Straight to the Tree: Constituency Parsing with Neural Syntactic Distance

Yikang Shen*, Zhouhan Lin*,
Athul Paul Jacob, Alessandro Sordoni, Aaron Courville, Yoshua Bengio

University of Montreal, Microsoft Research, University of Waterloo

Overview

- Motivation
- Syntactic Distance based Parsing Framework
- Model
- Experimental Results

Overview

- **Motivation**
- Syntactic Distance based Parsing Framework
- Model
- Experimental Results

ICLR 2018:
Neural Language Modeling by Jointly Learning Syntax and Lexicon

Syntactic
Dis

Structured

Supervised Constituency Parsing with Syntactic Distance?

(61 ppl)

(68 UF1)

[Shen et al. 2018]

Chart Neural Parsers

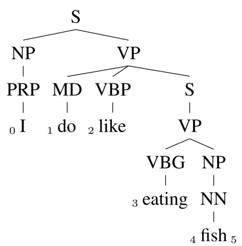
CYK table

S						
	VP					
S						
	VP			PP		
S		NP			NP	
NP	V, VP	Det.	N	P	Det	N
she	eats	a	fish	with	a	fork

1. High computational cost:
   Complexity of CYK is O(n^3).
2. Complicated loss function:

$$\max\left(0, \max_{T}\left[s(T) + \Delta(T, T^*)\right] - s(T^*)\right)$$

Transition based Neural Parsers



(a) gold parse tree

steps	structural action	label action	stack after	bracket
1–2	sh(I/PRP)	label-NP	$_0\triangle_1$	$_0\mathrm{NP}_1$
3–4	sh(do/MD)	nolabel	$_0\triangle_1\triangle_2$	
5–6	sh(like/VBP)	nolabel	$_0\triangle_1\triangle_2\triangle_3$	
7–8	comb	nolabel	$_0\triangle_1\triangle_3$	
9–10	sh(eating/VBG)	nolabel	$_0\triangle_1\triangle_3\triangle_4$	
11–12	sh(fish/NN)	label-NP	$_0\triangle_1\triangle_3\triangle_4\triangle_5$	$_4\mathrm{NP}_5$
13–14	comb	label-S-VP	$_0\triangle_1\triangle_3\triangle_5$	$_3\mathrm{S}_5$, $_3\mathrm{VP}_5$
15–16	comb	label-VP	$_0\triangle_1\triangle_5$	$_1\mathrm{VP}_5$
17–18	comb	label-S	$_0\triangle_5$	$_0\mathrm{S}_5$

(b) static oracle actions

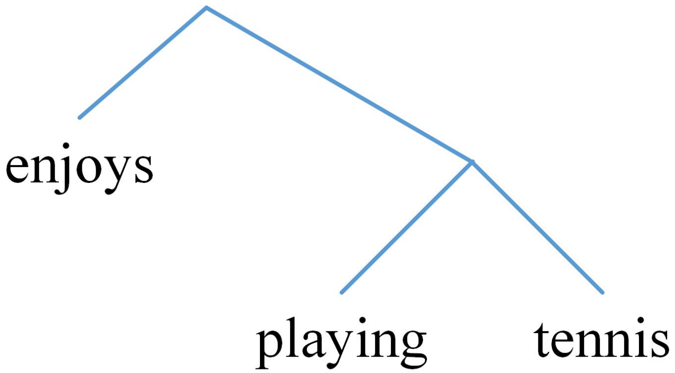
1. Greedy decoding:
   Incompleted tree (the shift and reduce steps may not match).
2. Exposure bias
   The model is never exposed to its own mistakes during training

[Stern et al., 2017; Cross and Huang, 2016]

Overview

- Motivation
- **Syntactic Distance based Parsing Framework**
- Model
- Experimental Results

Intuitions



Only the order of split (or combination) matters for reconstructing the tree.

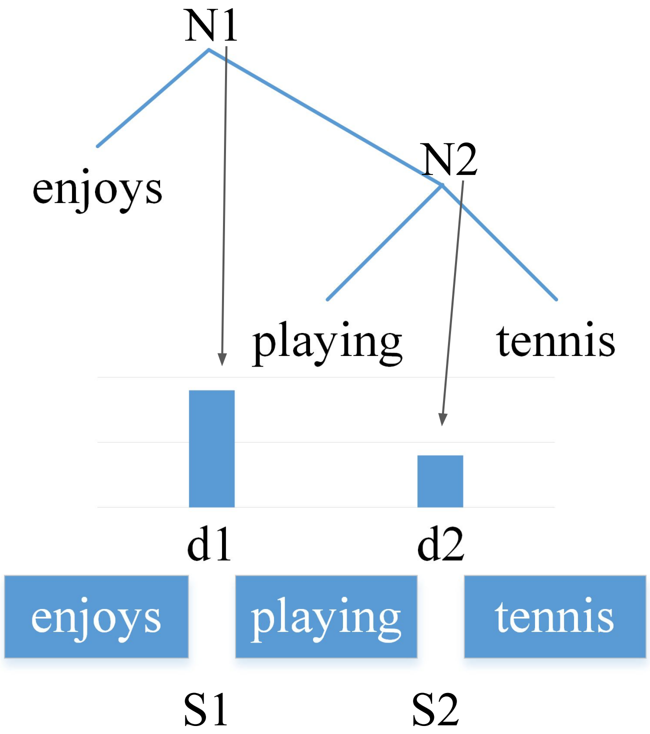
Can we model the order directly?

