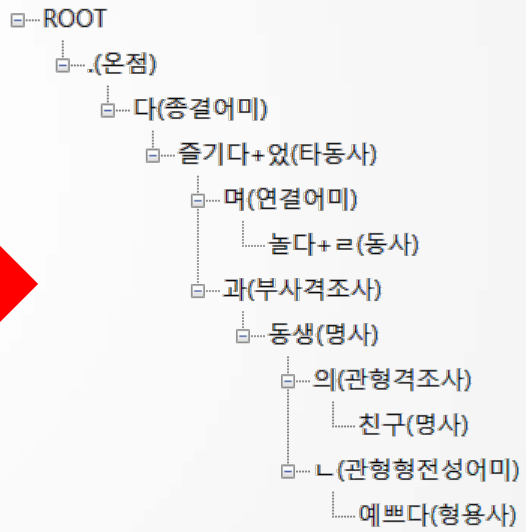

(((①, ②), ③), ④), ⑤)
(((①, ②), ③), (④, ⑤))
((①, ②), ③), ④)
(①, ②), ③)
((②, ③), (④, ⑤))
((③, ④), ⑤)
(①, ②)
(③, ④)
1
예쁜
2
친구의
3
동생과
4
놀며
5
즐겼다.
(②, ③)
(④, ⑤)
(①, (②, ③))
((②, ③), ④)
(③, (④, ⑤))
((①, (②, ③)), ④)
(((①, (②, ③)) ,④), ⑤)
((①, (②, ③)), (④, ⑤))


ROOT
.(온점)
다(종결어미)
즐기다+었(타동사)
며(연결어미)
놀다+ㄹ(동사)
과(부사격조사)
동생(명사)
의(관형격조사)
친구(명사)
ㄴ(관형형전성어미)
예쁘다(형용사)

후보 파스 트리 4


(((①, ②), ③), ④), ⑤)
((①, ②), ③), (④, ⑤))
((①, ②), ③), ④)
(①, ②), ③)
((②, ③), (④, ⑤))
((③, ④), ⑤)
(①, ②)
(③, ④)
①
예쁜
②
친구의
③
동생과
④
놀며
⑤
즐겼다.
(②, ③)
(④, ⑤)
(①, (②, ③))
((②, ③), ④)
(③, (④, ⑤))
((①, (②, ③)), ④)
(((①, (②, ③)) ,④), ⑤)
((①, (②, ③)), (④, ⑤))

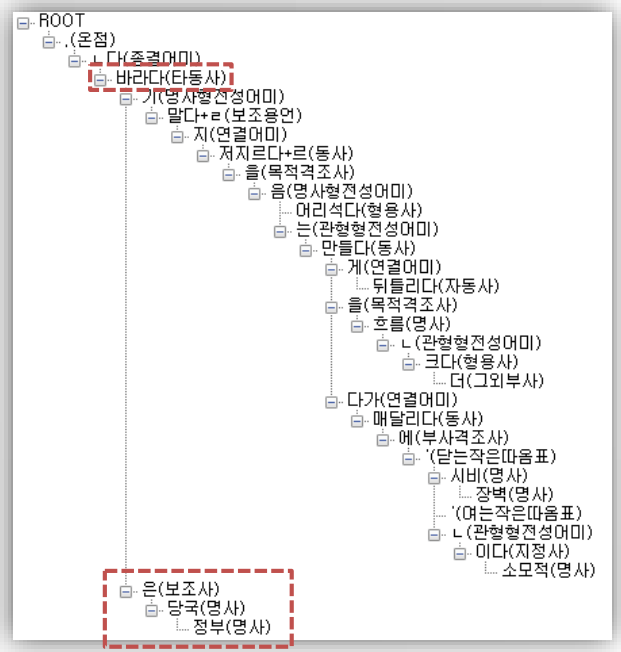



ROOT
.(온점)
다(종결어미)
즐기다+었(타동사)
며(연결어미)
놀다+ㄹ(동사)
과(부사격조사)
동생(명사)
의(관형격조사)
친구(명사)
ㄴ(관형형전성어미)
예쁘다(형용사)

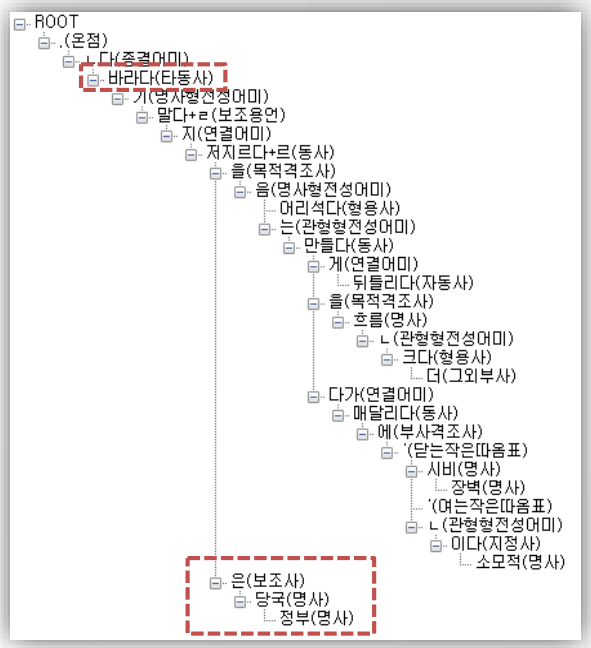
한국어 문장 분석과 tree-bank를 만들기 어려운 이유(구문 중의성)

<예문 1>

정부 당국은 소모적인 '장벽 시비'에 매달리다가, 더 큰 흐름을 뒤틀리게 만드는 어리석음을 저지르지 말기 바란다.



후보 파스 트리 1
(정부 당국은 – 바라다)

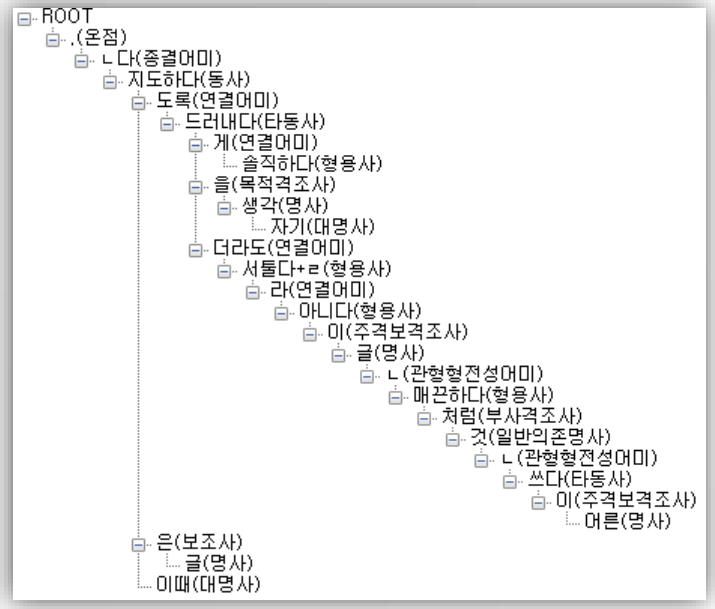


후보 파스 트리 2
(정부 당국은 – 저지르다)

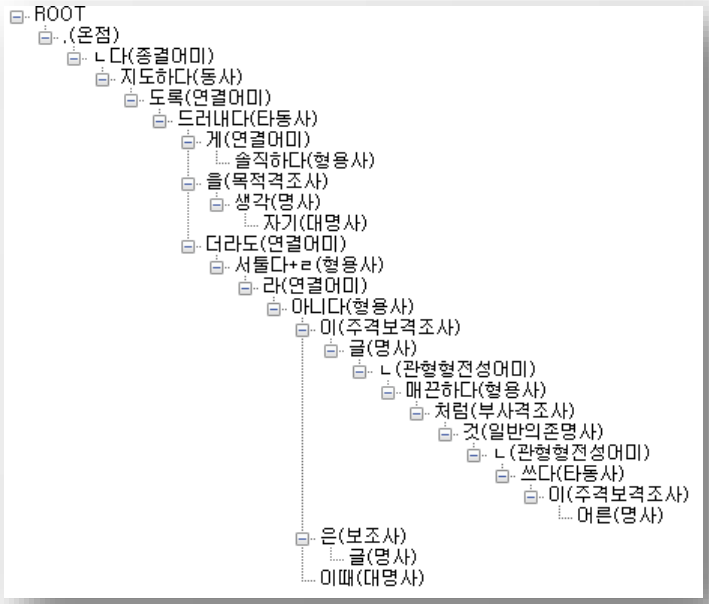
구문 중의성 예시

<예문 2>

이 때 글은 어른이 쓴 것처럼 매끈한 글이 아니라 서툴더라도 자기 생각을 솔직하게 드러내도록 지도한다.



