의 존 구 문 분 석
(Dependency Parsing)

남궁영
한국해양대학교 컴퓨터공학과
young_ng@kmou.ac.kr

2019. 04. 05.

국립 한국해양대학교
KOREA MARITIME AND OCEAN UNIVERSITY

Content

- Recap
  - Transition-based dependency parsing
- Korean Dependency Parsing
  - arc-standard (Nivre, 2004, 2008)
  - arc-eager (Nivre, 2003, 2008) ~ ☆한국어
- Treebank

:: R E C A P ::

Parsing

❖ Syntactic Parsing

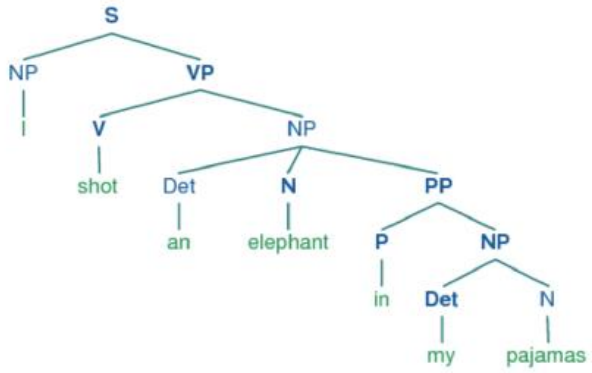
- the process of analyzing the construction of a sentence
  by recognizing a sentence and assigning a syntactic structure to it

  - **Input**: String or Grammar
  - **Output**: Parse tree(s)

Grammar & Parsing

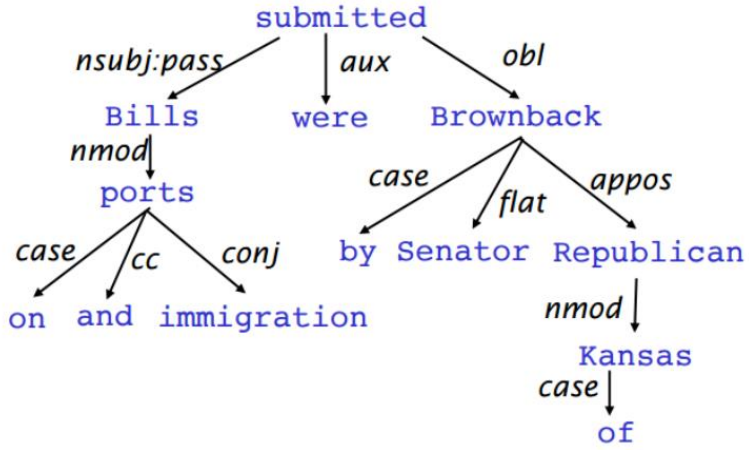
Phrase Structure Grammar

- Constituents
- Consist of
  - Rules
  - Terminals & Non-terminals



Dependency Grammar

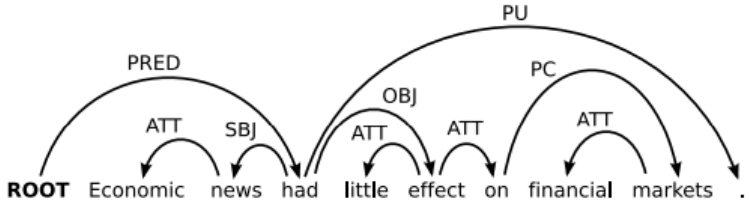
- Head-dependent relations
  - nodes: words
  - links: dependencies between words
- Suitable for free word-order languages

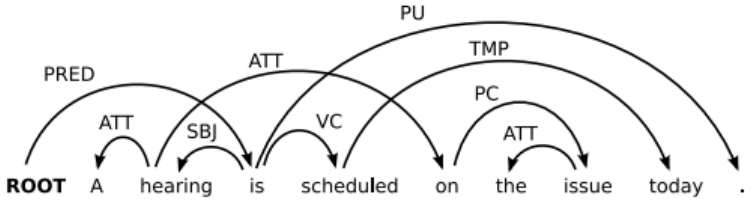


Dependency Tree

❖ A parse is a directed rooted tree:

- Connected, Acyclic, Single-head
- Arcs indicate certain **grammatical relations** between words
- non-projective / **projective**
  - projective
    - the arc (i,l,j) implies that i →∗ k for every k such that min(i, j) < k < max(i, j)
  - many DP algorithms can only handle projective trees
  - Non-projective trees do occur in natural language
  - Transition-based parsing
    - using a swap-transition
    - using more than one stack
  - Graph-based parsing
    - Minimum spanning tree algorithms





| Sara Stymne. 2014. 'Transition-based dependency parsing', Syntactic analysis. Uppsala univ.
https://cl.lingfil.uu.se/~sara/kurser/5LN455-2014/lectures/5LN455-F8.pdf

Arc-standard transition-based parser

❖ Basic transition-based dependency parser

Shift

Reduce
: create dependencies

Start: $\sigma = [\text{ROOT}], \beta = w_1, \ldots, w_n, A = \varnothing$

1. Shift $\quad \sigma, w_i \mid \beta, A \rightarrow \sigma \mid w_i, \beta, A$
2. Left-Arc$_r$ $\quad \sigma \mid w_i \mid w_j, \beta, A \rightarrow \sigma \mid w_j, \beta, A \cup \{r(w_j, w_i)\}$
3. Right-Arc$_r$ $\quad \sigma \mid w_i \mid w_j, \beta, A \rightarrow \sigma \mid w_i, \beta, A \cup \{r(w_i, w_j)\}$

Finish: $\sigma = [w], \beta = \varnothing$

| https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/lecture_notes/cs224n-2017-notes4.pdf

Neural Dependency Parsing (FFN)

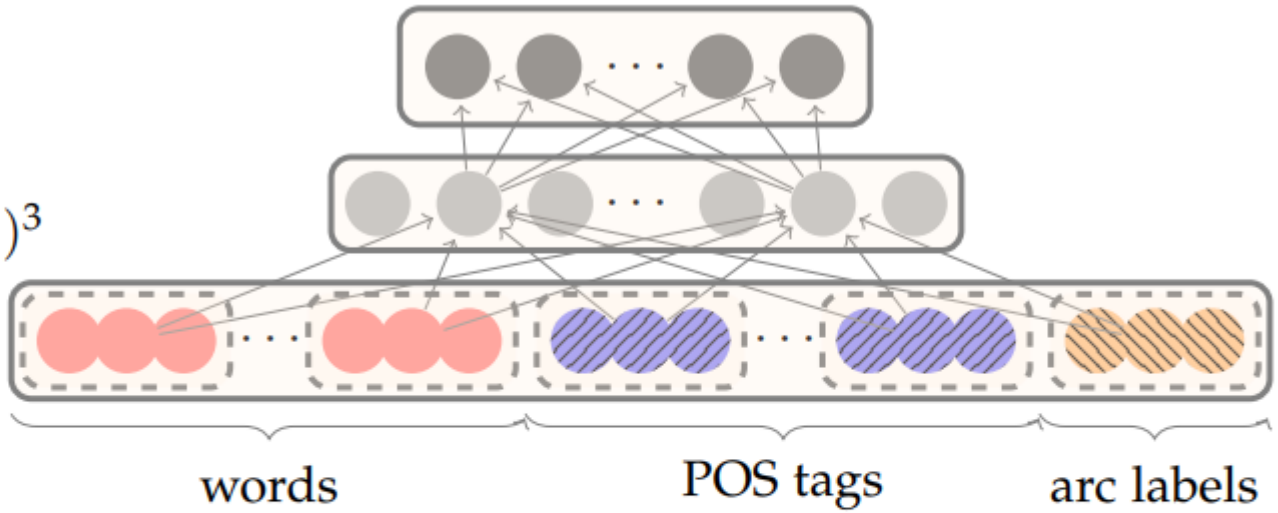
Softmax layer:

$$p = \texttt{softmax}(W_2 h)$$

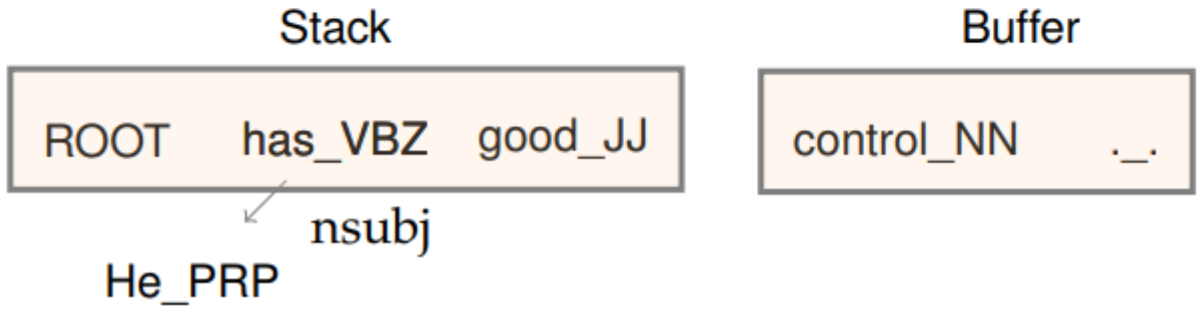
Hidden layer:

$$h = (W_1^w x^w + W_1^t x^t + W_1^l x^l + b_1)^3$$

Input layer: $[x^w, x^t, x^l]$



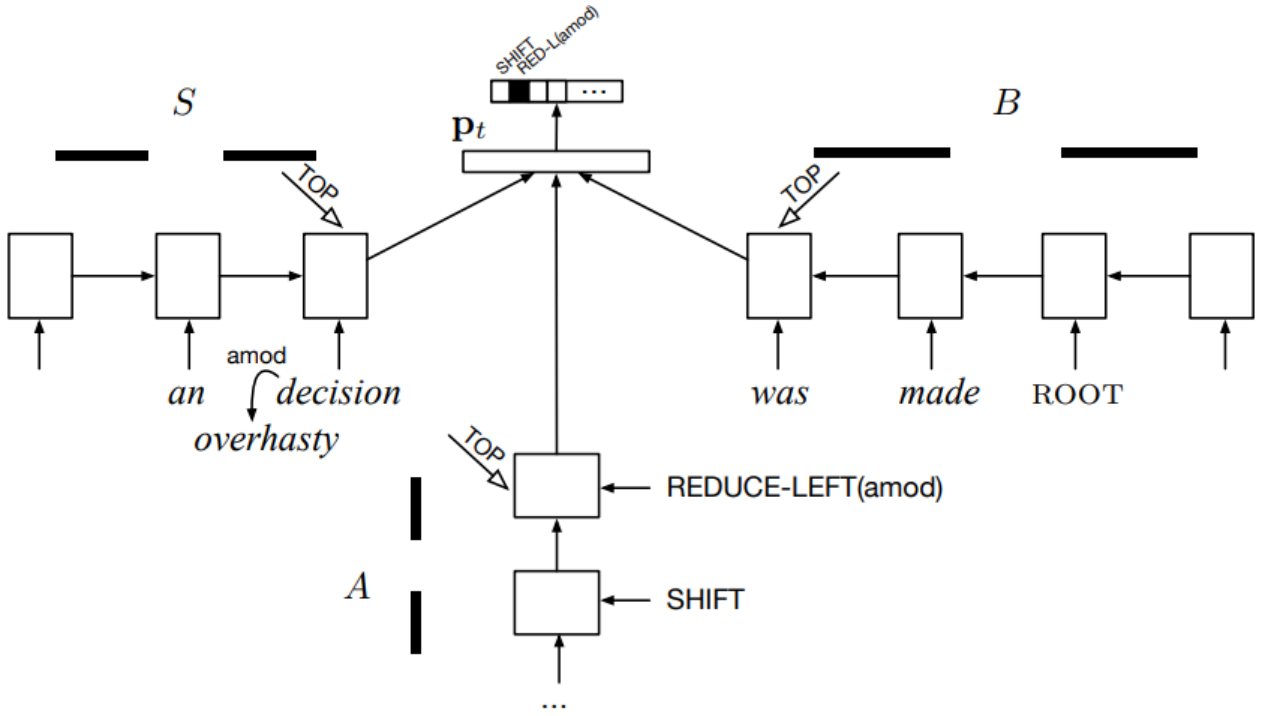
Configuration



| https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/lecture_notes/cs224n-2017-notes4.pdf