Syntactic distance

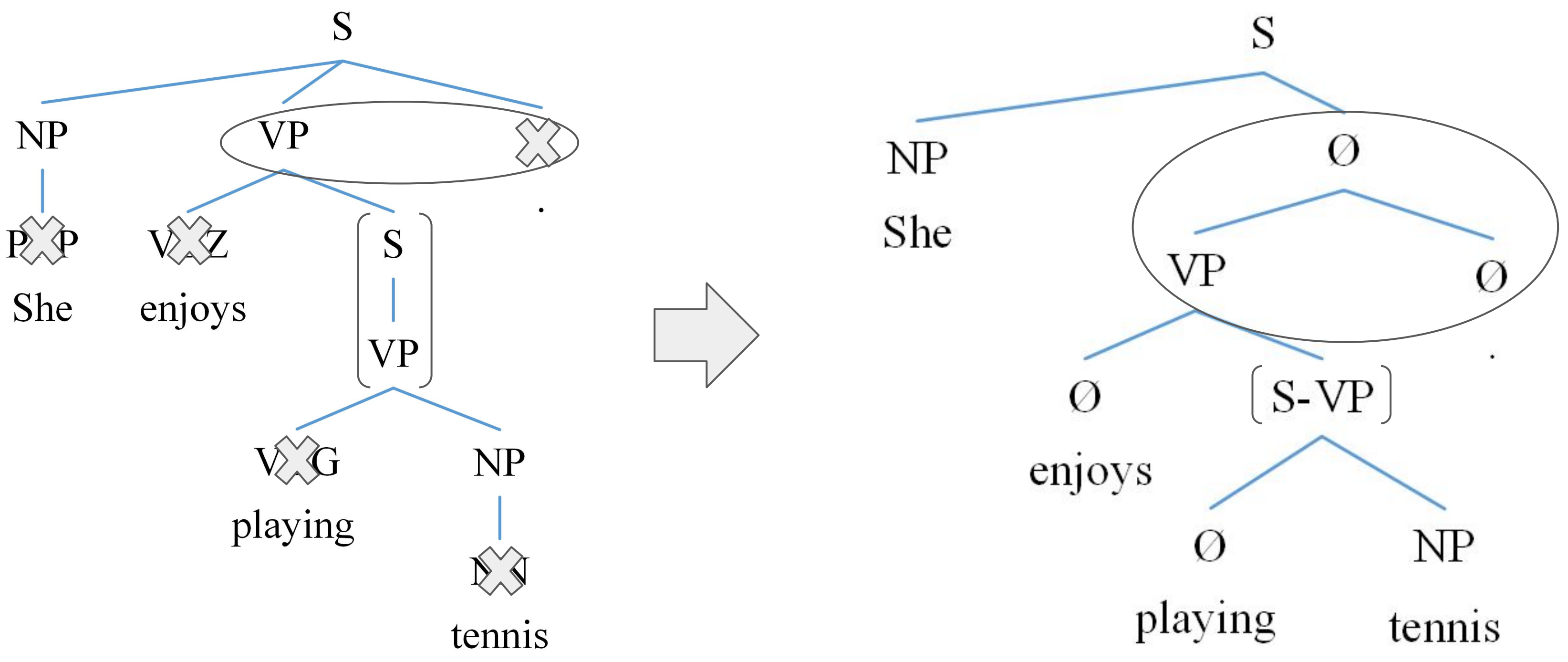
Definition 2.1. Let $\mathbf{T}$ be a parse tree that contains a set of leaves $(w_0, ..., w_n)$. The height of the lowest common ancestor for two leaves (w_i, w_j) is noted as $\tilde{d}_j^i$. The syntactic distances of $\mathbf{T}$ can be any vector of scalars $\mathbf{d} = (d_1, ..., d_n)$ that satisfy:

$$\mathrm{sign}(d_i - d_j) = \mathrm{sign}(\tilde{d}_i^{i-1} - \tilde{d}_j^{j-1}) \qquad (1)$$

For each **split point**, their **syntactic distance** should share the same order as the height of **related node**



Convert to binary tree



[Stern et al., 2017]

Tree to Distance

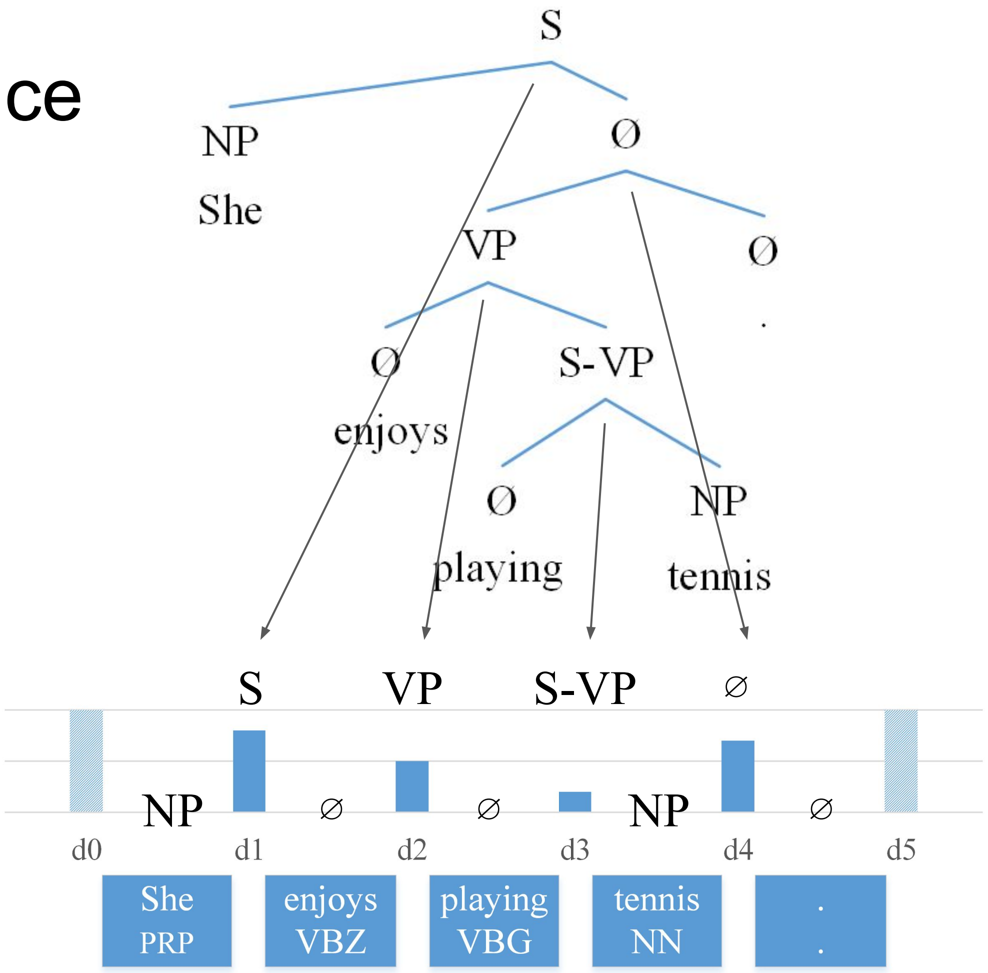
The height for each non-terminal node is the maximum height of its children plus 1