후보 파스 트리 1



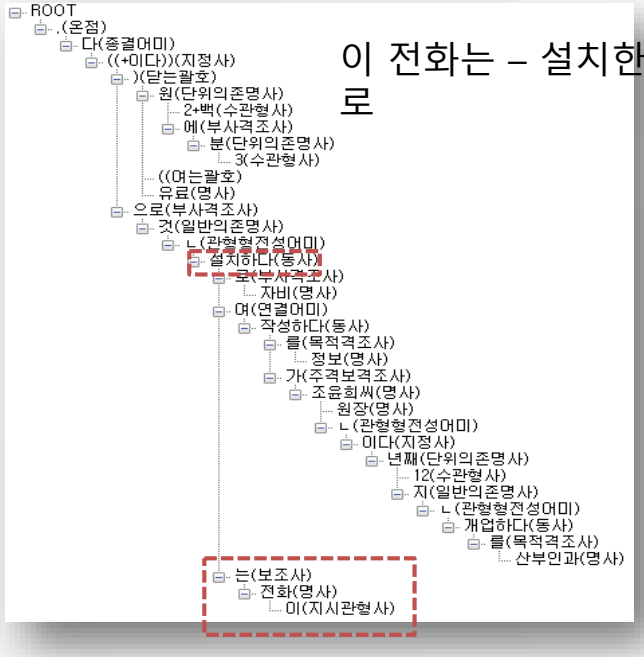
후보 파스 트리 2

문장 자체가 이상

구문 중의성 예시

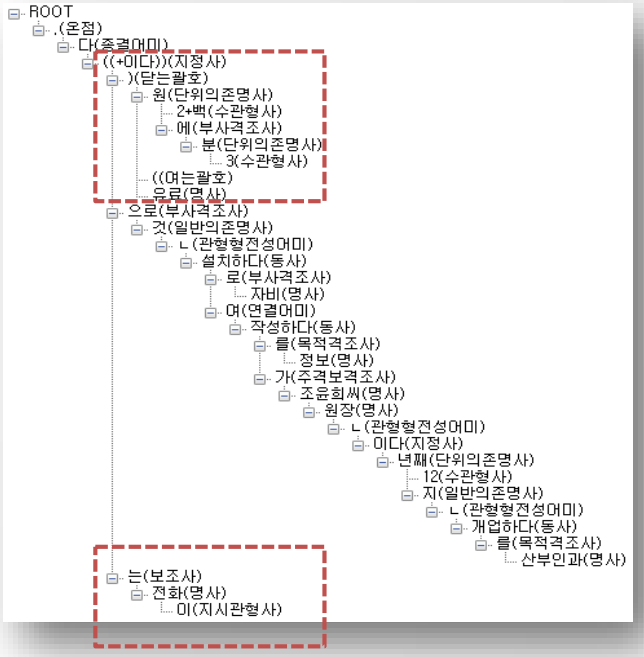
<예문 3>

이 전화는 산부인과를 개업한 지 12년째인 원장 조윤희 씨가 정보를 작성해 자비로 설치한 것으로 유료(3분에 2백원)다.



이 전화는 – 설치한 것으로

후보 파스 트리 1

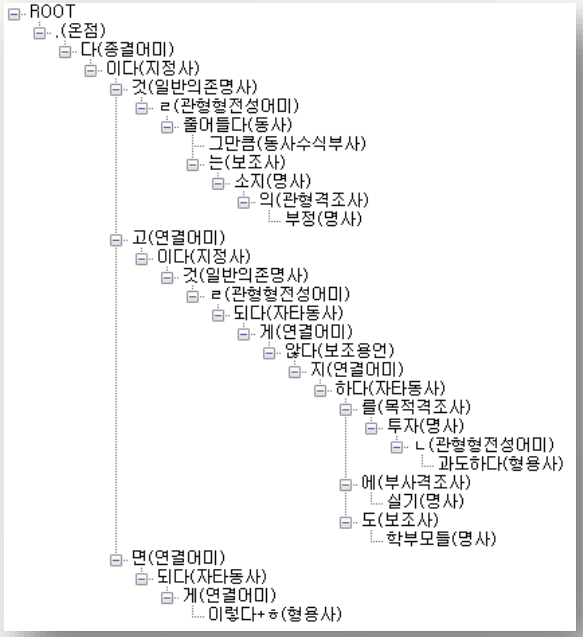


후보 파스 트리 2

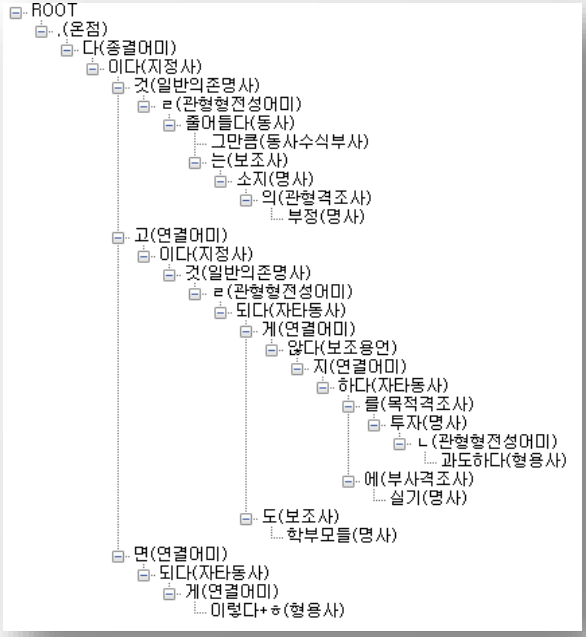
구문 중의성 예시

<예문 4>

이렇게 되면 학부모들도 실기에 과도한 투자를 하지 않게 될 것이고 부정의 소지는 그만큼 줄어들 것이다.



