

2019 winter course 수업 기준 필기.

수업 목차

Linguistic Structure: Dependency parsing

1. Syntactic Structure: Consistency and Dependency
2. Dependency Grammar and Treebanks
3. Transition-based dependency parsing
4. Neural dependency parsing

## 1. Consistency and Denpendency Parsing

**Parsing**이란?

: 각 문장의 grammatical structure, phrase 를 분석하는 과정 (output - 구문분석 tree 형태)

**Consistency Parsing**

: 문장을 중첩된(nested) 성분으로 나누어서 문장의 구조를 파악한다. 영어처럼 어순이 고정적인 언어에서 주로 사용된다.

문법적 구조에 따라서 문장으로부터 constituency(consistency)-based parse tree 를 추출하는 것이 주된 목표다.

문장 성분으로 분류된 words 가 주어지고, 같은 분류 기준에 따라 묶인 word 는 phrase 를 구성한다. 이 phrases 는 서로 결합되어 더 큰 phrase 를 재귀적으로 만들 수 있다.

**Starting unit: words are given a category (part of speech = pos)**

the, cat, cuddly, by, door  
Det N Adj P N

**Words combine into phrases with categories**

the cuddly cat, by the door  
NP → Det Adj N PP → P NP

**Phrases can combine into bigger phrases recursively**

the cuddly cat by the door  
NP → NP PP

Grammar	Lexicon
$S \rightarrow NP VP$	$Det \rightarrow that   this   the   a$
$S \rightarrow Aux NP VP$	$Noun \rightarrow book   flight   meal   money$
$S \rightarrow VP$	$Verb \rightarrow book   include   prefer$
$NP \rightarrow Pronoun$	$Pronoun \rightarrow I   she   me$
$NP \rightarrow Proper-Noun$	$Proper-Noun \rightarrow Houston   NWA$
$NP \rightarrow Det Nominal$	$Aux \rightarrow does$
$Nominal \rightarrow Noun$	$Preposition \rightarrow from   to   on   near   through$
$Nominal \rightarrow Nominal Noun$	
$Nominal \rightarrow Nominal PP$	
$VP \rightarrow Verb$	
$VP \rightarrow Verb NP$	
$VP \rightarrow Verb NP PP$	
$VP \rightarrow Verb PP$	
$VP \rightarrow VP PP$	
$PP \rightarrow Preposition NP$	

**Figure 13.1** The  $\mathcal{L}_1$  miniature English grammar and lexicon.

## Dependency Parsing

: 단어 간의 의존관계 혹은 수식관계를 파악한다. (전체적인 문장 구조보다는 word relationships 에 집중) 한국어와 같이 자유 어순을 갖거나, 문장성분이 생략 가능한 언어에서 선호한다.

## 2. Dependency Grammar and Treebanks

(Human language 와 programming language 의 차이 - Programming languages have hard rules as to how you meant to interpret things that dangle afterwards)

- Prepositional phrase(PP) attachment ambiguity 문제 발생

- Coordination scope ambiguity

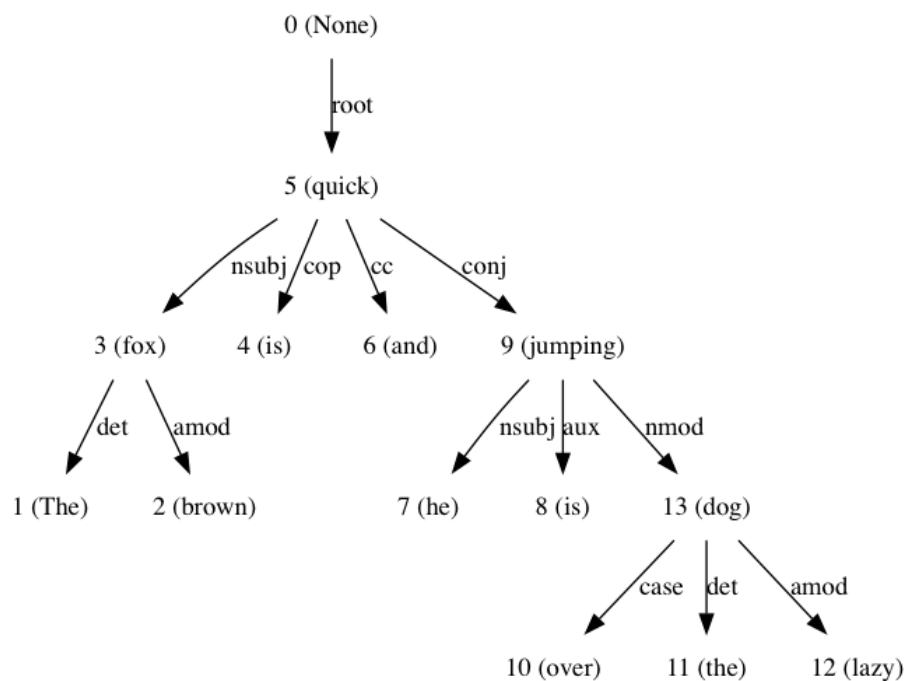
- Adjectival Modifier Ambiguity

- Verb Phrase(VP) attachment ambiguity

다양하게 발생하는 ambiguity 문제 --> Parsing 이 필요하다!

Parsing 을 통해 문장 내 성분들이 정확히 어떻게, 어디에 연결되어 있는지 이해할 필요가 있다.