Algorithm 1 Binary Parse Tree to Distance

($\cup$ represents the concatenation operator of lists)

function DISTANCE(node)
    if node **is** leaf **then**
        $\mathbf{d} \leftarrow []$
        $\mathbf{c} \leftarrow []$
        $\mathbf{t} \leftarrow [\text{node.tag}]$
        $h \leftarrow 0$
    else
        $\text{child}_l, \text{child}_r \leftarrow$ children of node
        $\mathbf{d}_l$, $\mathbf{c}_l$, $\mathbf{t}_l$, $h_l \leftarrow \text{Distance}(\text{child}_l)$
        $\mathbf{d}_r$, $\mathbf{c}_r$, $\mathbf{t}_r$, $h_r \leftarrow \text{Distance}(\text{child}_r)$
        $h \leftarrow \max(h_l, h_r) + 1$
        $\mathbf{d} \leftarrow \mathbf{d}_l \cup [h] \cup \mathbf{d}_r$
        $\mathbf{c} \leftarrow \mathbf{c}_l \cup [\text{node.label}] \cup \mathbf{c}_r$
        $\mathbf{t} \leftarrow \mathbf{t}_l \cup \mathbf{t}_r$
    end if
    return $\mathbf{d}$, $\mathbf{c}$, $\mathbf{t}$, h
end function

Tree to Distance



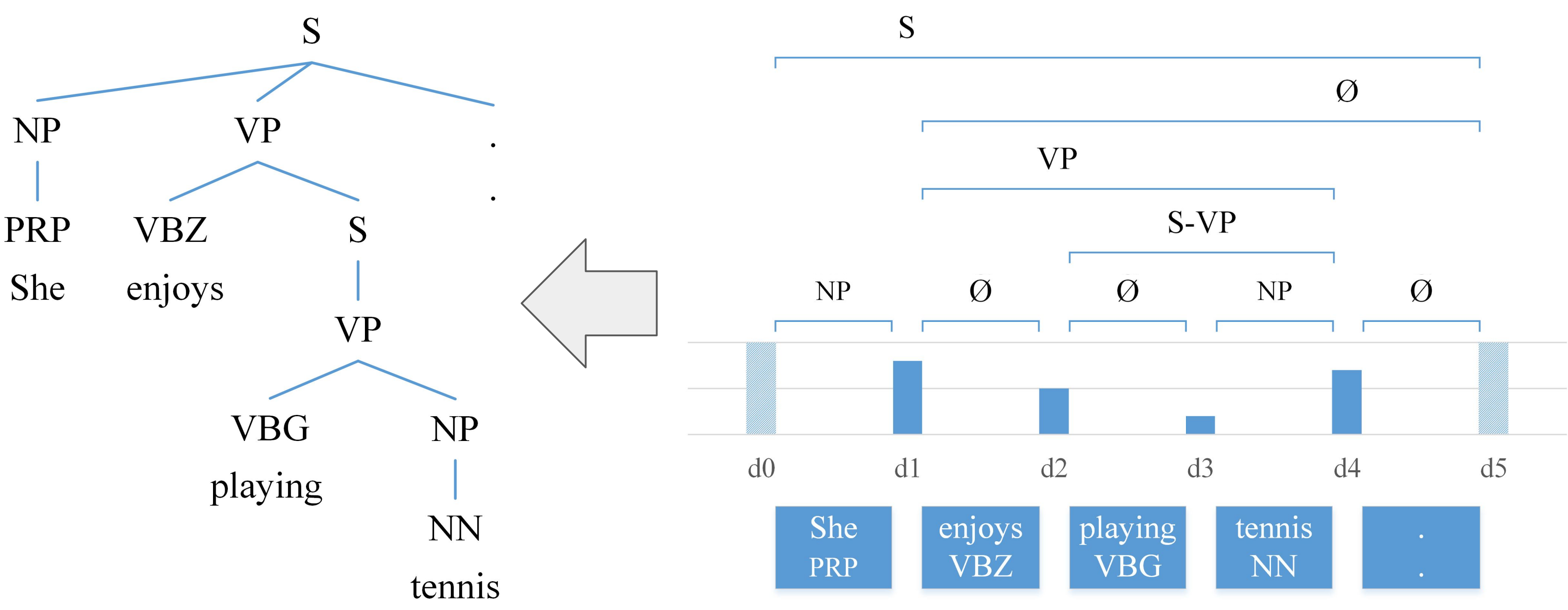
Distance to Tree

Split point for each bracket is the one with maximum distance.