후보 파스 트리 1



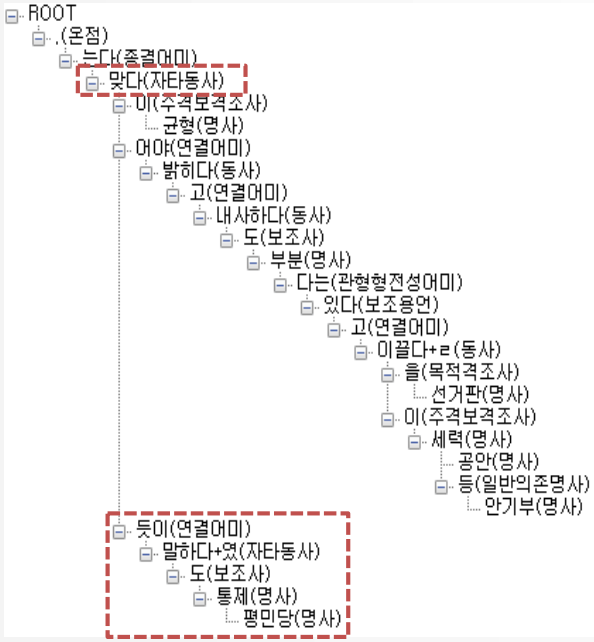
후보 파스 트리 2

"~하지 않게 될 것이고" ->
보조사구("학부모들도")가 "하다"에 구문이
연결 되야 할 것인지, "되다"에 구문이
연결 되야 할 것인지에 대한 중의성

구문 중의성 예시

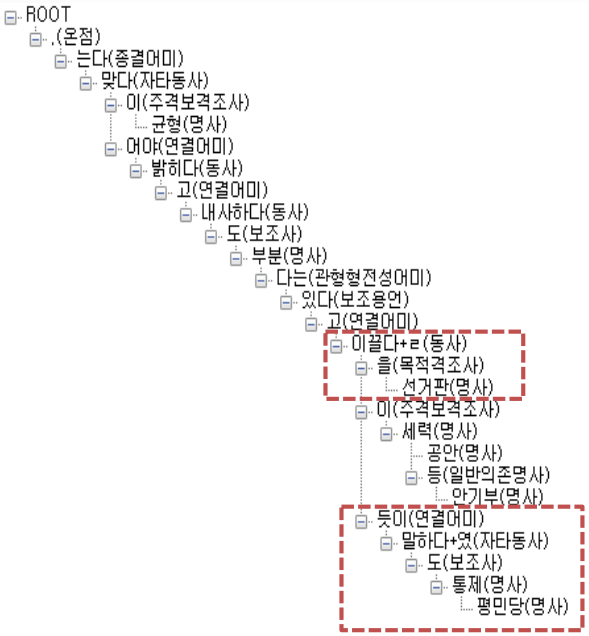
<예문 5>

평민당 통재도 말했듯이 안기부 등 공안 세력이 선거판을 이끌고 있다는 부분도 내사하고 밝혀야 균형이 맞는다.



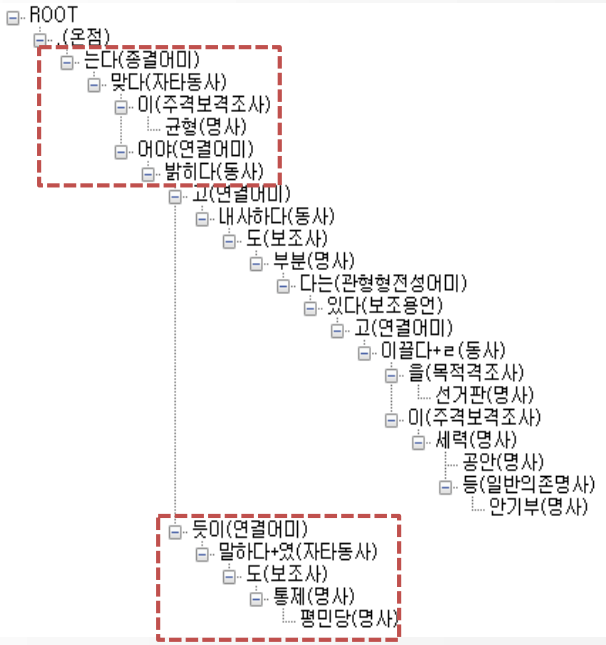
후보 파스 트리 1

말했듯이 - 맞다



후보 파스 트리 2

말했듯이 - 이끌다



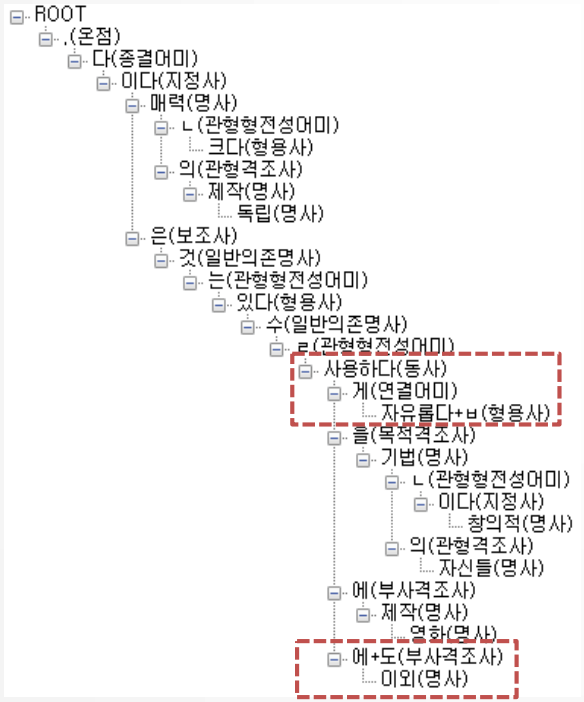
후보 파스 트리 3

말했듯이 - 밝히다

구문 중의성 예시

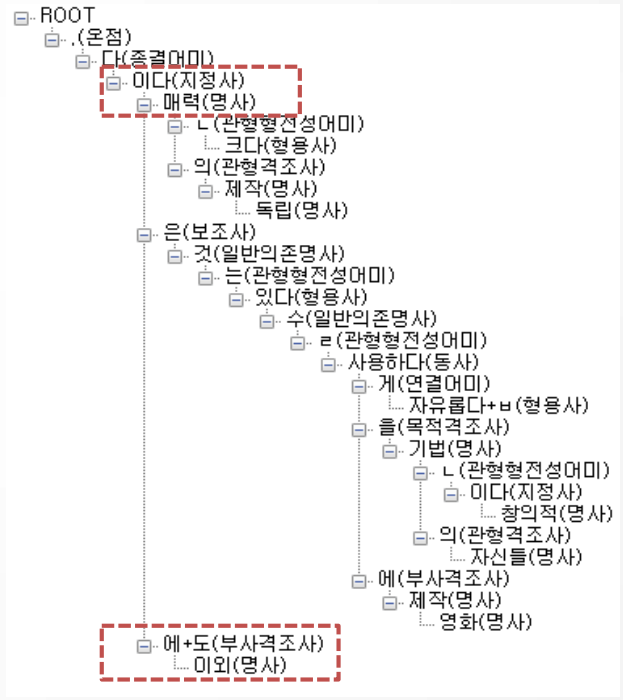
<예문 6>

이외에도 영화제작에 자신들의 창의적인 기법을 자유롭게 사용할 수 있는 것은 독립 제작의 큰 매력이다.