Neural Dependency Parsing (stack LSTM)

❖ Transition-Based Dependency Parsing with Stack Long Short-Term Memory



- using three Stack-LSTM for S, B, A
- computes a composite representation of the stack states

  → to predict an action to take

| C. Dyer et al. "Transition-Based Dependency Parsing with Stack Long Short-Term Memory", ACL 2015

:: Korean Dependency Parsing ::

Arc-standard & Arc-eager

arc-standard

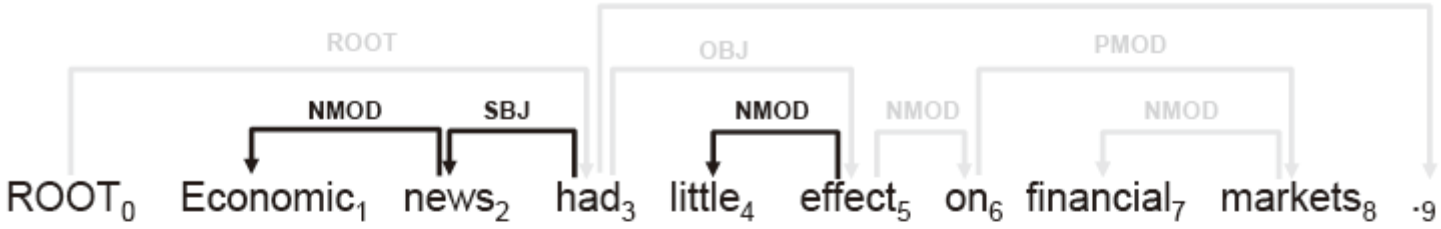
Shift	$(\sigma, i\|\beta, A) \Rightarrow (\sigma\|i, \beta, A)$
LArc	$(\sigma\|i\|j, \beta, A) \Rightarrow (\sigma\|j, \beta, A \cup \{(j \rightarrow i)\})$
RArc	$(\sigma\|i\|j, \beta, A) \Rightarrow (\sigma\|i, \beta, A \cup \{(i \rightarrow j)\})$

arc-eager

Shift	$(\sigma, i\|\beta, A) \Rightarrow (\sigma\|i, \beta, A)$
LArc	$(\sigma\|i, j\|\beta, A) \Rightarrow (\sigma, j\|\beta, A \cup \{(j \rightarrow i)\})$
RArc	$(\sigma\|i, j\|\beta, A) \Rightarrow (\sigma\|i\|j, \beta, A \cup \{(i \rightarrow j)\})$
Reduce	$(\sigma\|i, \beta, A) \Rightarrow (\sigma, \beta, A)$

- ❖ can create arcs earlier
- ❖ reduce when a node already find its head

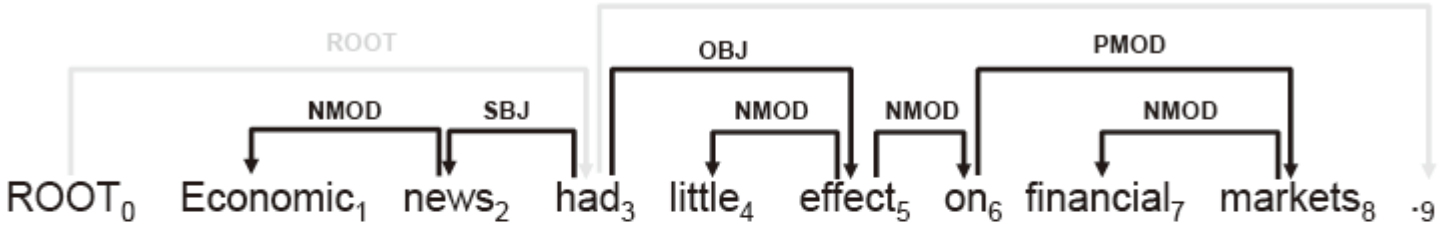
Arc-standard (1)