Algorithm 2 Distance to Binary Parse Tree

function $\textsc{Tree}(\mathbf{d},\mathbf{c},\mathbf{t})$
    if $\mathbf{d} = []$ **then**
        $\text{node} \leftarrow \text{Leaf}(\mathbf{t})$
    else
        $i \leftarrow \arg\max_i(\mathbf{d})$
        $\text{child}_l \leftarrow \text{Tree}(\mathbf{d}_{<i}, \mathbf{c}_{<i}, \mathbf{t}_{<i})$
        $\text{child}_r \leftarrow \text{Tree}(\mathbf{d}_{>i}, \mathbf{c}_{>i}, \mathbf{t}_{\geq i})$
        $\text{node} \leftarrow \text{Node}(\text{child}_l, \text{child}_r, \mathbf{c}_i)$
    end if
    return node
end function

Distance to Tree



Overview

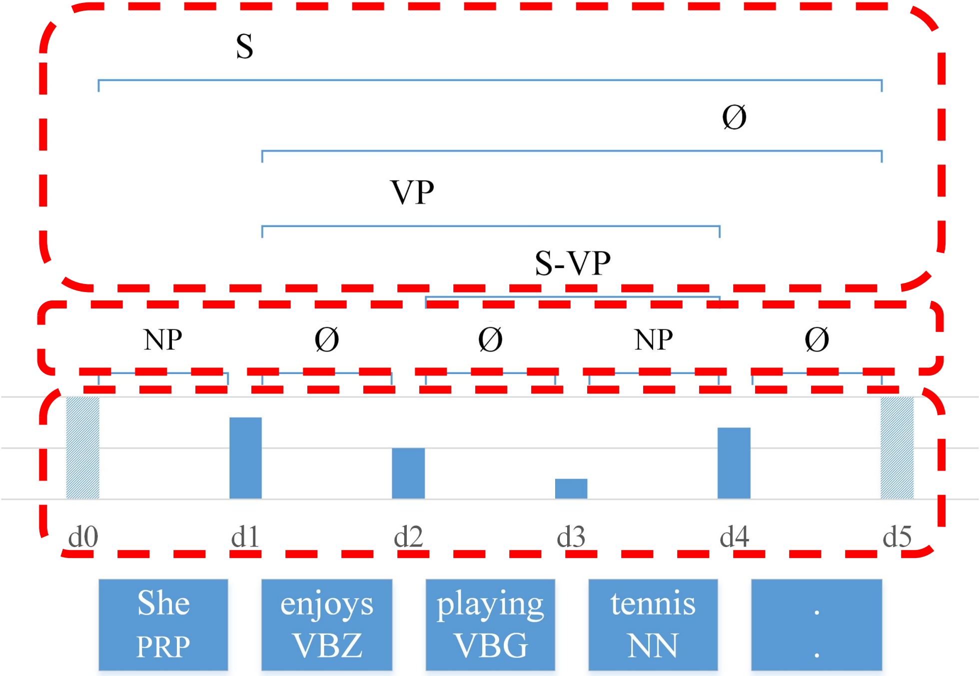
- Motivation
- Syntactic Distance based Parsing Framework
- **Model**
- Experimental Results

Framework for inferring the distances and labels

Labels for non-leaf nodes

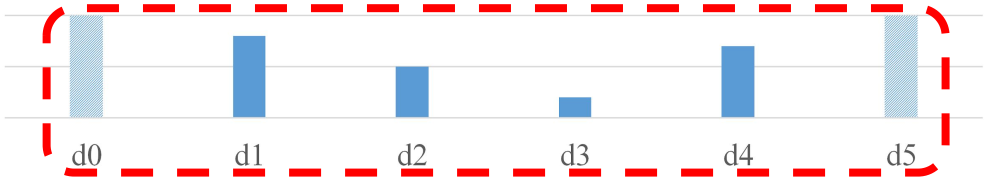
Labels for leaf nodes

Distances

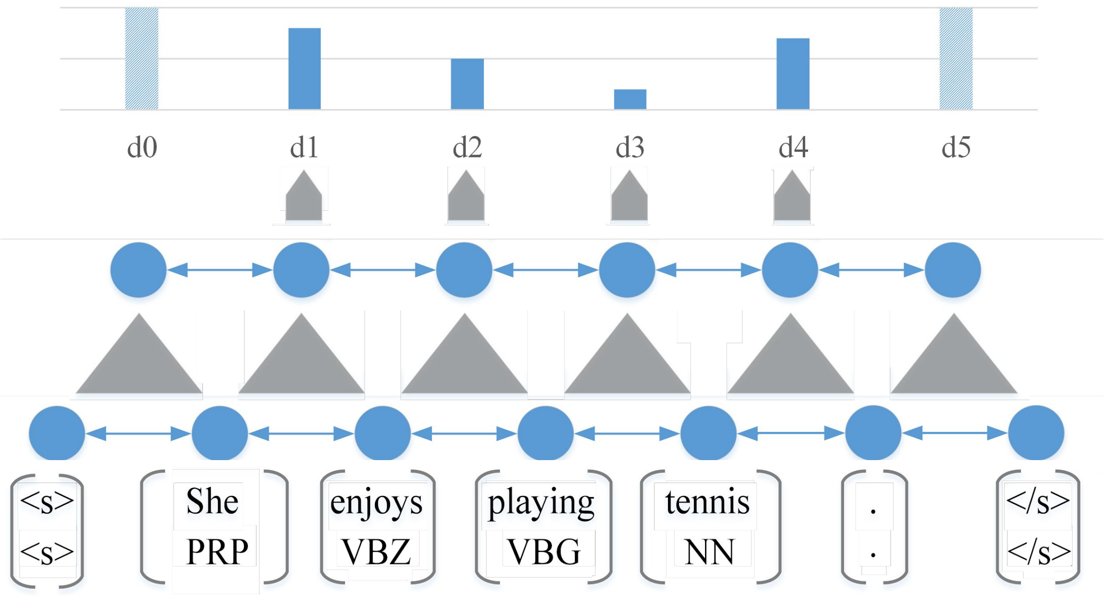


Inferring the distances

Distances



Inferring the distances



Pairwise learning-to-rank loss for distances

$$L_{\text{dist}}^{\text{rank}} = \sum_{i,j>i} [1 - \text{sign}(d_i - d_j)(\hat{d}_i - \hat{d}_j)]^+$$