후보 파스 트리 1

이외에도 - 사용하다



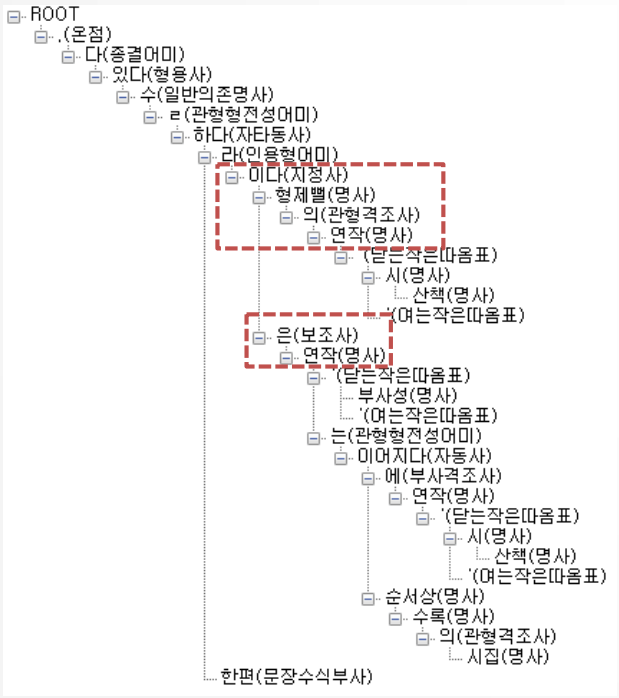
후보 파스 트리 2

이외에도 - 매력이다

구문 중의성 예시

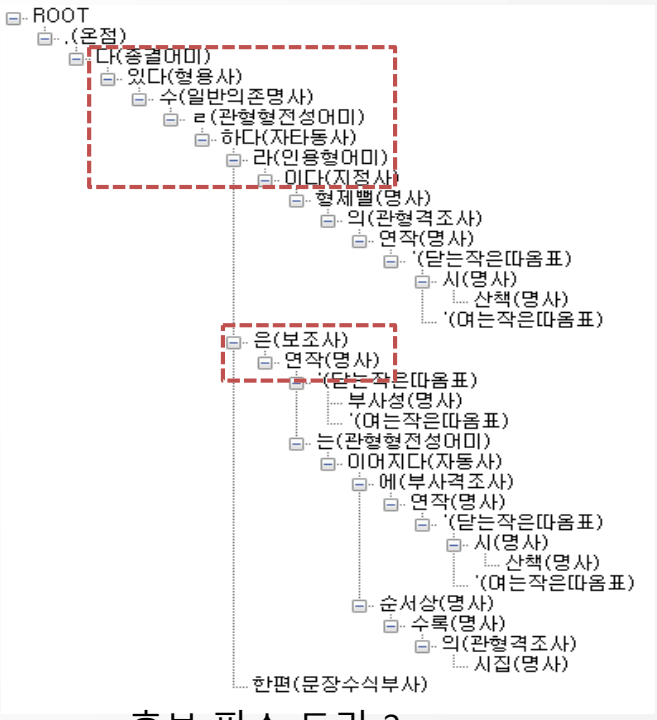
<예문 7>

한편 시집의 수록 순서상 '산책시' 연작에 이어지는 '부사성' 연작은 '산책시' 연작의 형제뻘이라 할 수 있다.



후보 파스 트리 1

연작은 - 형제뻘이다



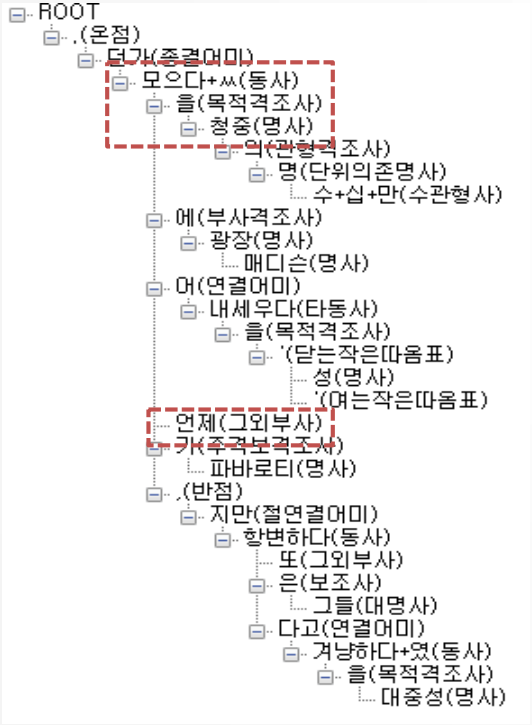
후보 파스 트리 2

연작은 - ~이라 할 수 있다

구문 중의성 예시

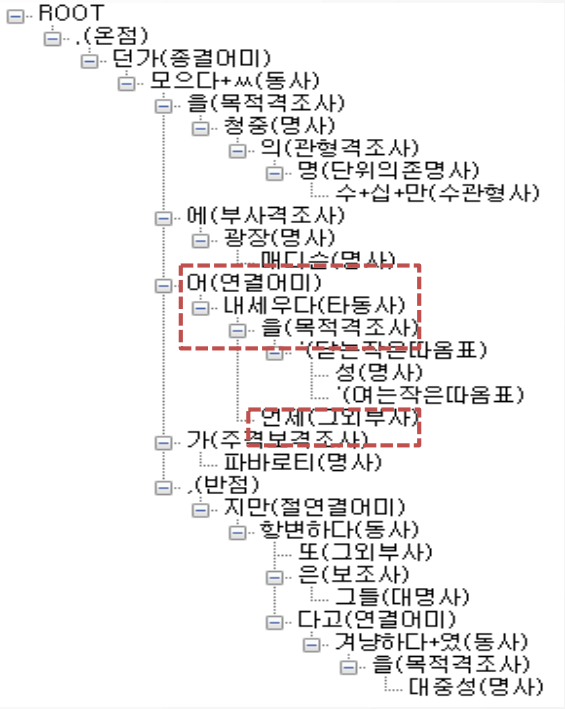
<예문 8>

*대중성을 겨냥했다고 그들은 또 항변하지만 파바로티가 언제 '성'을 내세워 메디슨 광장에 수십 만명의 청중을 모았던가



후보 파스 트리 1

언제 - 모았던가



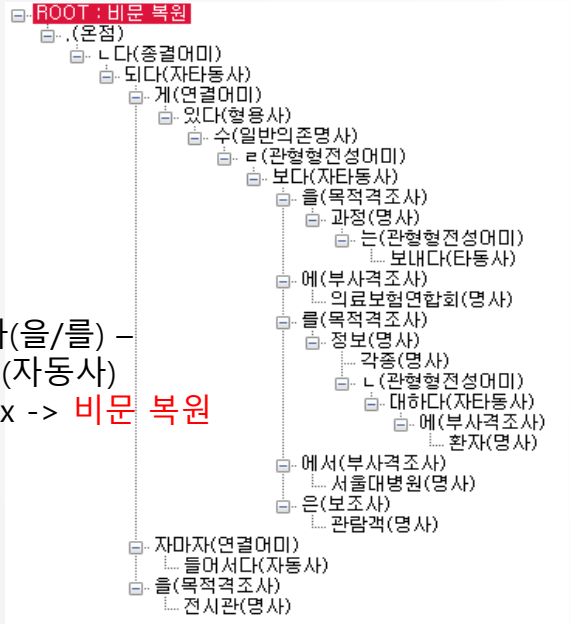
후보 파스 트리 2

언제 - 내세워

구문 수정 예시

<예문 9>

전시관을 들어서자마자 관람객은 서울대병원에서 환자에 대한 각종 정보를 의료보험연합회에 보내는 과정을 볼 수 있게 된다.



목적격조사(을/를) – "들어서다"(자동사) 구문 연결 x -> 비문 복원

ParseTree Case 1

수정 후

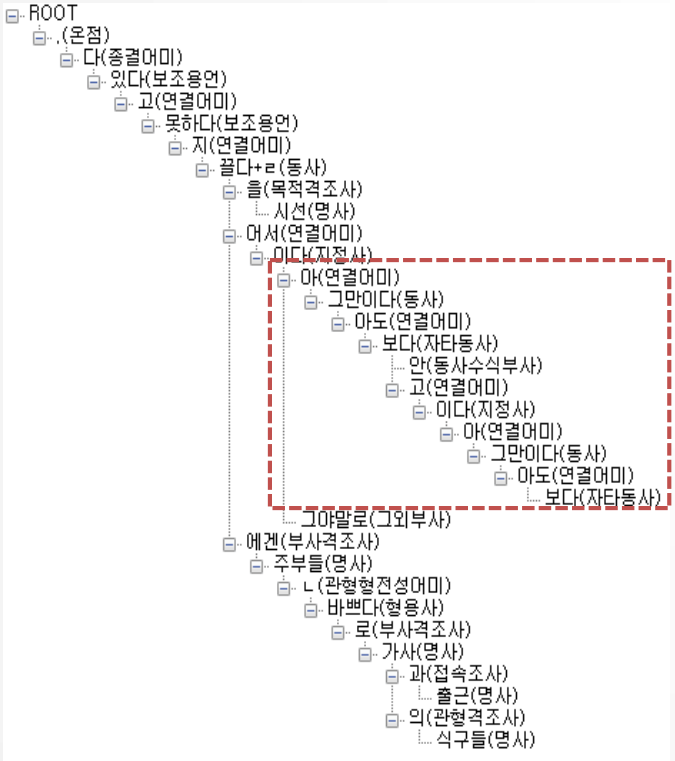
목적격 조사 – 자동사 구문에서 그 뒤 어절이 연결어미인 "-**자마자**"가 올 경우 구문 연결 허용

ParseTree Case 2

구문 수정 예시

<예문 10>

식구들의 출근과 가사일로 바쁜 주부들에겐 그야말로 보아도 그만이고 안 보아도 그만이어서 시선을 끌지 못하고 있다.