ACTION	Stack	Buffer	Relation
	[0]	[1..9]	
shift	[0 1]	[2..9]	
shift	[0 1 2]	[3..9]	
Left-ARC	[0 2]	[3..9]	(NMOD, news→Econo..)
shift	[0 2 3]	[4..9]	
Left-ARC	[0 3]	[4..9]	(SBJ, had→news)
shift	[0 3 4]	[5..9]	
shift	[0 3 4 5]	[6..9]	
Left-ARC	[0 3 5]	[6..9]	(NMOD, effect→little)
shift	[0 3 5 6]	[7 8 9]	

| 이공주, 2018 자연언어처리 튜토리얼, pp.25-59

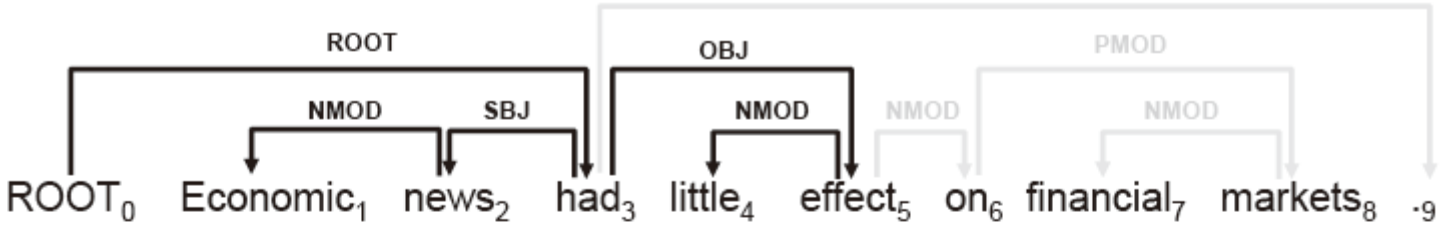
Arc-standard (2)



ACTION	Stack	Buffer	Relation
shift	[0 3 5 6]	[7 8 9]	
shift	[0 3 5 6 7]	[8 9]	
shift	[0 3 5 6 7 8]	[9]	
Left-ARC	[0 3 5 6 8]	[9]	(NMOD, market→finan..)
Right-ARC	[0 3 5 6]	[9]	(PMOD, on→market)
Right-ARC	[0 3 5]	[9]	(NMOD, effect→on)
Right-ARC	[0 3]	[9]	(OBJ, had→effect)
shift	[0 3 9]	[]	

| 이공주, 2018 자연언어처리 튜토리얼, pp.25-59

Arc-eager



ACTION	Stack	Buffer	Relation
	[0]	[1..9]	
shift	[0 1]	[2..9]	
Left-ARC	[0]	[3..9]	(NMOD, news→Econo…)
shift	[0 2]	[3..9]	
Left-ARC	[0]	[3..9]	(SBJ, had→news)
Right-ARC	[0 3]	[4..9]	(ROOT, ROOT→had)
shift	[0 3 4]	[5..9]	
Left-ARC	[0 3]	[5..9]	(NMOD, effect→little)
Right-ARC	[0 3 5]	[6..9]	(OBJ, had→effect)