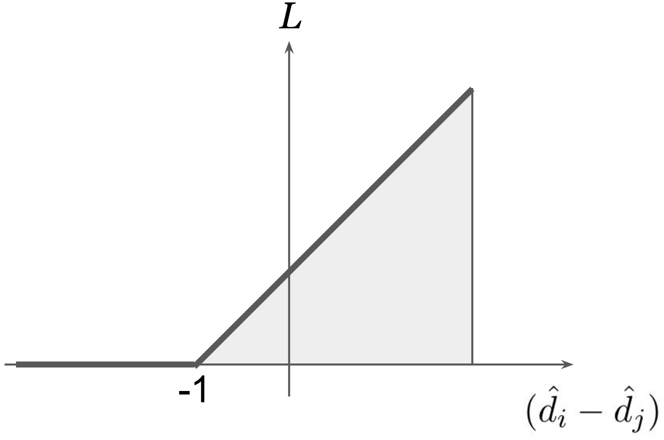
$$\text{sign}(x) \quad = \quad \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$$

a variant of hinge loss

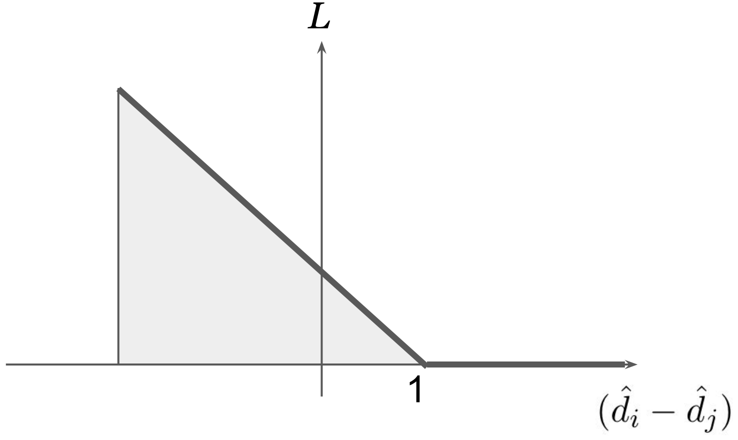
Pairwise learning-to-rank loss for distances

$$L_{\text{dist}}^{\text{rank}} = \sum_{i,j>i} [1 - \text{sign}(d_i - d_j)(\hat{d}_i - \hat{d}_j)]^+$$

While $d_i > d_j$:



While $d_i < d_j$:

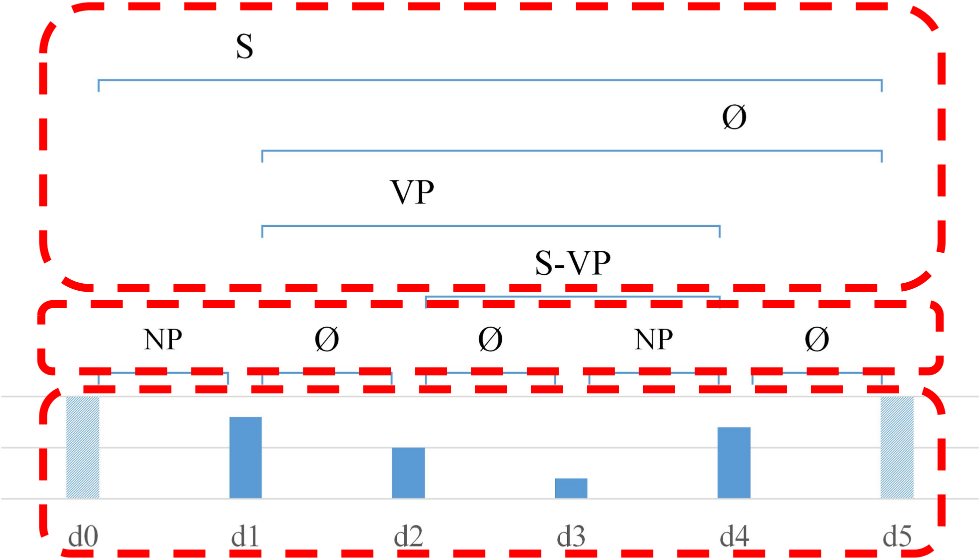


Framework for inferring the distances and labels

Labels for non-leaf nodes

Labels for leaf nodes

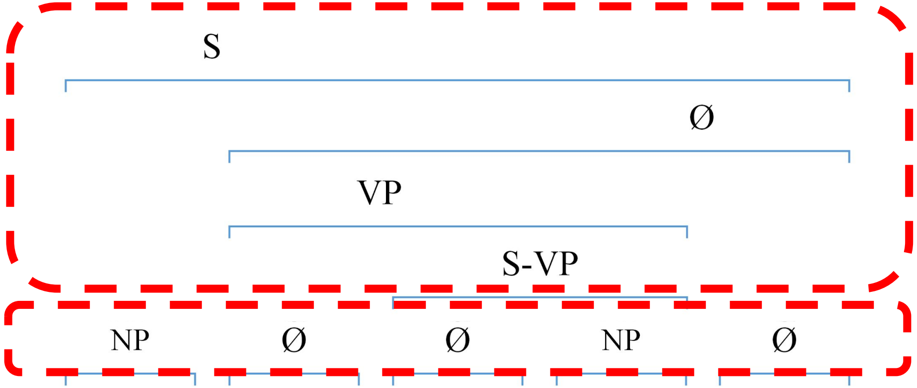
Distances



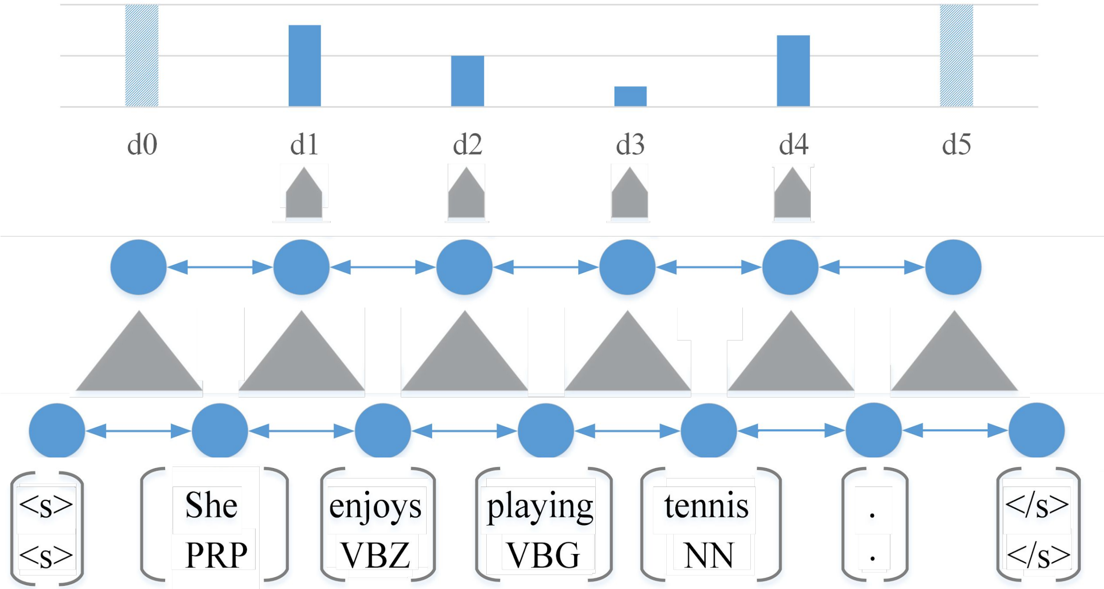
Framework for inferring the distances and labels