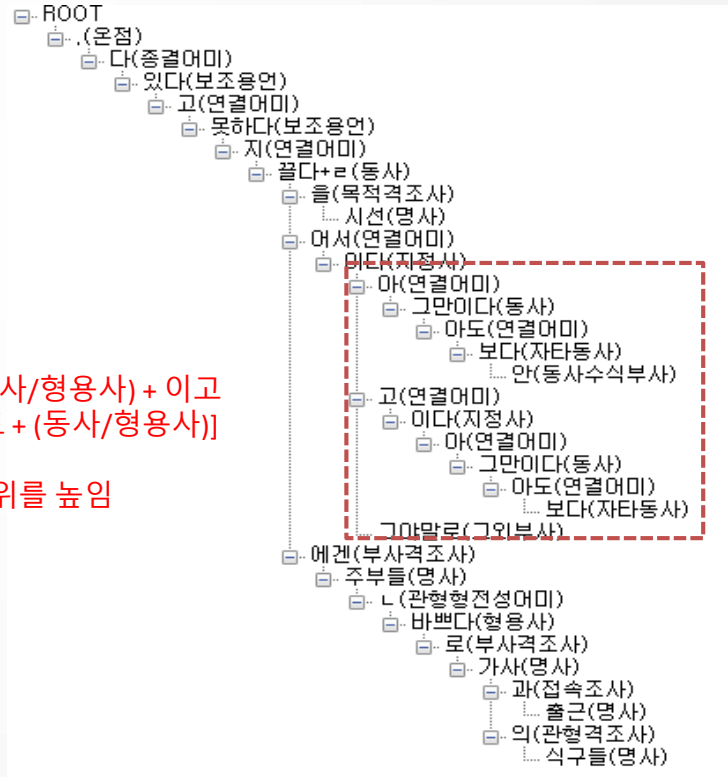


ParseTree Case 1

수정 후

[(동사) + 아도/어도 + (동사/형용사) + 이고 + 안 + (동사) + 아도/여도 + (동사/형용사)]

-> 후보 우선 순위를 높임



ParseTree Case 2

가중치 부여

- 인용형어미 – 대화 동사 가중치 부여
  - Ex) 배고프다고 말한 여자친구와 맛집을 찾는다.
    - (다고 – 말하다) > (다고 – 찾는다)
- Topical phrase – 본용언 가중치 부여
  - Ex) 나는 배가 고프지만 참았다.
    - (나는 – 참았다) > (나는 – 고프다)
- 관형형 – 명사 가중치 부여
  - Ex) 힘든 한국사 공부
    - (힘든 – 공부) > (힘든 – 한국사)
- 격조사 구 관련 가중치 부여
  - Ex) 파리에서 본 에펠탑을 잊을 수 없다.
    - (에서 – 본) > (에서 – 잊다)
- 의존관계 거리 순 가중치 부여
  - 전체 의존관계 거리가 짧은 파스 트리에 가중치를 부여한다.

문장 분석 후보를 위한 가중치

$$ptw_i = W^i_{pos} + W^i_{distance} + W^i_{sr}$$

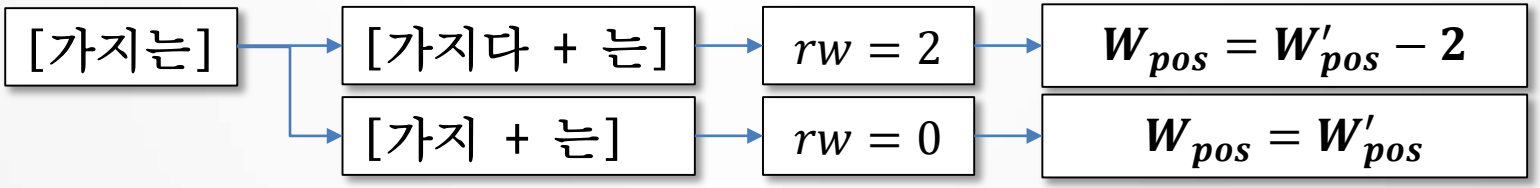
- W_{pos} : 형태소 분석기의 어절 별 가중치
- $W_{distance}$: 거리에 따른 가중치
- W_{sr} : 특수 규칙에 따른 가중치

형태소 분석기의 어절 별 가중치 이용

- 품사 태거의 규칙 가중치(rule weight, rw)를 이용해 형태소 별 가중치 부여 [2]
  - 하나의 어절에 대해 다양한 형태소 분석 후보가 존재할 때, 발생할 확률이 가장 높은 형태소 후보가 더 높은 가중치를 가짐.

$$W_{pos} = -\left(rw(w_1^{\alpha}) + rw(w_2^{\beta}) + \cdots + rw(w_n^{\gamma})\right)$$

  - Ex) 그리고 헌법은 국가는 개인이 *가지는* 불가침의 기본적 인권을 확인하고 이를 보장할 의무를 진다고 규정하고 있습니다.



$$W'_{pos} = -\left(\sum_{i=0}^{3} rw(w_i) + \sum_{j=5}^{15} rw(w_j)\right)$$

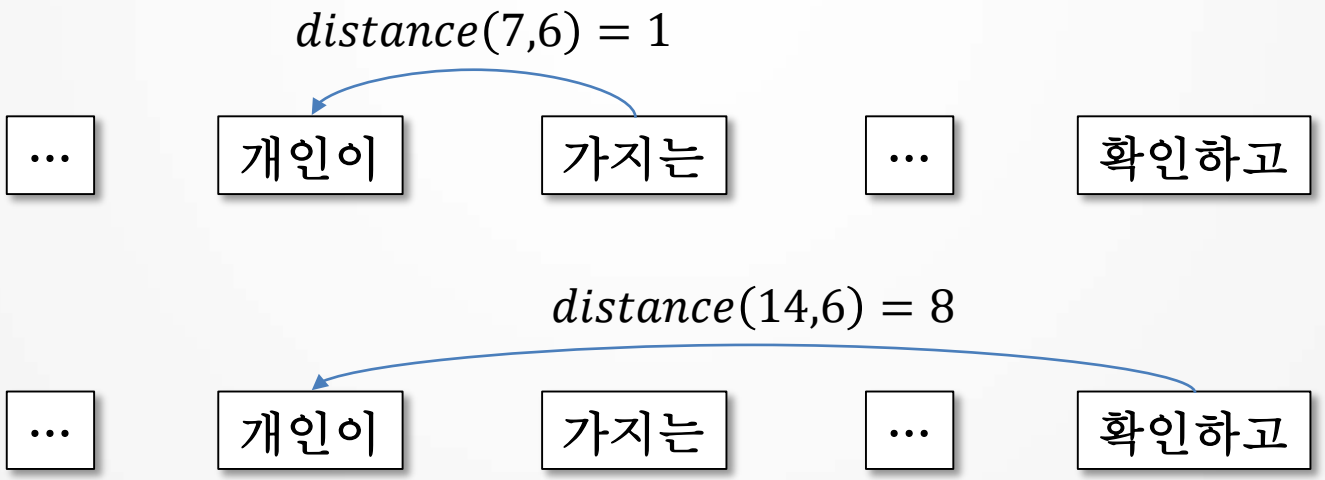
[2] 황명진, 강미영, 권혁철, "규칙과 어절 확률을 이용한 혼합 품사 태깅 모델," 한국정보과학회 학술발표논문집, 33(2B), pp.11-15, 2006

수식 거리에 따른 가중치 부여

- 한국어의 특징상 수식 거리가 1인 의존소가 많이 존재함 [3]

$$W_{distance} = \sum_{\{H,T\} \in PT} Distance(H,T)$$

- Ex) 그리고 헌법은 국가는 *개인이*[6] *가지는*[7,8] 불가침의[9] 기본적[10] 인권을[11,12] *확인하고*[13,14] 이를 보장할 의무를 진다고 규정하고 있습니다.

