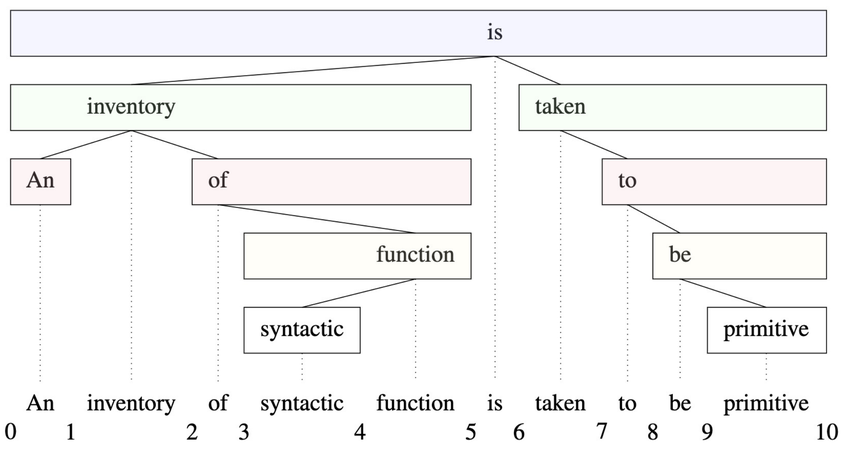
Code Paper

Headed-Span-Based Projective Dependency Parsing

Songlin Yang, Kewei Tu
School of Information Science and Technology, ShanghaiTech University

Introduction



Motivation: First-order graph-based parsers cannot capture sufficient subtree information. Most of transition-based parsers cannot perform global optimization.

Observation: In a projective parse tree, the whole subtree rooted at a headword forms a contiguous sequence (i.e., span) in the surface order. We call such a span-headword pair as **headed span**. A projective parse tree can be viewed as a collection of headed spans, similar to the span-based view of constituency parsing.

Method: We decompose the score of a dependency tree into scores of headed spans and design a novel cubic dynamic programming algorithm to enable global optimization. Modeling headed spans allows us to use rich span embedding to encode more subtree information.

Parsing as deduction

Axioms:

β-INIT: $\dfrac{}{s^{\text{span}}_{i,i+1,i+1}}$ (triangle: $i \quad i+1 \quad i+1$)

α-INIT: $\dfrac{}{0}$ (rectangle: $i \quad i$)

Deduction Rules:

S-CONC $\dfrac{s_1 \quad s_2}{s_1 + s_2}$ (rectangle: $i \quad j$)

C-CONC $\dfrac{s_1 \quad s_2}{s_1 + s_2 + s^{\text{span}}_{i,j,k}}$ (triangle: $i \quad k \quad j$)

HEADLESS $\dfrac{s_1}{s_1}$ (triangle: $i \quad k \quad j$; rectangle: $i \quad j$)

The corresponding recursive formulas:

$$\begin{aligned}
\beta_{i,i+1,i+1} &= s^{\text{span}}_{i,i+1,i+1} \\
\alpha_{i,i} &= 0 \\
\beta_{i,j,k} &= \alpha_{i,k-1} + \alpha_{k,j} + s^{\text{span}}_{i,j,k} \\
\alpha_{i,j} &= \max\Big(\max_{i<k<j}(\alpha_{i,k} + \alpha_{k,j}), \max_{i<h\le j}(\beta_{i,j,h})\Big)
\end{aligned}$$

- $\alpha_{i,j}$: the accumulated score of span (i, j) serving as a left or right child span
- $\beta_{i,j,k}$: the accumulated score of the headed span (i, j, k).
- S-CONC: the rule used to concatenate two consecutive child spans into a single child span.
- C-CONC: the rule used to concatenate left and right child span $(i, k-1)$ and (k, j) along with the root word-span $(k-1, k)$ to form a headed span (i, j, k).
- HEADLESS: the rule used to obtain a headless child span from a headed span.