### Dependency grammar



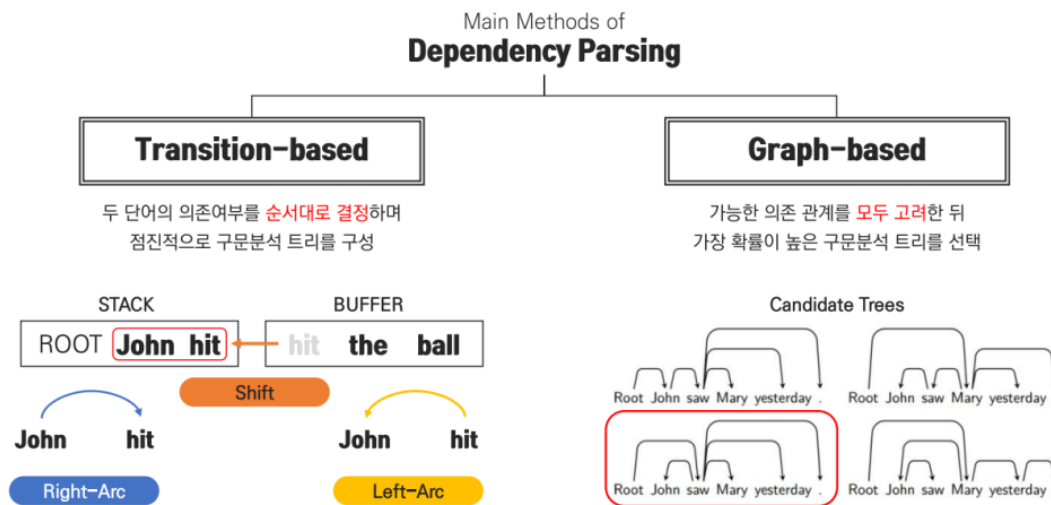
[how to get a dependency tree with Stanford NLP parser - Stack Overflow](#)

Dependency Structure 는 위 그림과 같이 표현할 수 있다.

화살표는 head(governor, superior, regent)에서 dependent(modifier, inferior, subordinate)로 향하며, 화살표 위의 label 은 단어 간의 문법적 관계(dependency)를 의미한다.

여기서 'Root'는 어떠한 단어의 수식도 받지 않는 단어에 붙는 가상의 노드인데, 이를 통해서 모든 단어가 최소한 1 개의 node 의 dependent 가 되도록 만든다.

## Methods of Dependency Parsing

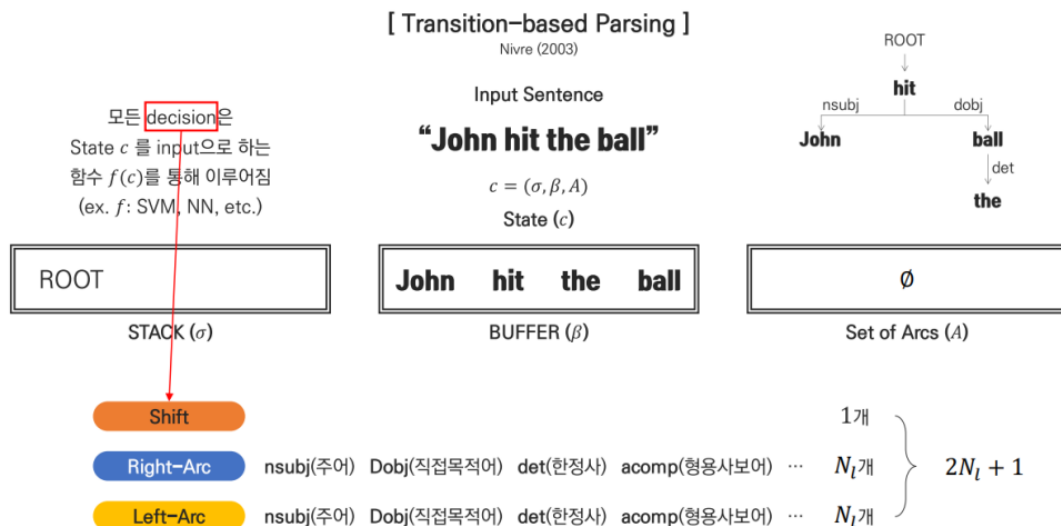


출처: dsba 의 cs224n 세미나 자료

모든 경우를 따져보는 Graph-based Parsing 에 비해서 Transition-based Parsing 은 정확도가 떨어진다는 한계를 갖고 있었지만,

Chen and Manning(2014)에서 dense feature 를 사용한 신경망 기반 transition-based parser 를 제안하면서 속도와 성능을 모두 향상시켰다.

이 강의에서 주로 이 transition-based parser 를 자세히 다룬다.



initial state 를 state  $c$  로 설정하고, 이 state  $c$  를 임베딩한 후,  $f(c)$ 함수에 입력함으로써 dependency 를 결정한다.

$c$  는 stack, buffer, set of arcs 로 이루어져있다. Shift 는 buffer 에서 stack 로 토큰이 이동하는 과정을 의미한다.

한편, Chen and Manning(2014)에서는 State  $c$  가 임베딩되는 과정과,  $f(c)$ 의 input 으로 들어가는 과정은 conventional feature representation(Sparse Feature, ML Model 사용)과 neural dependency parser(Dense Feature, NN Model 사용) 부분에서 설명하고 있다.

\*Projectivity & Non-Projectivity

- projectivity: dependency arc 가 서로 겹치지 않는 성질
- non-projectivity: dependency arc 가 겹치는 성질
- 대부분의 dependency parser 가 input 의 projectivity 를 가정하고 있다.
- non-projectivity 를 처리하기 위한 별도의 연구 분야가 있으며, 전처리를 통해 non-projectivity 를 projectivity 문장으로 변환하거나, non-projectivity input 에 대한 결정을 보류하는 decision 을 softmax layer 에 추가하기도 한다.

참고한 자료

[Constituency parsing | NLP-progress \(nlpprogress.com\)](http://nlp-progress.com)