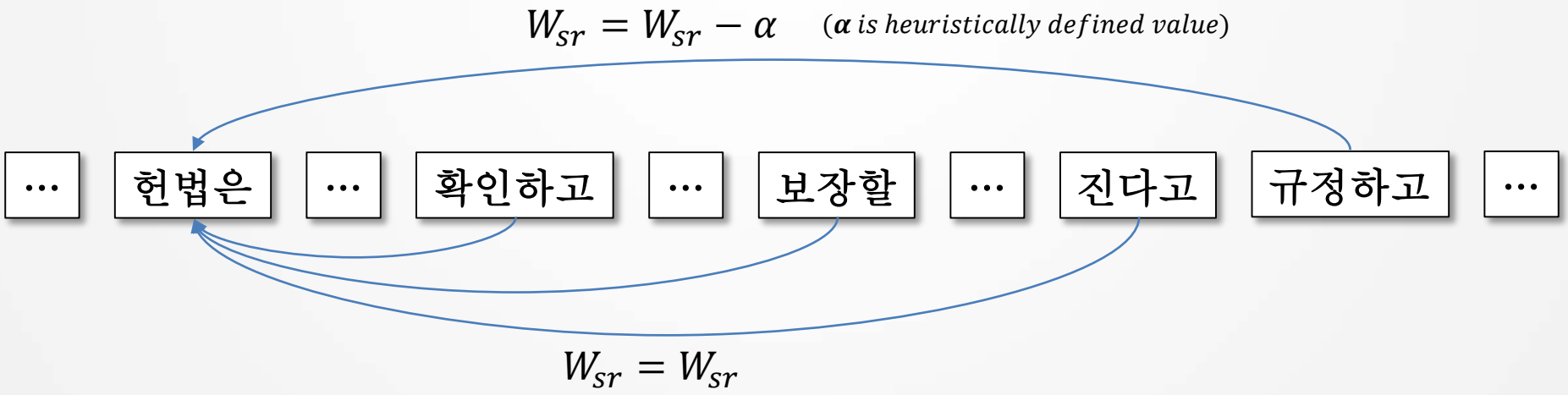


[3] 우연문, 송영인, 박소영, 임해창, "지배가능 경로 문맥을 이용한 의존 구문 분석의 수식 거리 모델," 정보과학회논문지 : 소프트웨어 및 응용, 34(2), pp. 140-149, 2007

특수 규칙을 통한 의존관계 가중치 부여

- 원거리 의존관계 반영 및 특정 패턴에 대한 가중치 부여
  - 보조사구-본용언 우선 규칙
  - 관형형-명사 나열형 가중치 규칙
  - 문장수식부사-본용언 우선 규칙
  - 격조사구 관련 규칙

  - Ex) 그리고 *헌법은* 국가는 개인이 *가지는* 불가침의 기본적 인권을 *확인하*고 이를 *보장할* 의무를 *진다*고 *규정하*고 있습니다.



문장 분석 시스템 성능

평가데이터

- 형태소 단위로 구축된 평가 데이터가 존재하지 않음
- "세종 형태 분석 말뭉치"에서 임의 추출하여 정답을 구축 (106문장)
- 최소 13어절, 최대 16어절, 평균 15어절

평가 방법 - Unlabeled Attachment Score (UAS)

$$UAS = \frac{\text{시스템이 올바르게 찾은 의존관계의 개수}}{\text{평가 데이터의 모든 의존관계 개수}}$$

모델별 문장 분석 성능 비교

	가중치 적용 기준	Rank 1이 정답인 문장 수	정답 트리의 평균 순위	UAS (%)
Model 1	*None*	3	72.48	87.86%
Model 2	*Model 1 + W_{pos}*	5	29.30	89.99%
Model 2	*Model 2 + $W_{distance}$*	9	23.51	92.04%
Model 4	*Model 3 + W_{sr}*	19	19.36	**93.34%**

향후 연구 내용

규칙 추가를 이용한 성능향상 방안

- 규칙 기반 구문 분석기이므로 문법적으로 허용되지 않는 문장에 대해서도 의존관계를 가질 수 있도록 규칙을 추가하여 성능을 향상시켜야 함
  - ex) 신혼여행은 발리로

문형 정보를 이용한 성능향상 방안

- 세종상세전자사전, 표준국어대사전의 용언에 포함된 문형 정보를 이용해 해당 문형 관계를 가진 에지에 가중치를 부여하여 구문 분석기 성능을 향상시킴
  - ex) (인간)이 (인간)을 **따르다** / (인간)이 (용기)에 (음료)를 **따르다**