| 이공주, 2018 자연언어처리 튜토리얼, pp.25-59

Korean Dependency Parsing

❖ Arc-eager + Backward transition-based dependency parsing

❖ 한국어의 지배소(head)는 대부분 후위에 위치

→ Backward 방식이 유리

- 문장의 용언(지배소)들이 먼저 stack에 쌓임
- 특정 노드가 어느 용언에 지배되는지 결정하는데 도움

<입력문장> CJ그룹이 대한통운 인수계약을 체결했다

1. **Stack=[root], Buffer=[체결했다$_4$ 인수계약을$_3$ 대한통운$_2$ CJ그룹이$_1$], Arc={}**
   ⬇ Right-arc(VP)
2. [root 체결했다$_4$], [인수계약을$_3$ …], A={(root→체결했다$_4$)}
   ⬇ Right-arc(NP_OBJ)
3. [root 체결했다$_4$ 인수계약을$_3$], [대한통운$_2$ …], {(체결했다$_4$→인수계약을$_3$), …}
   ⬇ Right-arc(NP_MOD)
4. [root 체결했다$_4$ 인수계약을$_3$ 대한통운$_2$], [CJ그룹이$_1$], {(인수계약을$_3$→대한통운$_2$), …}
   ⬇ Reduce
5. [root 체결했다$_4$ 인수계약을$_3$], [CJ그룹이$_1$], {(인수계약을$_3$→대한통운$_2$), …}
   ⬇ Reduce
6. [root 체결했다$_4$], [CJ그룹이$_1$], {(인수계약을$_3$→대한통운$_2$), …}
   ⬇ Right-arc(NP_SUB)
7. [root 체결했다$_4$ CJ그룹이$_1$], [], {(체결했다$_4$→CJ그룹이), …}

| 이창기, 김준석, 김정희. 2014. 딥러닝을 이용한 한국어 의존 구문 분석, HCLT 논문집, pp.87-91

:: Treebanks ::

Treebanks

- Treebanks are corpora in which each sentence has been annotated with a syntactic analysis.
- The annotation process requires detailed guidelines and measures for quality control.
- Producing a high-quality treebank is both time-consuming and expensive.

| Sara Stymne. 2014. 'Treebank Grammars and Parser Evaluation', Syntactic analysis. Uppsala univ. https://cl.lingfil.uu.se/~sara/kurser/5LN455-2014/lectures/5LN455-F4.pdf

Dependency Treebank

- **Prague Dep. treebank 3.0 (2013)**
  - 1,506,484 words, 87,913 sentences
- **Danish Dep. treebank 1.0 (2004)**
  - 100,200 words, 5,540 sentences
- **Quranic Arabic Dep. Treebank (QADT)**
- **Italian Stanford Dep. Treebank (ISDT)**
  - 298,344 words, 14,167 sentences
- Converted phrase-structured treebanks (e.g. Penn)
  - J. Chun, N. Han, J. D. Hwang, J. D. Choi. 2018. **Building Universal Dependency Treebanks in Korean**. Proceedings of the 11th LREC, pp. 2194-2202
  - Phrase structure trees in the Penn Korean Treebank & KAIST Treebank are automatically converted into dependency trees using head-finding rules and linguistic heuristics.

| https://en.wikipedia.org/wiki/Treebank#cite_note-9
| http://universaldependencies.org/

Korean Treebank (1)

- 21세기 세종 계획 (98’~07’)
- 구문 분석 말뭉치 (21세기 세종 계획 국어기초자료구축)
  - 세종 ‘의미 분석 말뭉치 ' 에 통사 구조/기능 표지를 부가
- Size: 약 45만 (433, 839어절, 43,828문장)

| 황용주, 최정도. 2016. 21세기 세종 말뭉치 제대로 살펴보기,「새국어생활」, 국립국어원.
https://www.korean.go.kr/nkview/nklife/2016_2/26_0204.pdf

Korean Treebank (2)

❖구문분석 말뭉치 구축 지침

· 구문 표지

	범주	사례
S	문장	
Q	인용절	인용부호(“”) 안에 들어 있는 두 개 이상의 문장
NP	체언구	체언(명사, 대명사, 수사)
VP	용언구	용언(동사, 형용사, 보조용언)
VNP	긍정 지정사구	긍정 지정사 ‘이다’와 결합한 구
AP	부사구	부사
DP	관형사구	관형사
IP	감탄사구	감탄사

· 기능 표지

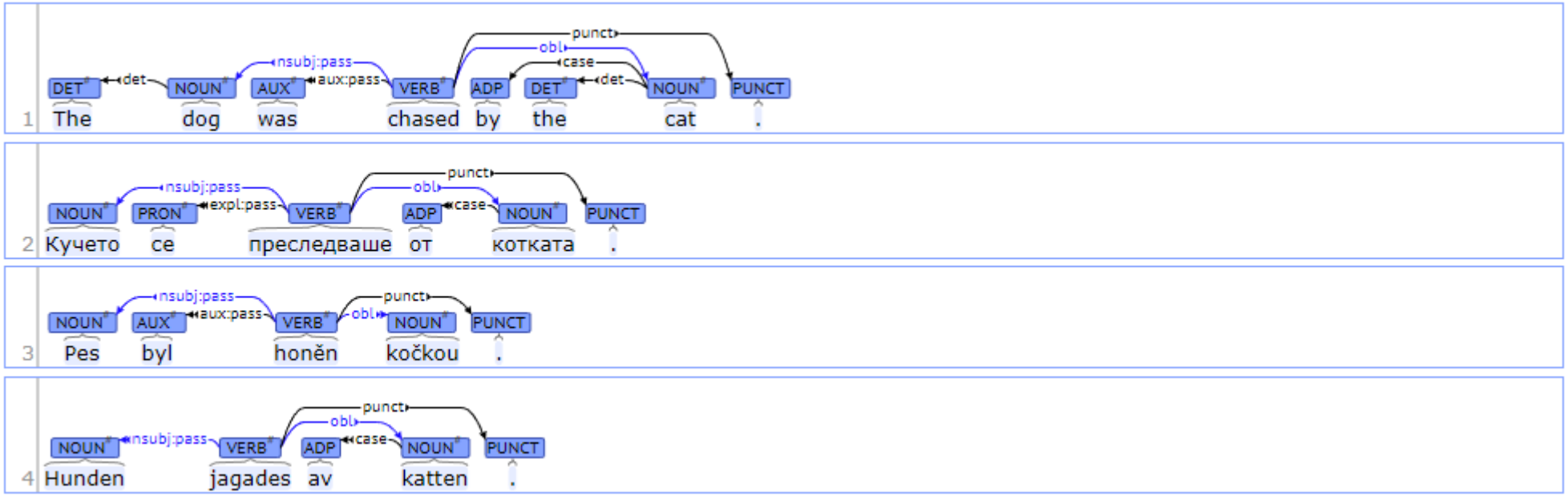
	범주	사례
SBJ	주어	주격 체언구, 명사 전성 용언구, 명사절 (NP_SBJ, VP_SBJ, S_SBJ, VNP_SBJ)
OBJ	목적어	목적격 체언구, 명사 전성 용언구, 명사절 (NP_OBJ, VP_OBJ, S_OBJ, VNP_OBJ)
CMP	보어	보격 체언구, 명사 전성 용언구, 인용절 (NP_CMP, VP_CMP, S_CMP, VNP_CMP)
MOD	체언 수식어	관형격 체언구, 관형형 용언구, 관형절 (NP_MOD, VP_MOD, S_MOD, VNP_MOD)
AJT	용언 수식어	부사격 체언구, 문말어미+부사격조사 (NP_AJT VP_AJT, S_AJT, VNP_AJT)
CNJ	접속어	접속격 체언(NP_CNJ, VNP_CNJ)
INT	독립어	체언 (NP_INT)

예시)

```
; 우리는 오관을 통하여 우주의 삼라만상에 관한 정보를 내 속으로 받아들인다.
(S       (NP_SBJ 우리/NP + 는/JX)
         (VP      (VP      (NP_OBJ 오관/NNG + 를/JKO)
                           (VP 통하/VV + 아/EC))
                  (VP      (NP_OBJ          (VP_MOD          (NP_AJT          (NP_MOD 우주/NNG + 의/JKG)
                                                                              (NP_AJT 삼라만상/NNG + 에/JKB))
                                                            (VP_MOD 관하/VV + ㄴ/ETM))
                                            (NP_OBJ 정보/NNG + 를/JKO))
                           (VP      (NP_AJT          (NP_MOD 나/NP + 의/JKG)
                                                     (NP_AJT 속/NNG + 으로/JKB))
                                    (VP 받아들이/VV + ㄴ다/EF + ./SF))))))
```

Universal dependency treebank

- Universal Dependencies (UD)
  - developing cross-linguistically consistent treebank annotation for many languages, with the goal of facilitating multilingual parser development, cross-lingual learning, and parsing research from a language typology perspective.



감 사 합 니 다.

남궁영
한국해양대학교 컴퓨터공학과
young_ng@kmou.ac.kr

2019. 04. 05.