Labels for non-leaf nodes

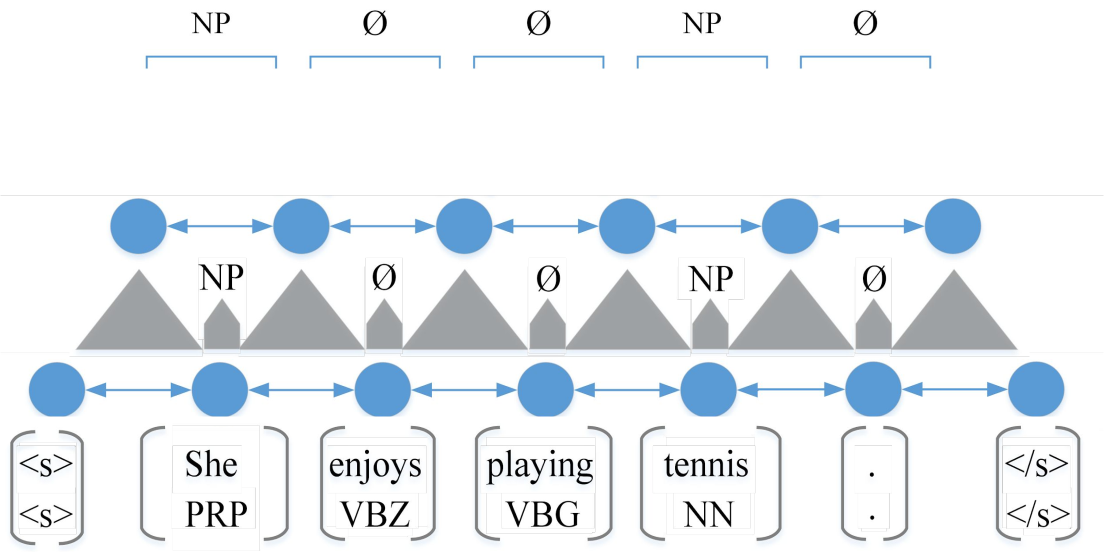
Labels for leaf nodes



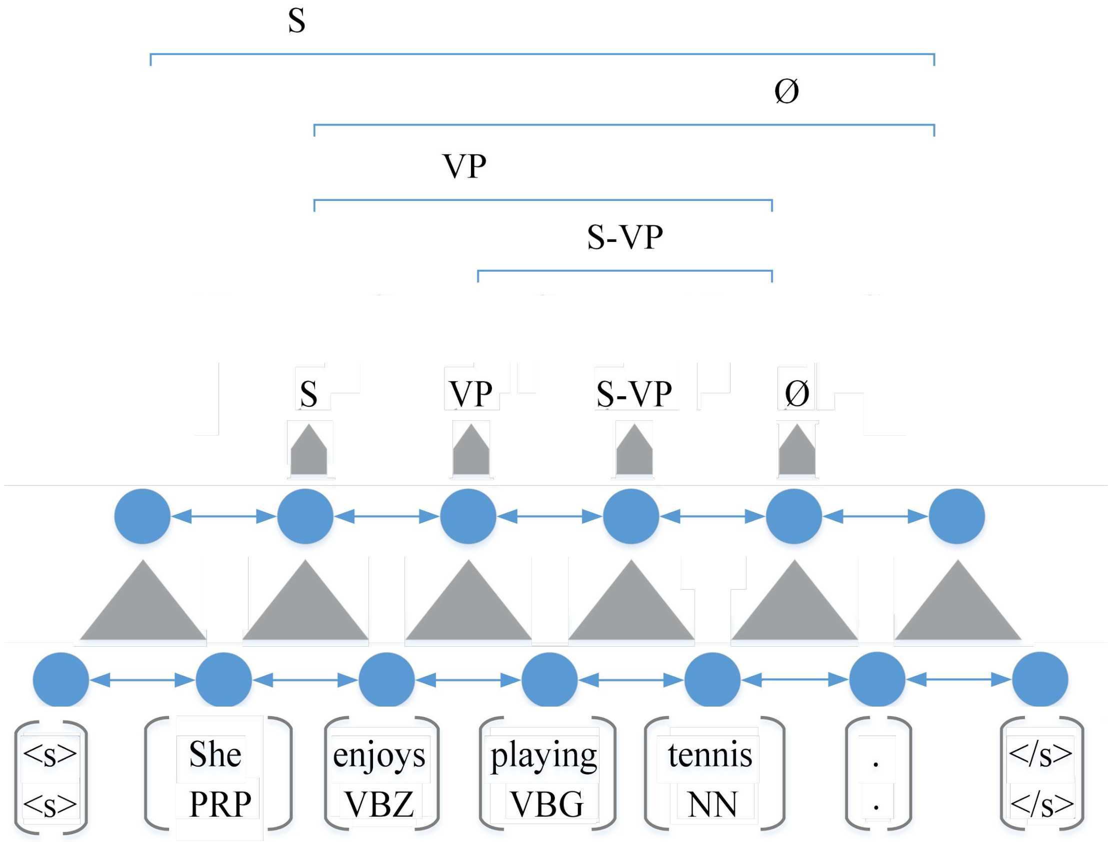
Inferring the Labels



Inferring the Labels

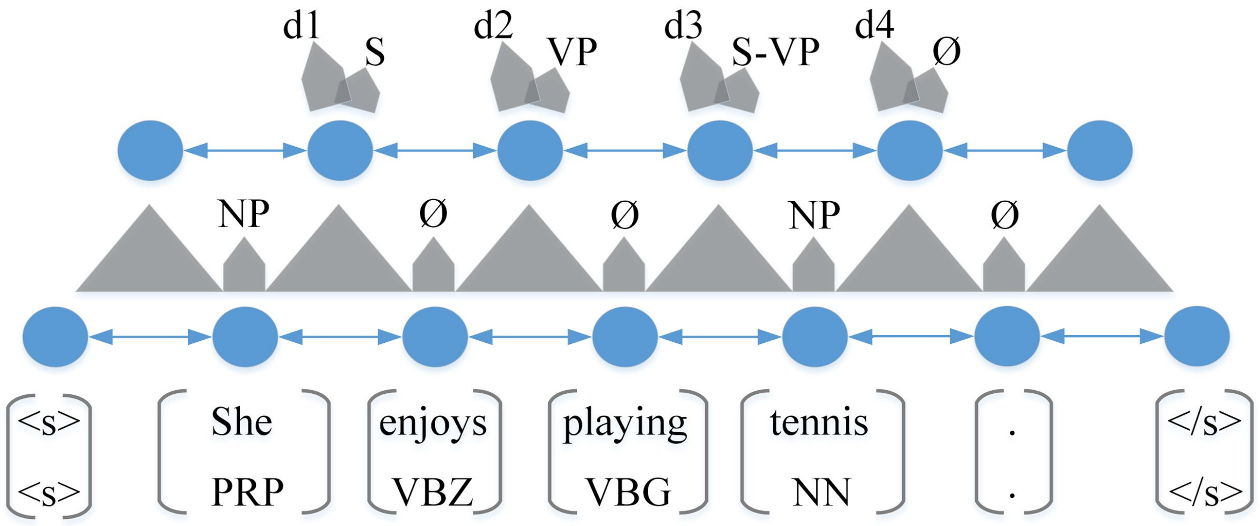


Inferring the Labels

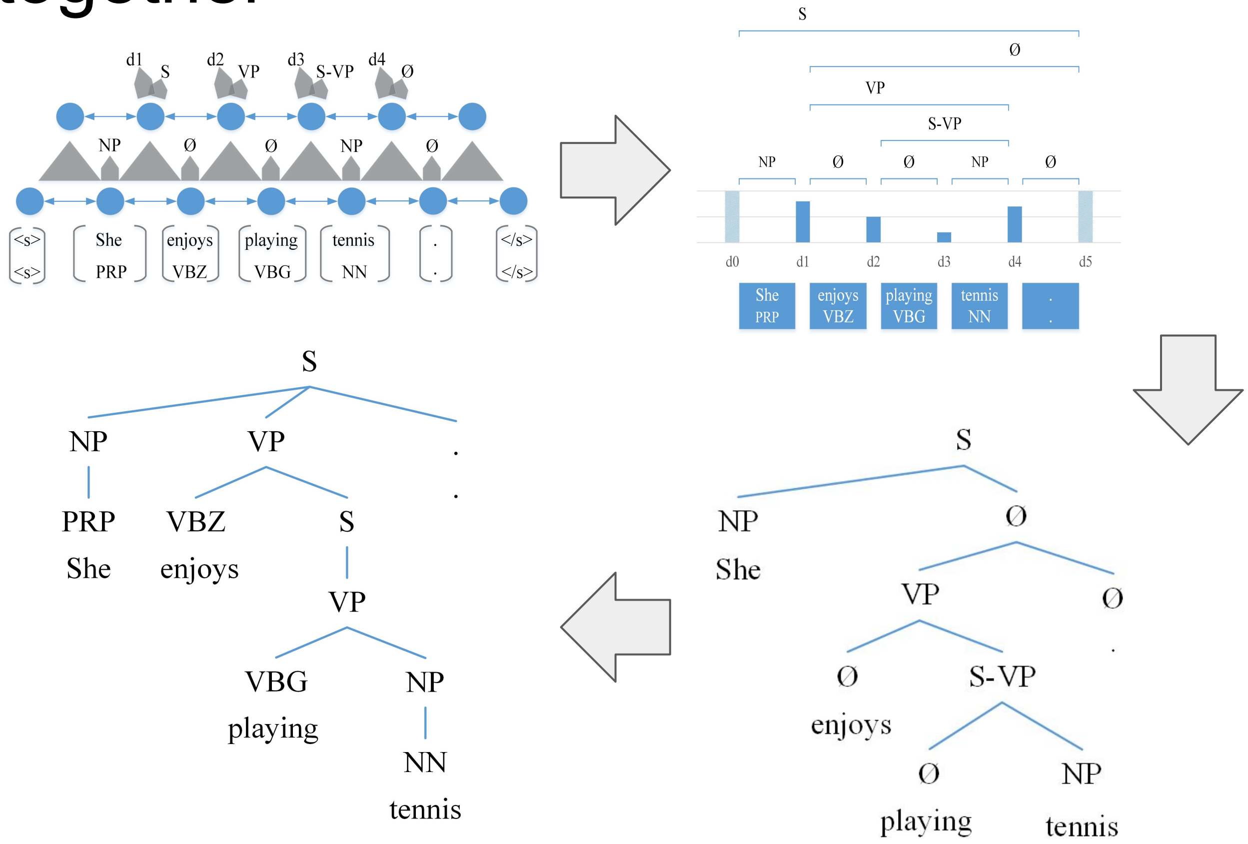


Putting it together

$$L = L_{\text{label}} + L_{\text{dist}}^{\text{rank}}$$



Putting it together



Overview

- Motivation
- Syntactic Distance based Parsing Framework
- Model
- **Experimental Results**

Experiments: Penn Treebank

Model	LP	LR	F1
Single Model			
Vinyals et al. (2015)	-	-	88.3
Zhu et al. (2013)	90.7	90.2	90.4
Dyer et al. (2016)	-	-	89.8
Watanabe and Sumita (2015)	-	-	90.7
Cross and Huang (2016)	92.1	90.5	91.3
Liu and Zhang (2017b)	92.1	91.3	91.7
Stern et al. (2017a)	93.2	90.3	91.8
Liu and Zhang (2017a)	-	-	91.8
Gaddy et al. (2018)	-	-	92.1
Stern et al. (2017b)	92.5	92.5	92.5
Our Model	92.0	91.7	91.8

Ensemble			
Shindo et al. (2012)	-	-	92.4
Vinyals et al. (2015)	-	-	90.5
Semi-supervised			
Zhu et al. (2013)	91.5	91.1	91.3
Vinyals et al. (2015)	-	-	92.8
Re-ranking			
Charniak and Johnson (2005)	91.8	91.2	91.5
Huang (2008)	91.2	92.2	91.7
Dyer et al. (2016)	-	-	93.3

Experiments: Chinese Treebank

Model	LP	LR	F1
Single Model			
Charniak (2000)	82.1	79.6	80.8
Zhu et al. (2013)	84.3	82.1	83.2
Wang et al. (2015)	-	-	83.2
Watanabe and Sumita (2015)	-	-	84.3
Dyer et al. (2016)	-	-	84.6
Liu and Zhang (2017b)	85.9	85.2	85.5
Liu and Zhang (2017a)	-	-	86.1
Our Model	86.6	86.4	86.5

Semi-supervised			
Zhu et al. (2013)	86.8	84.4	85.6
Wang and Xue (2014)	-	-	86.3
Wang et al. (2015)	-	-	86.6
Re-ranking			