통계 정보를 이용한 성능향상 방안

- 세종 계획 구문분석 말뭉치에 있는 의존관계를 통계 데이터로 이용하여 생성되는 에지 별 통계적 가중치를 부여하여 구문 분석기 성능을 향상시킴

기계 학습을 이용한 성능향상 방안

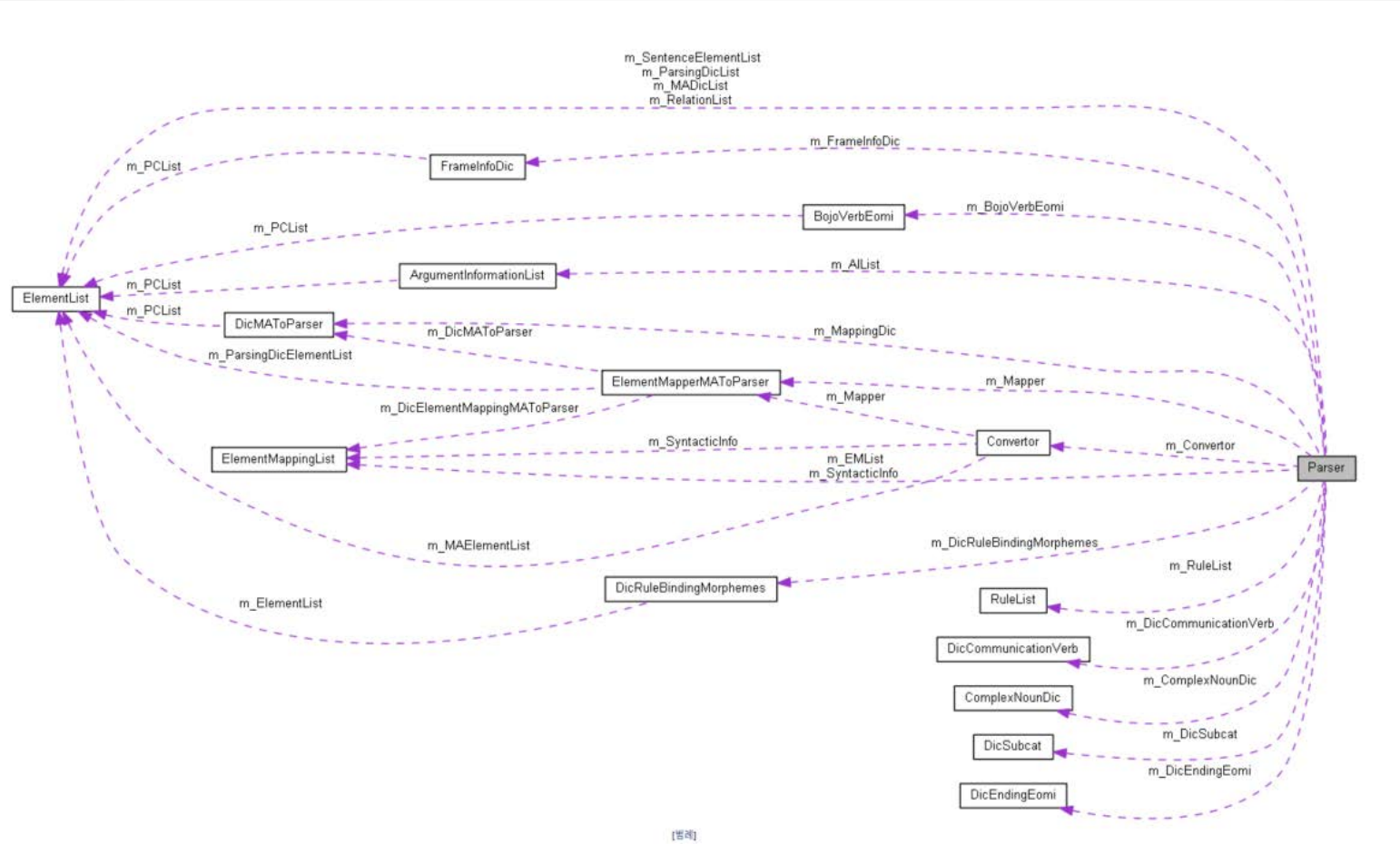
- 기계 학습을 이용해 수작업으로 찾아낼 수 없는 규칙이나 패턴을 찾고 이에 대한 규칙 추가 및 가중치 부여를 수행함

실습

- 개발환경
  - IDE: Microsoft Visual Studio Community 2017
  - 언어: C++
- 실습자료
  - 형태소 분석기 – 동적 라이브러리(PnuNlpCore.dll)
  - 품사 태거 – 동적 라이브러리(KLTagger.dll)
  - 구문 분석기 – 소스코드
- 실습방법
  - 몇 가지 분석 오류를 제시하고 그것을 고치는 과정을 설명함

주요 클래스 - Parser

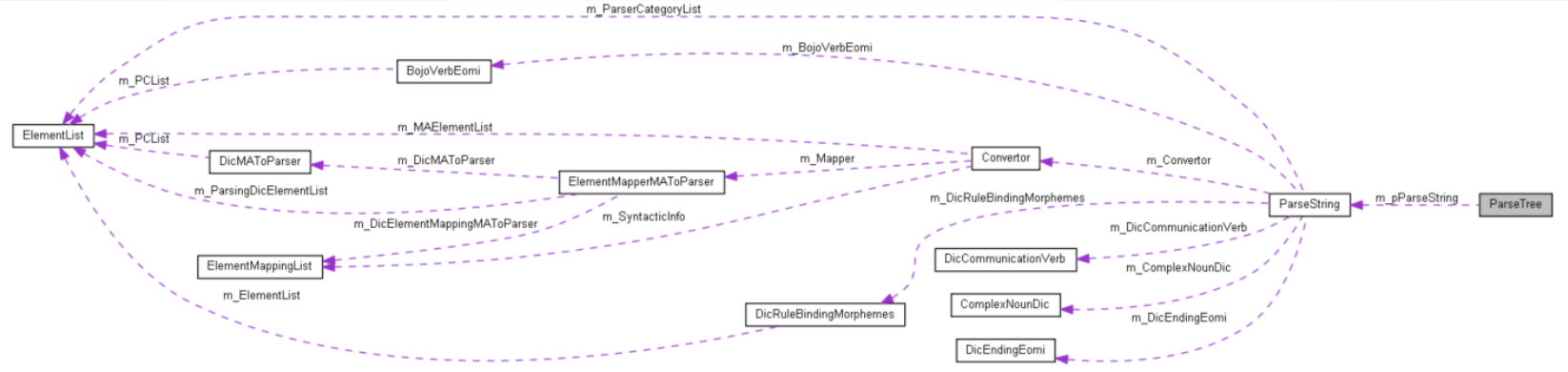
- 구문 분석의 주체가 되는 최상위 클래스

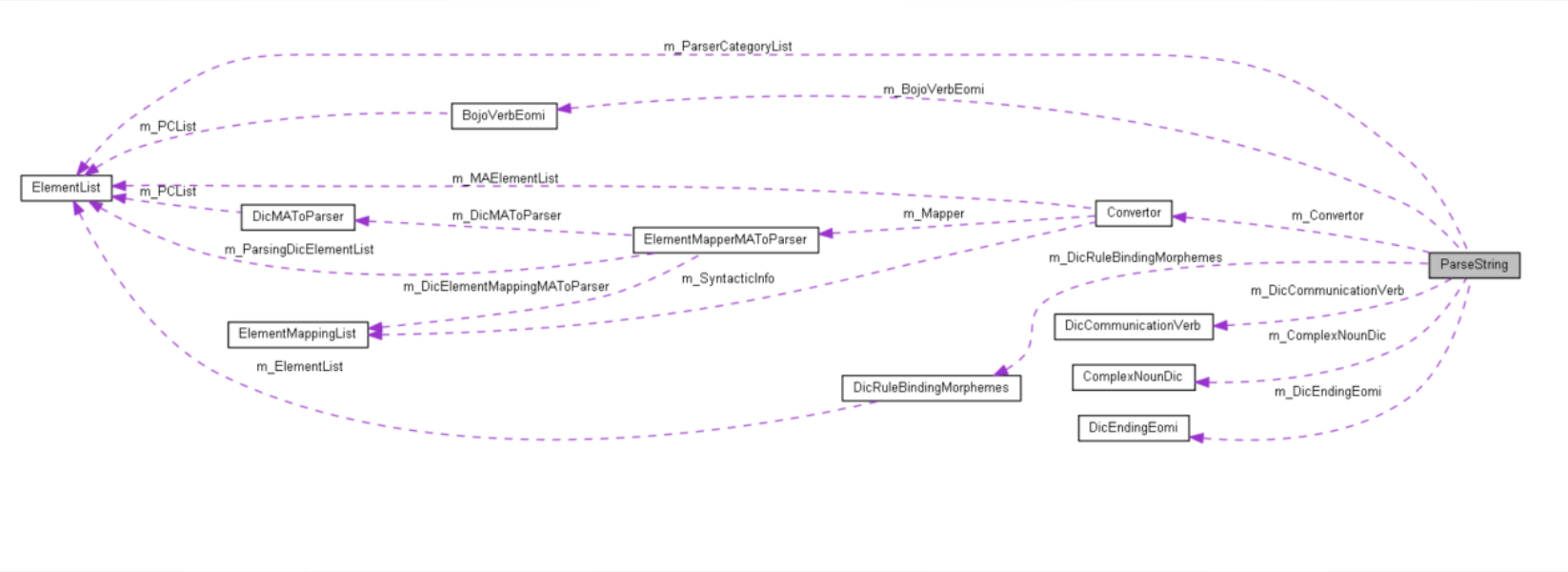


[범례]

주요 클래스 - ParseTree

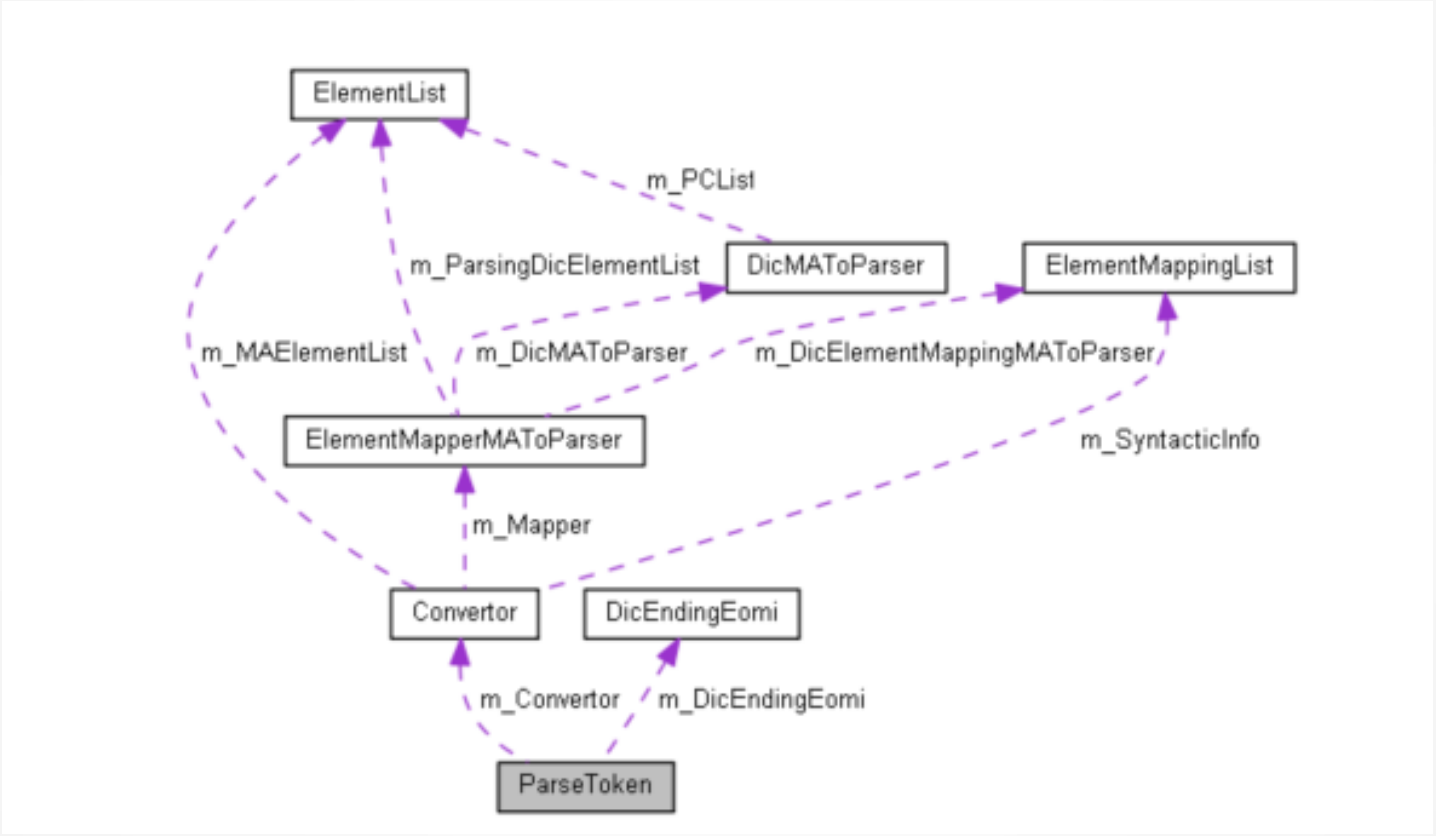
- 구문 분석 결과를 저장하기 위한 클래스





주요 클래스 - ParseToken

- 한 어절에 대한 정보를 저장하기 위한 클래스



문장 분석 시스템 활용

내부 연결을 위한 자료구조

```cpp
#pragma once

#include <string>
#include <vector>

#include "Parser.h"
using namespace std;

// 단일 형태소 하나를 저장하는 구조체
typedef struct
{
    int id_; //MorphemeInfo 순번. 0부터 시작
    int tab_; //ParseTree에서의 Level
    string word_; //어휘
    string pos_; //품사
} MorphemeInfo;

// 어절에 대한 형태소 분석 정보를 저장하는 구조체
typedef struct
{
    int id_; //WordInfo 순번. 0부터 시작
    vector<MorphemeInfo> morpheme_Info_List_Word_; //형태소 결합 정보. 해당 어절의 형태소 분석 정보만 저장
} WordInfo;

// 형태소(지배소, head)와 형태소(의존소, dependent) 간 의존관계를 저장하는 구조체
typedef struct
{
    int id_; //DependencyInfo 순번. 0부터 시작
    MorphemeInfo dependent_; //의존소
    MorphemeInfo head_; //지배소
} DependencyInfo;

// 용언과 논항의 관계를 저장하는 구조체
typedef struct
{
    int id_; //PredicateInfo 순번. 0부터 시작
    MorphemeInfo predicate_; // 용언
    vector<MorphemeInfo> argument_list_; // 논항
} PredicateInfo;
```

JSON Format


```
{
"text": "늦은 밤 한 남자가 해운대 바닷가를 홀로 걷고 있다.",
"word_list": [
        {"id" : 0, "word" : "늦은"},
        {"id" : 1, "word" : "밤"},
        {"id" : 2, "word" : "한"},
        {"id" : 3, "word" : "남자가"},
        {"id" : 4, "word" : "해운대"},
        {"id" : 5, "word" : "바닷가를"},
        {"id" : 6, "word" : "홀로"},
        {"id" : 7, "word" : "걷고"},
        {"id" : 8, "word" : "있다"},
        {"id" : 9, "word" : "."}
 ],
"morpheme_list": [
        {"id" : 0, "morpheme" : "늦다", "POS": "형용사", "word_id" : 0},
        {"id" : 1, "morpheme" : "은", "POS": "관형형전성어미", "word_id" : 0},
        {"id" : 2, "morpheme" : "밤", "POS": "일반명사", "word_id" : 1},
        {"id" : 3, "morpheme" : "한", "POS": "일반관형사", "word_id" : 2},
        {"id" : 4, "morpheme" : "남자", "POS": "일반명사", "word_id" : 3},
        {"id" : 5, "morpheme" : "가", "POS": "주격보격조사", "word_id" : 3},
        {"id" : 6, "morpheme" : "해운대", "POS": "고유명사", "word_id" : 4},
        {"id" : 7, "morpheme" : "바닷가", "POS": "일반명사", "word_id" : 5},
        {"id" : 8, "morpheme" : "를", "POS": "목적격조사", "word_id" : 5},
        {"id" : 9, "morpheme" : "홀로", "POS": "일반부사", "word_id" : 6},
        {"id" : 10, "morpheme" : "걷다", "POS": "자타동사", "word_id" : 7},
        {"id" : 11, "morpheme" : "고", "POS": "연결어미", "word_id" : 7},
        {"id" : 12, "morpheme" : "있다", "POS": "보조용언", "word_id" : 8},
        {"id" : 13, "morpheme" : "다", "POS": "종결어미", "word_id" : 8},
        {"id" : 14, "morpheme" : ".", "POS": "온점", "word_id" : 9}
],
"dependency": [
        {"id" : 0, "dependent" : "늦다", "head": 1},
        {"id" : 1, "dependent" : "은", "head": 2},
        {"id" : 2, "dependent" : "밤", "head": 10},
        {"id" : 3, "dependent" : "한", "head": 4},
        {"id" : 4, "dependent" : "남자", "head": 5},
```

Artficial Intelligence Laboratory

감사합니다

이 연구는 삼성전자 미래기술육성센터의 지원을
받아 수행된 연구임(SRFC-IT1402-03)

부산대학교
PUSAN NATIONAL UNIVERSITY