Our algorithm ≈ hook trick + head-splitting trick

The hook trick: reduce subtree into headless spans because the linking of heads and the concatenation of subtrees can be separated.

ES-R-CONC: $\dfrac{s_1 \quad s_2}{s_1 + s_2}$ (s_1: $i\ h\ k$; s_2: $h\ k+1\ j$; result: $i\ h\ j$)

ES-R-LINK: $\dfrac{}{s_1 + s^{\text{arc}}_{h,c}}$ (s_1: $i\ c\ j$; result: $h\ i\ j$)

ES-L-CONC: $\dfrac{s_1 \quad s_2}{s_1 + s_2}$ (s_1: $i\ k-1\ h$; s_2: $k\ h\ j$; result: $i\ h\ j$)

ES-L-LINK: $\dfrac{}{s_1 + s^{\text{arc}}_{h,c}}$ (s_1: $i\ c\ j$; result: $s_1 + s^{\text{arc}}_{h,c}$: $i\ j\ h$)

The head-splitting trick: split each subtree into a left and a right fragment. We can transform the above rules to R-CONC, L-CONC, and CONC using the head-splitting trick.

R-CONC: $\dfrac{s_1 \quad s_2}{s_1 + s_2}$ (s_1: $h\ i\ k$; s_2: $h\ k\ j$; result: $h\ i\ j$)

L-CONC: $\dfrac{s_1 \quad s_2}{s_1 + s_2}$ (s_1: $i\ k\ h$; s_2: $k\ j\ h$; result: $i\ j\ h$)

CONC: $\dfrac{s_1 \quad s_2}{s_1 + s_2}$ (s_1: $i\ h\ h$; s_2: $h\ h+1\ j$; result: $i\ h\ j$)

Ours: we do not consider arc scores, thus eliminate the need to maintain the bookkeeping of h. We can merge L-CONC and R-CONC to S-CONC; modify CONC to C-CONC, resulting in our proposed algorithm.

Experiments

	bg	ca	cs	de	en	es	fr	it	nl	no	ro	ru	Avg
TreeCRF2O	90.77	91.29	91.54	80.46	87.32	90.86	87.96	91.91	88.62	91.02	86.90	93.33	89.33
MFVI2O	90.53	92.83	92.12	81.73	89.72	92.07	88.53	92.78	90.19	91.88	85.88	92.67	90.07
						+BERT$_{\text{multilingual}}$							
MFVI2O	**91.30**	93.60	92.09	82.00	90.75	92.62	89.32	93.66	91.21	91.74	86.40	92.61	90.61
Biaffine+MM†	90.30	**94.49**	**92.65**	**85.98**	91.13	93.78	**91.77**	94.72	91.04	94.21	87.24	**94.53**	91.82
Ours	91.10	94.46	92.57	85.87	**91.32**	**93.84**	91.69	**94.78**	**91.65**	**94.28**	**87.48**	94.45	**91.96**

Table 1. Labeled Attachment Score (LAS) on twelve languages in UD 2.2.

	PTB		CTB	
	UAS	LAS	UAS	LAS
MFVI2O	95.98	94.34	90.81	89.57
TreeCRF2O	96.14	94.49	-	-
HierPtr	96.18	94.59	90.76	89.67
	+BERT$_{\text{base}}$		+BERT$_{\text{base}}$	
RNGTr	96.66	95.01	92.98	91.18
	+BERT$_{\text{large}}$		+BERT$_{\text{base}}$	
MFVI2O	96.91	95.34	92.55	91.69
HierPtr	97.01	95.48	92.65	91.47
Biaffine+MM†	97.22	95.71	93.18	92.10
Ours	**97.24**	**95.73**	**93.33**	**92.30**
	For reference			
	+XLNet$_{\text{large}}$		+BERT$_{\text{base}}$	
HPSG♭	97.20	95.72	-	-
HPSG+LAL♭	97.42	96.26	94.56	89.28

Table 2. Results for different model on PTB and CTB.