Charniak and Johnson (2005)	83.8	80.8	82.3
Dyer et al. (2016)	-	-	86.9

Experiments: Detailed statistics in PTB and CTB

dev/test result		Prec.	Recall	F1	label accuracy
PTB	labeled	91.7/92.0	91.8/91.7	91.8/91.8	94.9/95.4%
	unlabeled	93.0/93.2	93.0/92.8	93.0/93.0	
CTB	labeled	89.4/86.6	89.4/86.4	89.4/86.5	92.2/91.1%
	unlabeled	91.1/88.9	91.1/88.6	91.1/88.8	

Experiments: Ablation Test

Model	LP	LR	F1
Full model	92.0	91.7	91.8
w/o top LSTM	91.0	90.5	90.7
w. Char LSTM	92.1	91.7	91.9
w. embedding	91.9	91.6	91.7
w. MSE loss	90.3	90.0	90.1

Experiments: Parsing Speed

Model	# sents/sec
Petrov and Klein (2007)	6.2
Zhu et al. (2013)	89.5
Liu and Zhang (2017b)	79.2
Stern et al. (2017a)	75.5
Our model	111.1
Our model w/o tree inference	351

Conclusions and Highlights

- **A novel constituency parsing scheme**: predicting tree structure from a set of real-valued scalars (syntactic distances).
- Completely **free from compounding errors**.
- **Strong performance** compare to previous models, and
- **Significantly more efficient** than previous models
- **Easy deployment**: The architecture of model is no more than a stack of standard recurrent and convolutional layers.

One more thing...

- The research in rank loss is well-studied in the topic of learning-to-rank, since 2005 (Burges et al. 2005).
- Models that are good at learning these syntactic distances are not widely known until the rediscovery of LSTM in 2013 (Graves 2013).
- Efficient regularization methods for LSTM didn’t become mature until 2017 (Merity 2017).

Thank you!

Yikang Shen, Zhouhan Lin
MILA, Université de Montréal
{yikang.shn, lin.zhouhan}@gmail.com

Questions?

Code:

Paper: