Cooperative Learning of Disjoint Syntax and Semantics

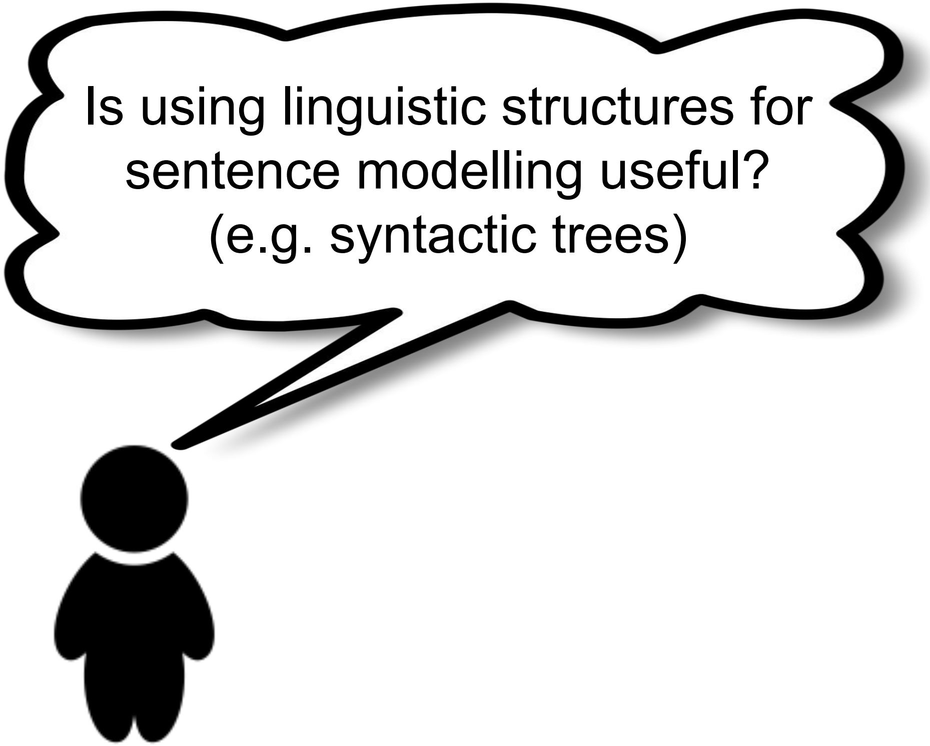
Serhii Havrylov

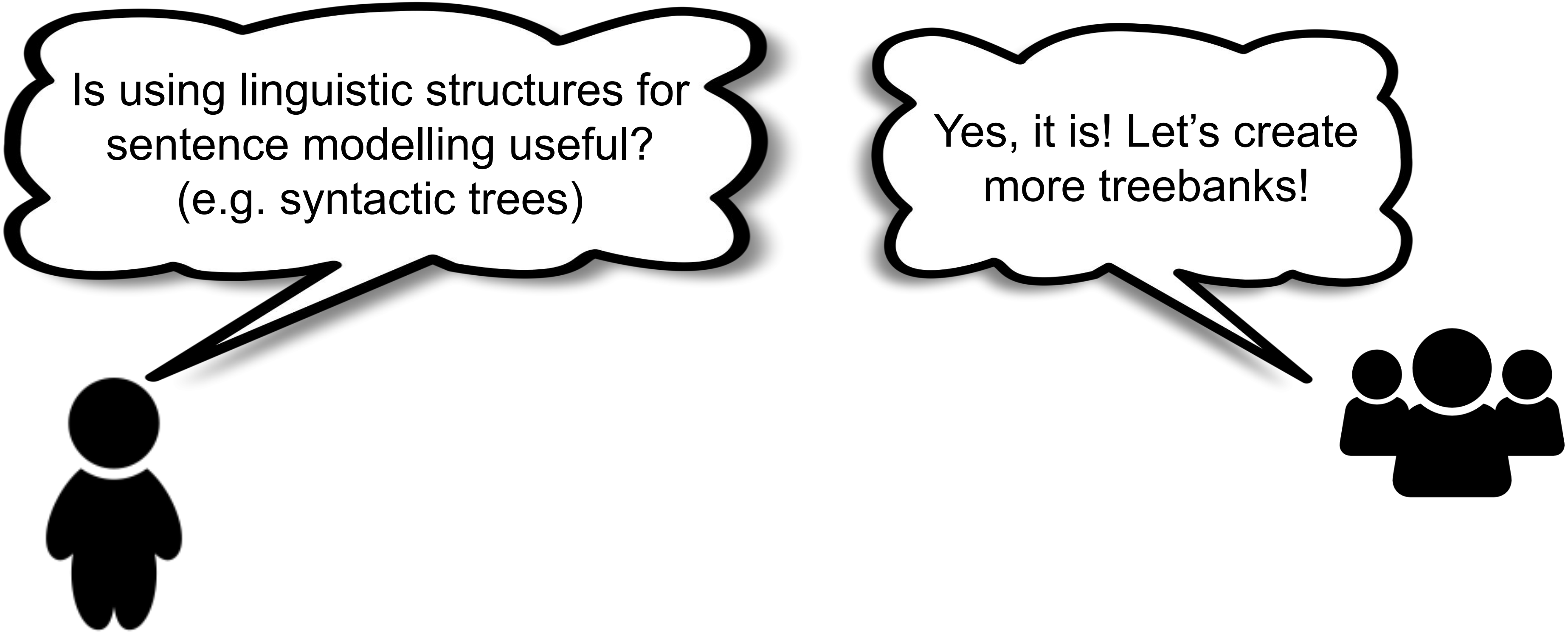
The University of Edinburgh

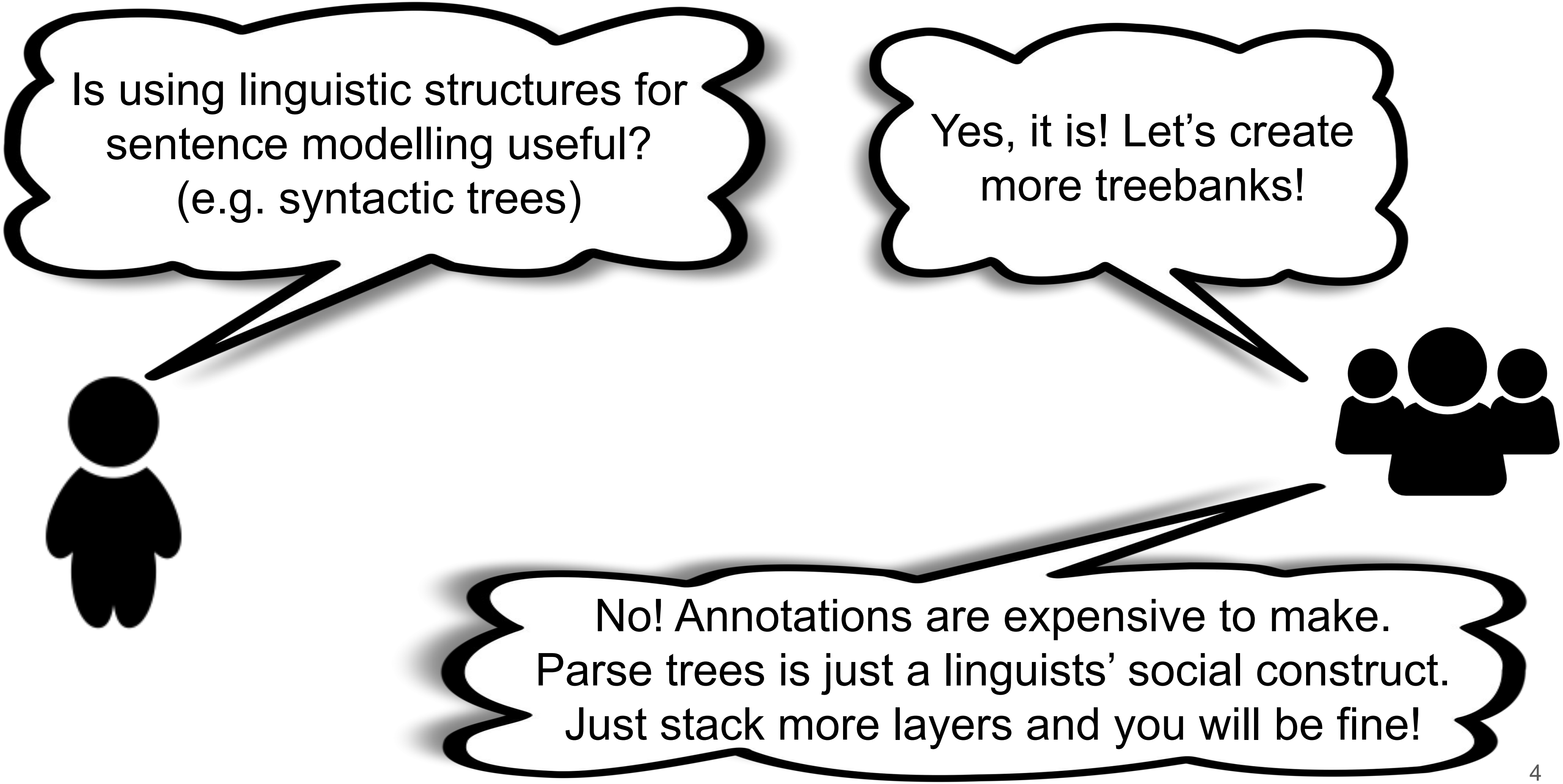
Edinburgh University of Edinburgh Natural Language Processing NLP

Germán Kruszewski
Armand Joulin

Facebook AI Research


Is using linguistic structures for
sentence modelling useful?
(e.g. syntactic trees)


Is using linguistic structures for sentence modelling useful? (e.g. syntactic trees)
Yes, it is! Let’s create more treebanks!


Is using linguistic structures for sentence modelling useful? (e.g. syntactic trees)
Yes, it is! Let’s create more treebanks!
No! Annotations are expensive to make. Parse trees is just a linguists’ social construct. Just stack more layers and you will be fine!

Recursive neural network

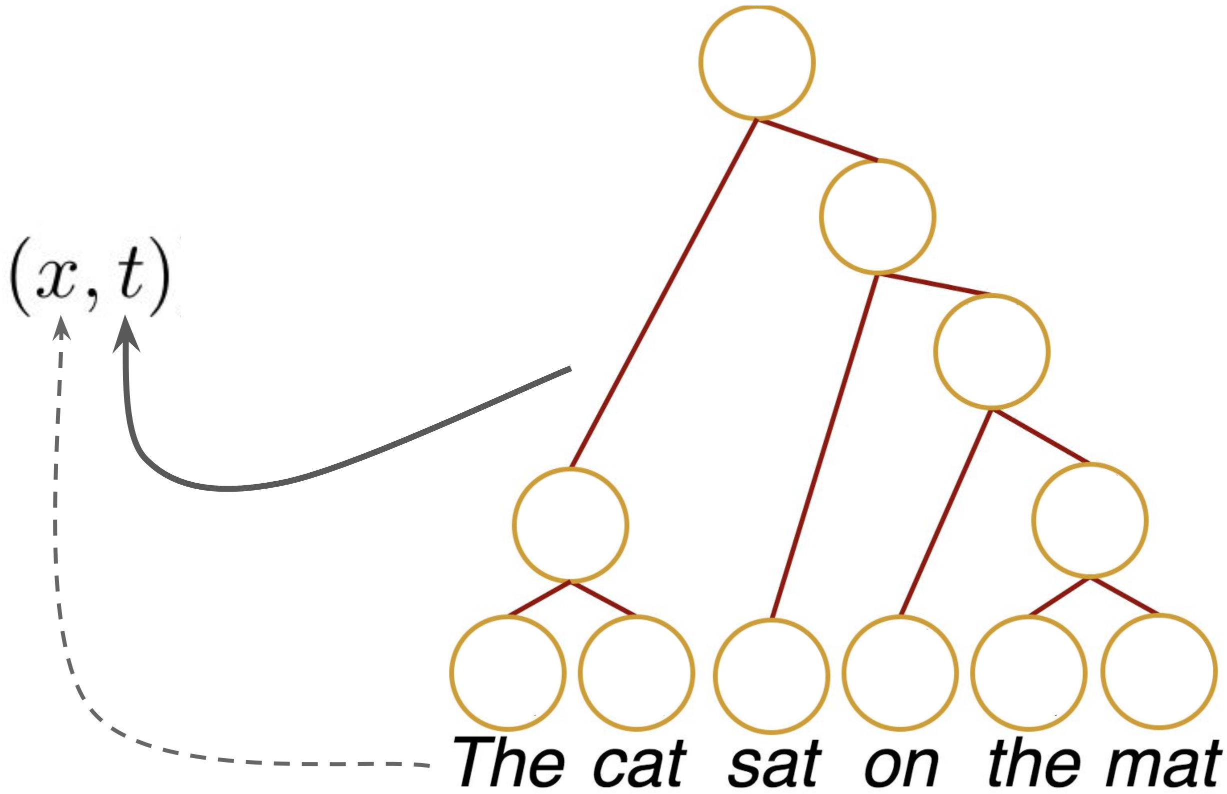
The cat sat on the mat

Recursive neural network

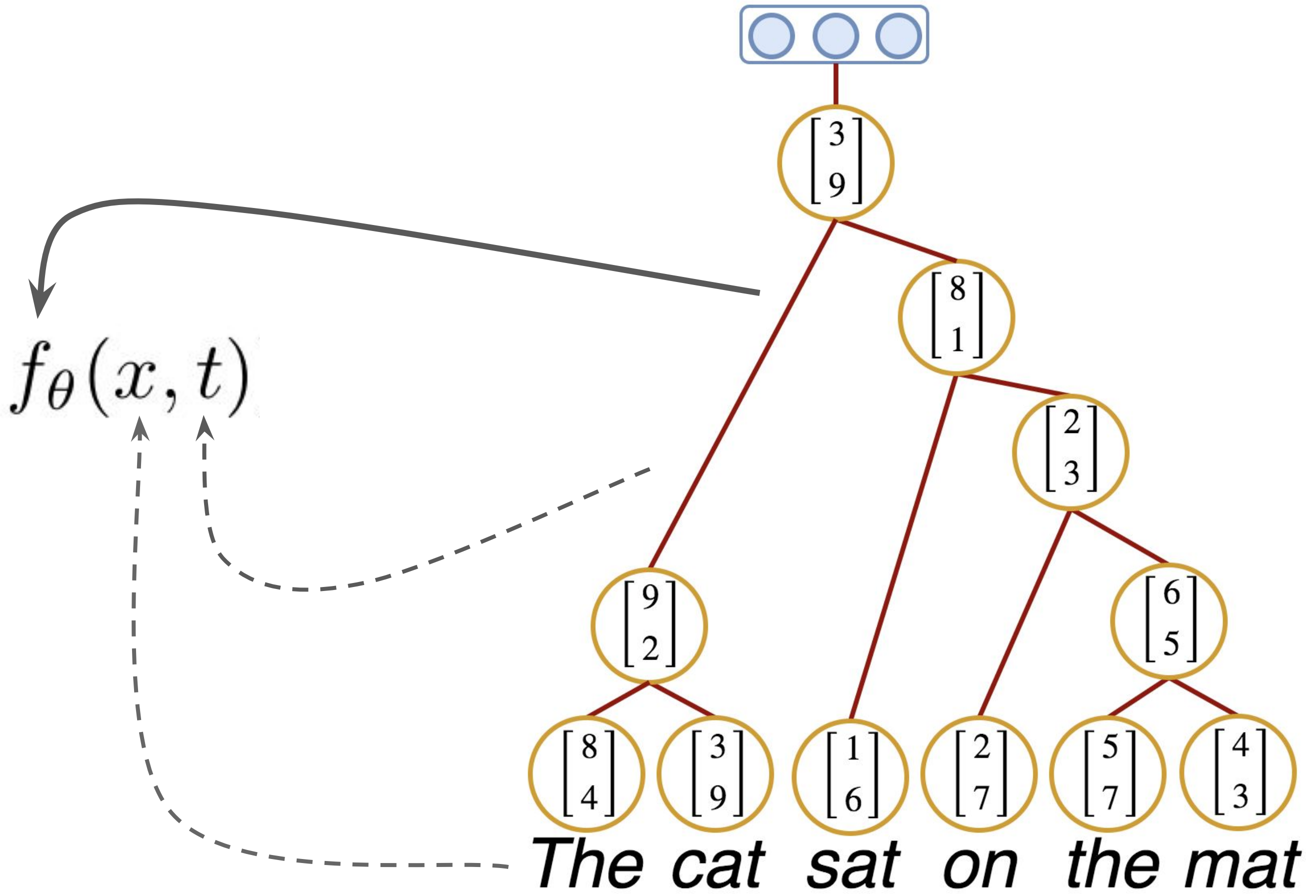
x

The cat sat on the mat

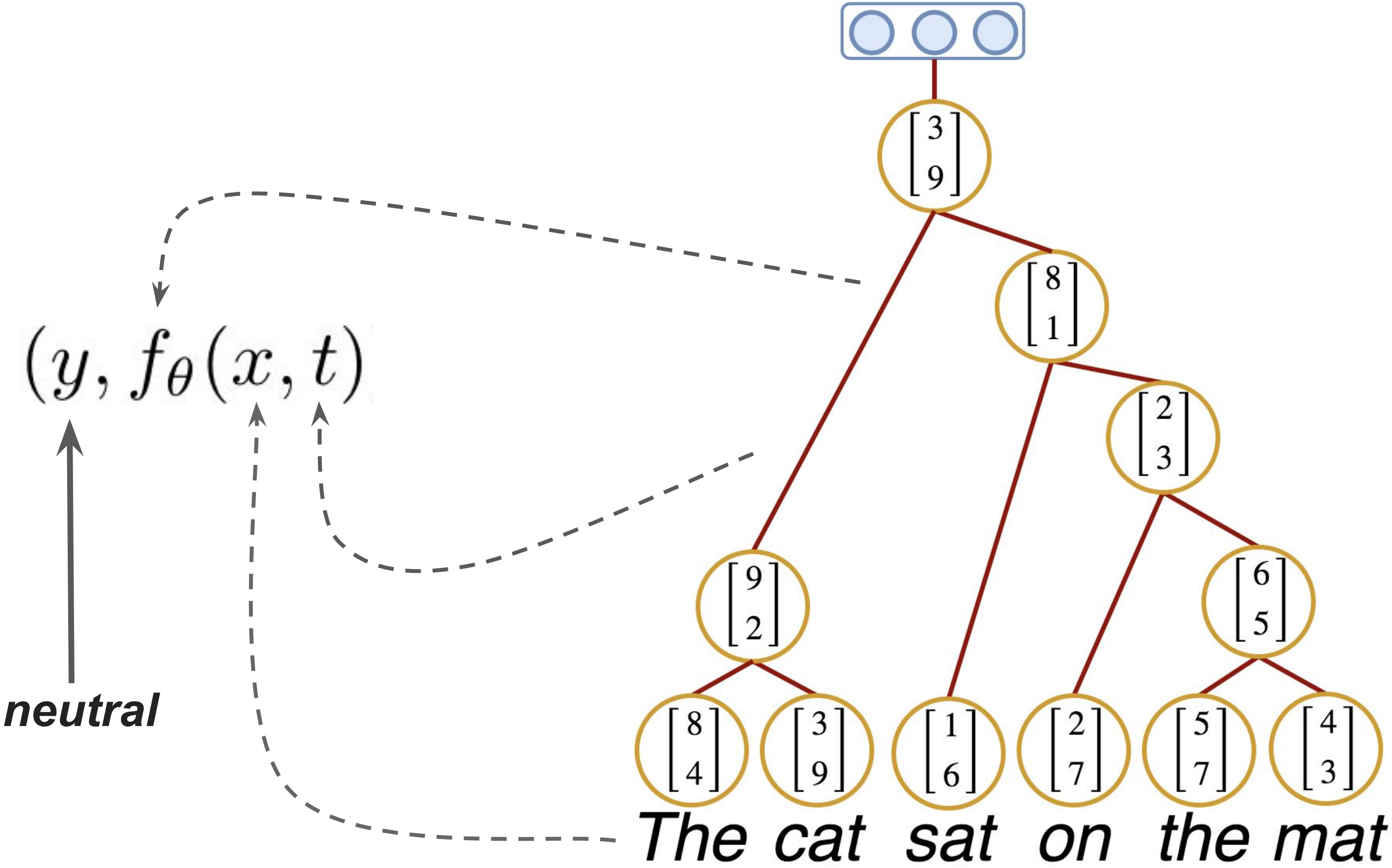
Recursive neural network



Recursive neural network



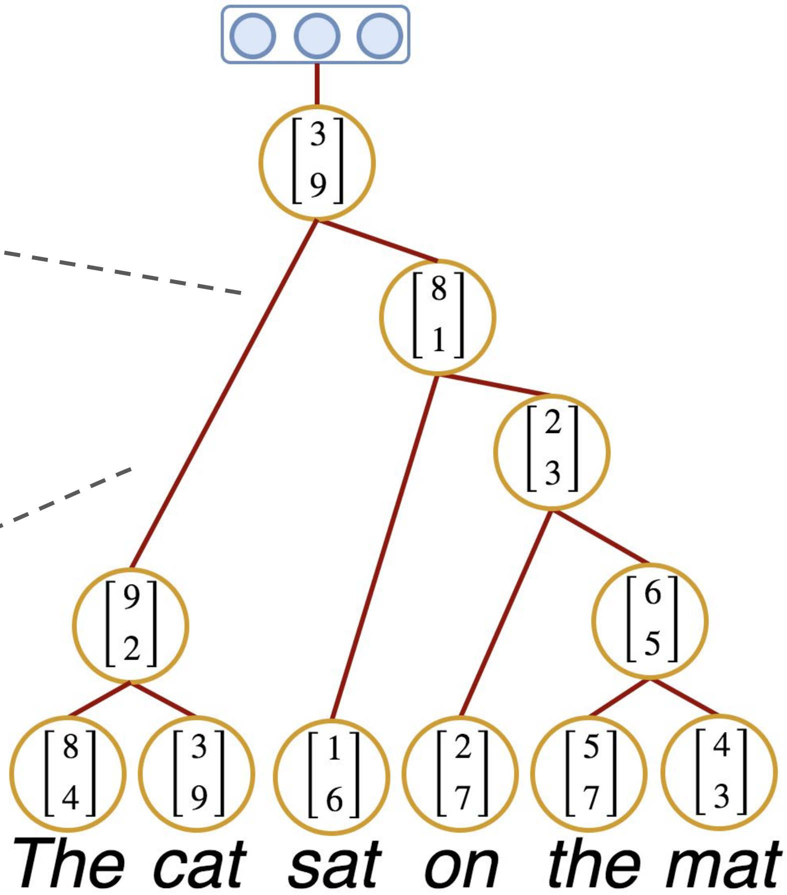
Recursive neural network



Recursive neural network

$\ell(y, f_\theta(x, t))$

neutral



Latent tree learning

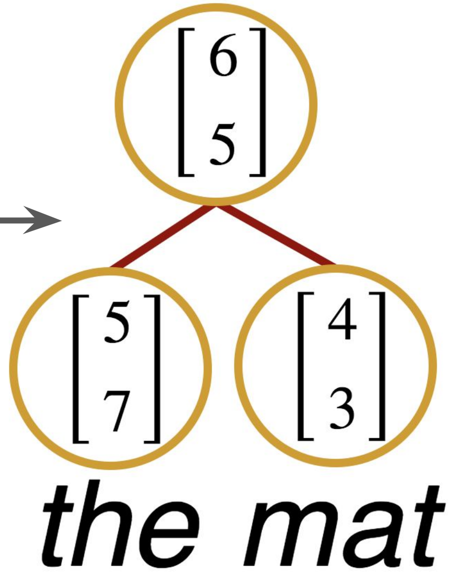
$$\ell(y, f_\theta(x, t))$$

Latent tree learning

$$\mathbb{E}_{p_\phi(t|x)}[\ell(y, f_\theta(x, t))]$$

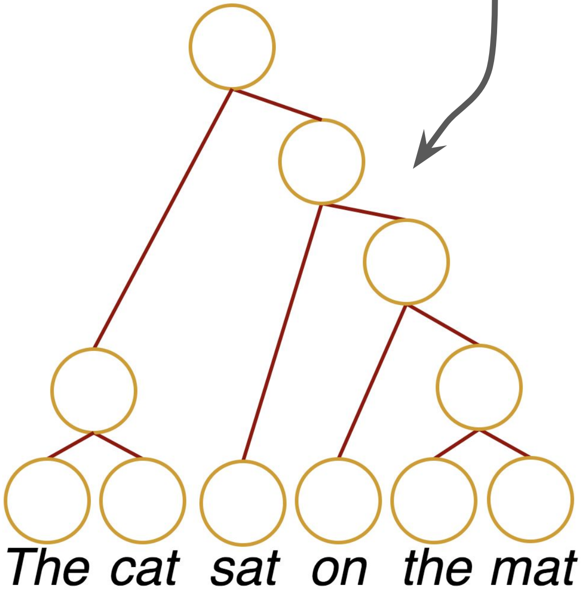
Latent tree learning

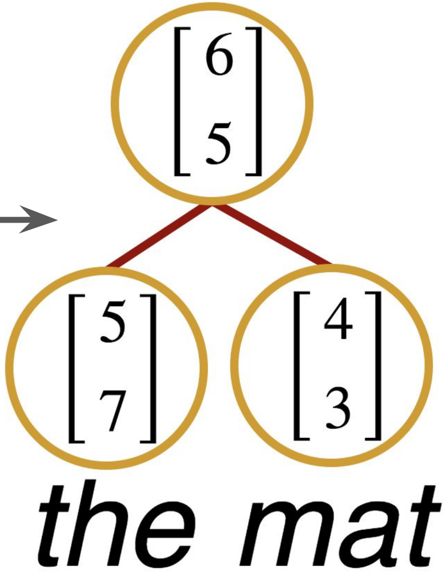
$$\mathbb{E}_{p_\phi(t|x)}[\ell(y, f_\theta(x,t))]$$



Latent tree learning

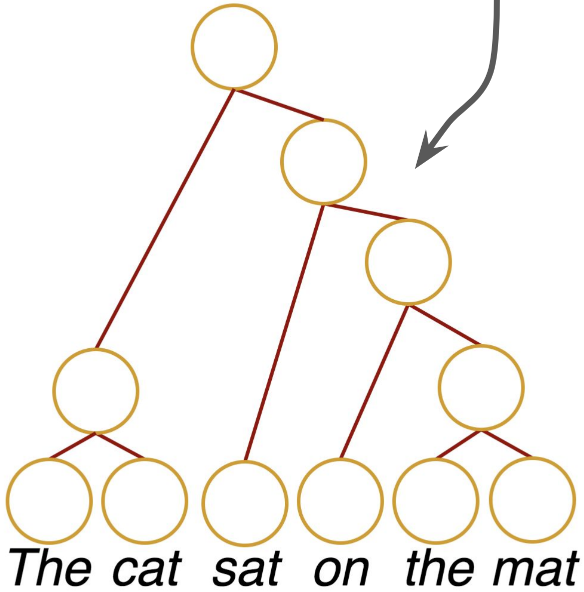
$$\mathbb{E}_{p_\phi(t|x)}[\ell(y, f_\theta(x, t))]$$

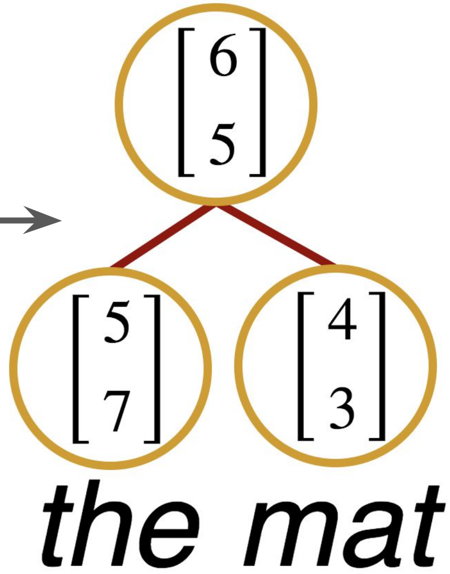




Latent tree learning

$$\mathbb{E}_{p_\phi(t|x)}[\ell(y, f_\theta(x, t))]$$





Latent tree learning

- RL-SPINN: Yogatama et al., 2016
- Soft-CYK: Maillard et al., 2017
- Gumbel Tree-LSTM: Choi et al., 2018

Latent tree learning

- RL-SPINN: Yogatama et al., 2016
- Soft-CYK: Maillard et al., 2017
- Gumbel Tree-LSTM: Choi et al., 2018

Recent work has shown that:

- Trees **do not resemble** any semantic or syntactic formalisms (Williams et al. 2018).

Latent tree learning

- RL-SPINN: Yogatama et al., 2016
- Soft-CYK: Maillard et al., 2017
- Gumbel Tree-LSTM: Choi et al., 2018

Recent work has shown that:

- Trees **do not resemble** any semantic or syntactic formalism (Williams et al. 2018).
- Parsing strategies **are not consistent** across random restarts (Williams et al. 2018).

Latent tree learning

- RL-SPINN: Yogatama et al., 2016
- Soft-CYK: Maillard et al., 2017
- Gumbel Tree-LSTM: Choi et al., 2018

Recent work has shown that:

- Trees **do not resemble** any semantic or syntactic formalisms (Williams et al. 2018).
- Parsing strategies **are not consistent** across random restarts (Williams et al. 2018).
- These models **fail to learn the simple context-free grammar** (Nangia et al. 2018).

ListOps (Nangia, & Bowman (2018))

[MIN 1 [MAX [MIN 9 [MAX 1 0] 2 9 [MED 8 4 3]] [MIN 7 5] 6 9 3]]

[MAX 1 4 0 9]

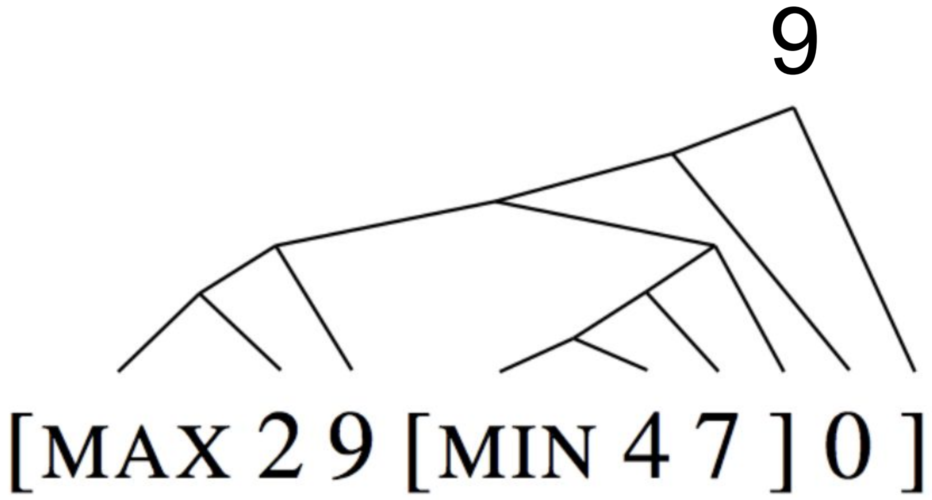
[MAX 7 1 [MAX 6 8 1 7] [MIN 2 6] 3]

ListOps (Nangia, & Bowman (2018))

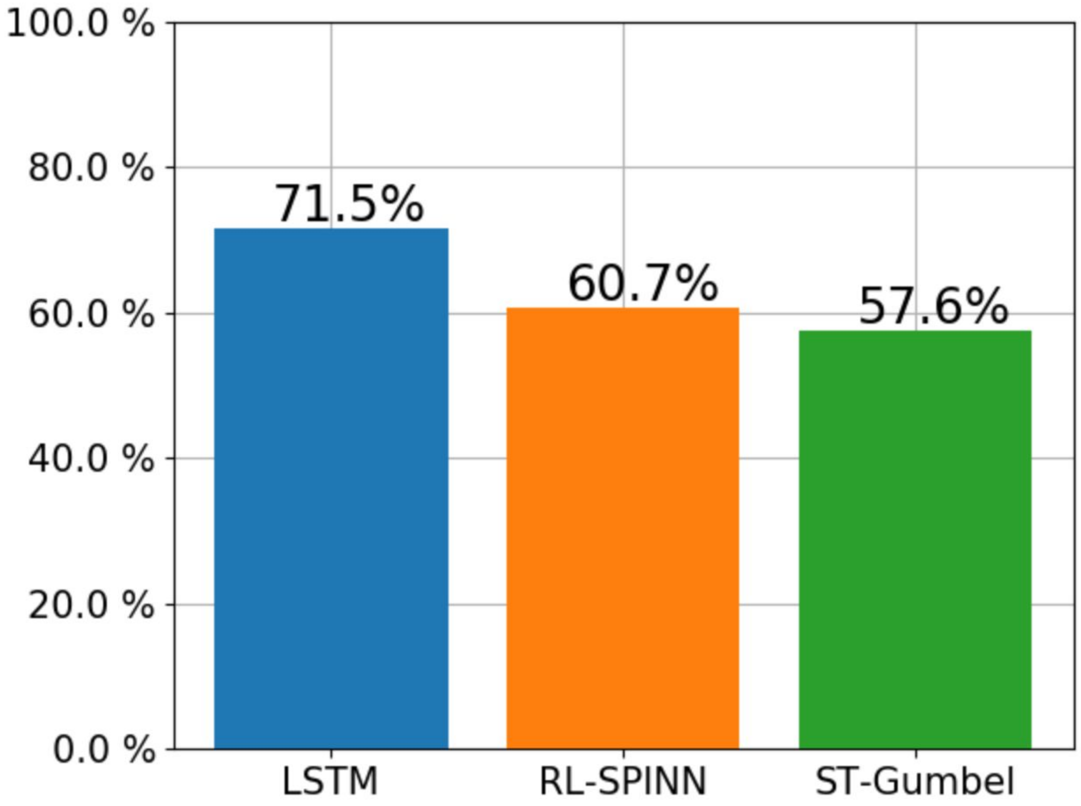
[MIN 1 [MAX [MIN 9 [MAX 1 0] 2 9 [MED 8 4 3]] [MIN 7 5] 6 9 3]]

[MAX 1 4 0 9]

[MAX 7 1 [MAX 6 8 1 7] [MIN 2 6] 3]

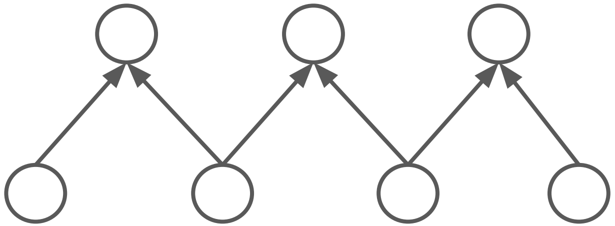


ListOps (Nangia, & Bowman (2018))



Tree-LSTM parser (Choi et al., 2018)

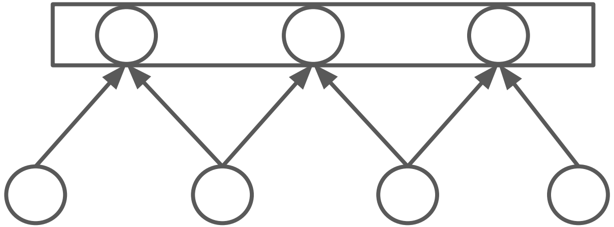
Tree-LSTM parser (Choi et al., 2018)



$$\mathbf{r}_i^{k+1} = \texttt{Tree-LSTM}(\mathbf{r}_i^k, \mathbf{r}_{i+1}^k)$$

Tree-LSTM parser (Choi et al., 2018)

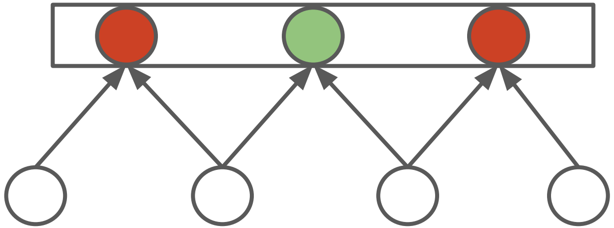
$$s_k(i) = \langle \mathbf{q}, \mathbf{r}_i^{k+1} \rangle$$



$$\mathbf{r}_i^{k+1} = \texttt{Tree-LSTM}(\mathbf{r}_i^k, \mathbf{r}_{i+1}^k)$$

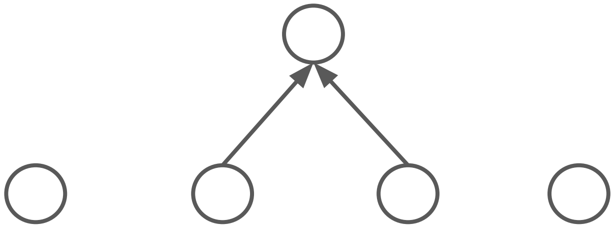
Tree-LSTM parser (Choi et al., 2018)

$$s_k(i) = \langle \mathbf{q}, \mathbf{r}_i^{k+1} \rangle$$

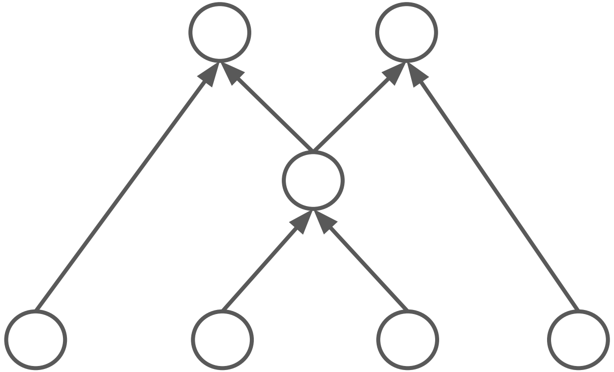


$$\mathbf{r}_i^{k+1} = \texttt{Tree-LSTM}(\mathbf{r}_i^k, \mathbf{r}_{i+1}^k)$$

Tree-LSTM parser (Choi et al., 2018)



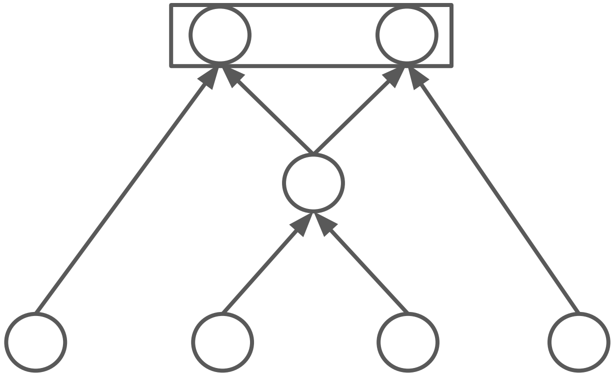
Tree-LSTM parser (Choi et al., 2018)



$$\mathbf{r}_i^{k+1} = \texttt{Tree-LSTM}(\mathbf{r}_i^k, \mathbf{r}_{i+1}^k)$$

Tree-LSTM parser (Choi et al., 2018)

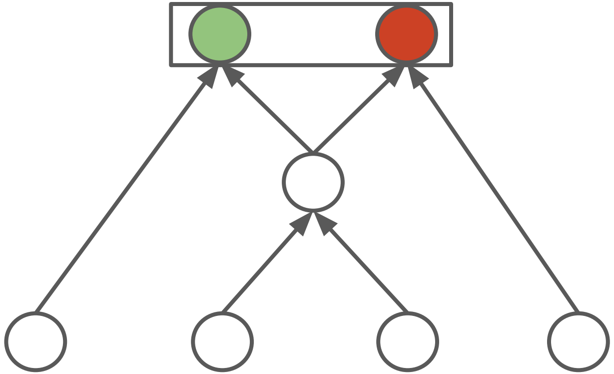
$$s_k(i) = \langle \mathbf{q}, \mathbf{r}_i^{k+1} \rangle$$



$$\mathbf{r}_i^{k+1} = \texttt{Tree-LSTM}(\mathbf{r}_i^k, \mathbf{r}_{i+1}^k)$$

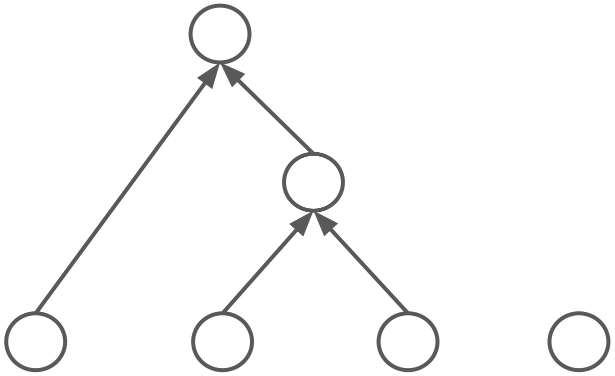
Tree-LSTM parser (Choi et al., 2018)

$$s_k(i) = \langle \mathbf{q}, \mathbf{r}_i^{k+1} \rangle$$

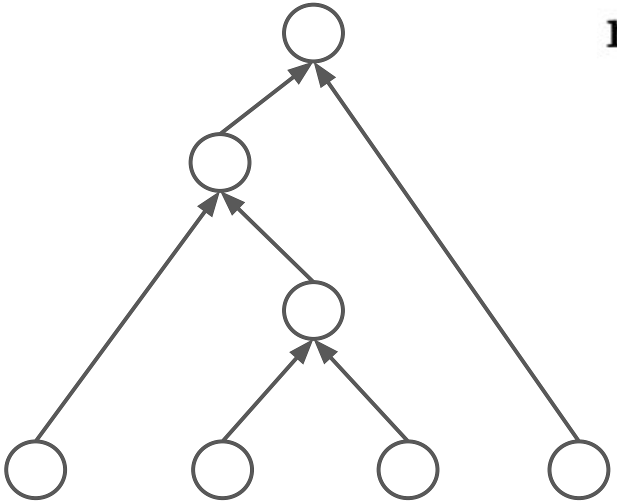


$$\mathbf{r}_i^{k+1} = \texttt{Tree-LSTM}(\mathbf{r}_i^k, \mathbf{r}_{i+1}^k)$$

Tree-LSTM parser (Choi et al., 2018)



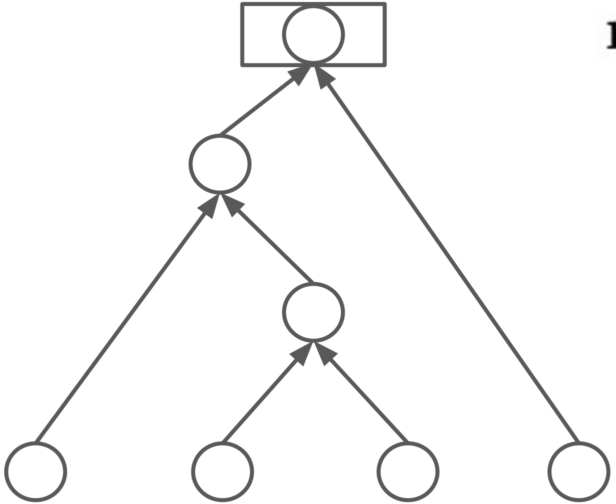
Tree-LSTM parser (Choi et al., 2018)



$$\mathbf{r}_i^{k+1} = \texttt{Tree-LSTM}(\mathbf{r}_i^k, \mathbf{r}_{i+1}^k)$$

Tree-LSTM parser (Choi et al., 2018)

$$s_k(i) = \langle \mathbf{q}, \mathbf{r}_i^{k+1} \rangle$$

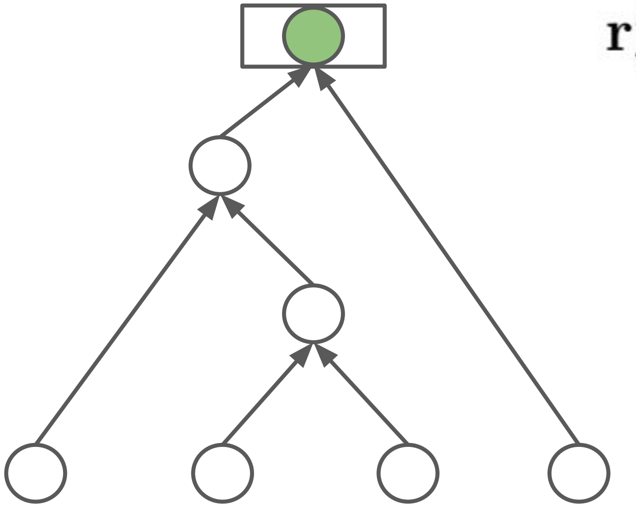


$$\mathbf{r}_i^{k+1} = \texttt{Tree-LSTM}(\mathbf{r}_i^k, \mathbf{r}_{i+1}^k)$$

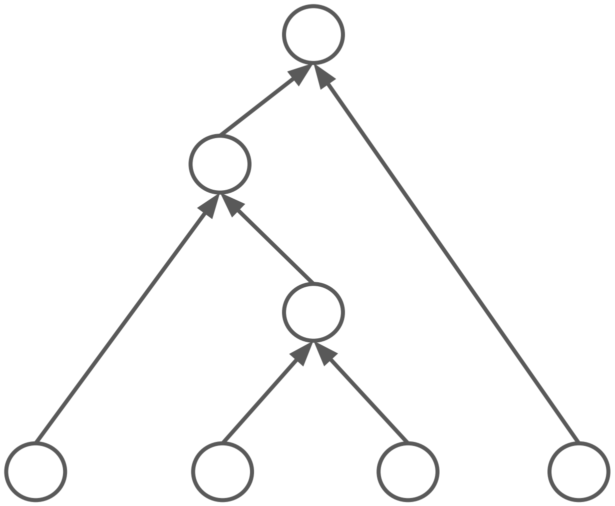
Tree-LSTM parser (Choi et al., 2018)

$$s_k(i) = \langle \mathbf{q}, \mathbf{r}_i^{k+1} \rangle$$

$$\mathbf{r}_i^{k+1} = \texttt{Tree-LSTM}(\mathbf{r}_i^k, \mathbf{r}_{i+1}^k)$$



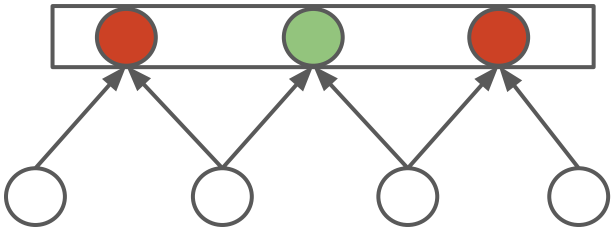
Tree-LSTM parser (Choi et al., 2018)



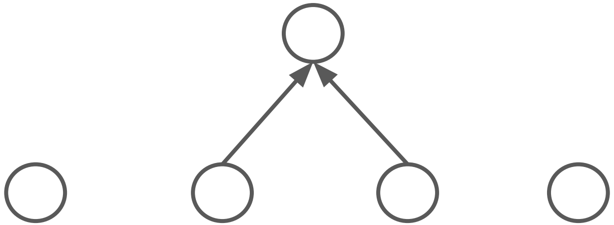
Separation of syntax and semantics

Parser ϕ | Compositional Function θ

$$s_k(i) = \langle \mathbf{q}, \texttt{Tree-LSTM}(\mathbf{r}_i^k, \mathbf{r}_{i+1}^k) \rangle$$

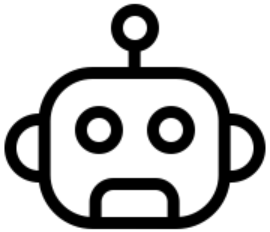


$$\mathbf{r}_i^{k+1} = \texttt{Tree-LSTM}(\mathbf{r}_i^k, \mathbf{r}_{i+1}^k)$$

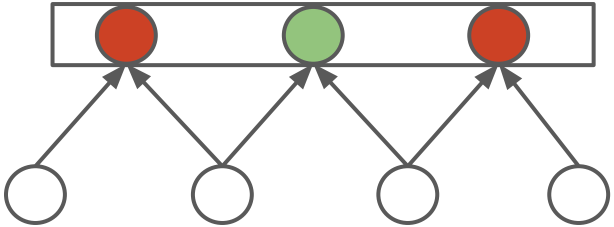


Parsing as a RL problem

Parser ϕ

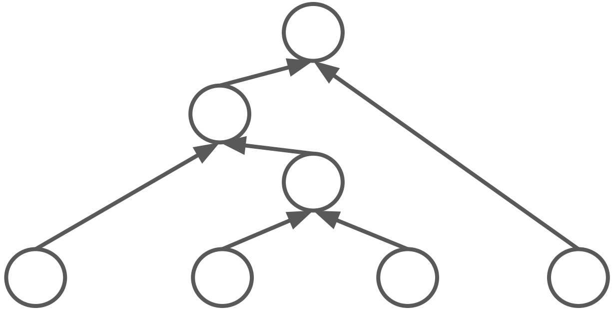


$$p_\phi(t|x) = \prod_{k=0}^{K} \pi_\phi(a_k^i|\mathbf{r}^k)$$



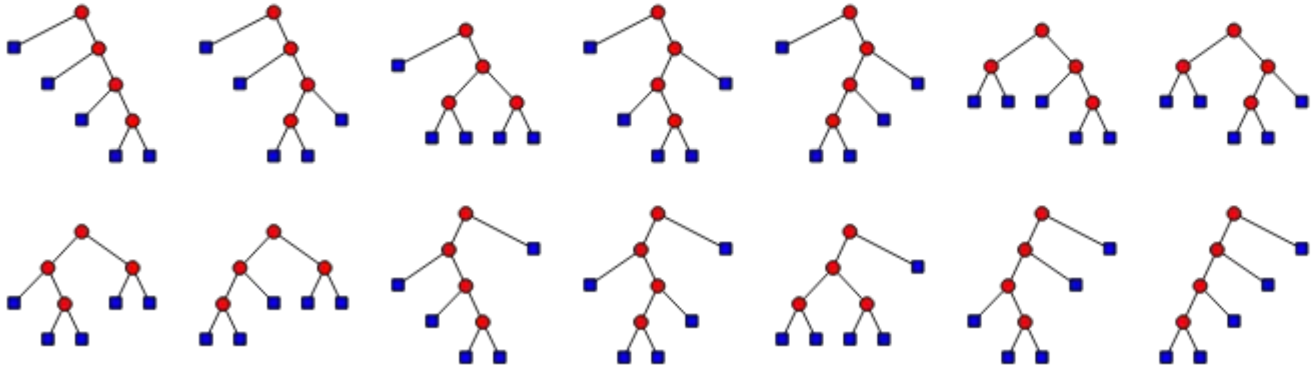
Compositional Function θ

$$\ell(f_\theta(x,t),y)$$



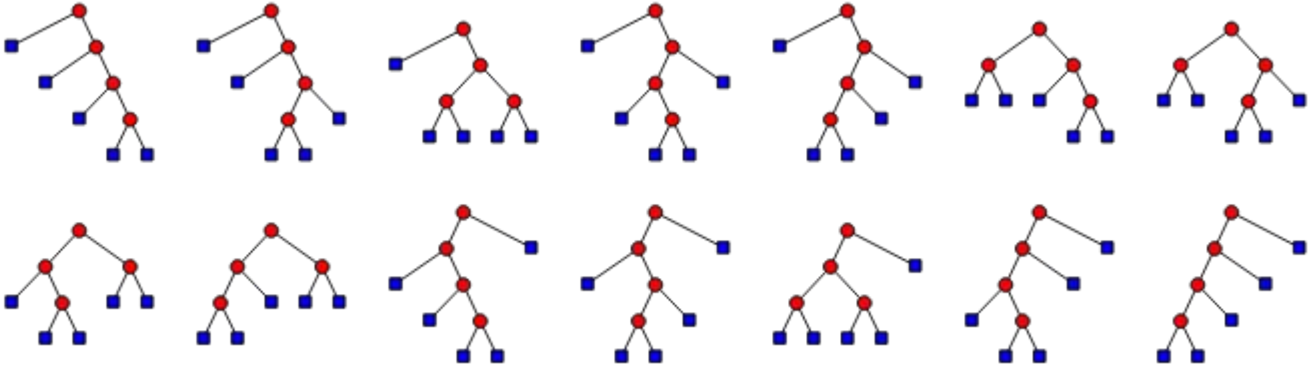
Optimization challenges

Size of the search space is $$C_n \sim \frac{4^n}{n^{3/2}\sqrt{\pi}}$$



Optimization challenges

Size of the search space is $$C_n \sim \frac{4^n}{n^{3/2}\sqrt{\pi}}$$



For a sentence with 20 words, there are **1_767_263_190** possible trees.

Optimization challenges

Syntax and semantic has to be learnt simultaneously

model has to infer from examples that `[MIN 0 1] = 0`

Optimization challenges

Syntax and semantic has to be learnt simultaneously

model has to infer from examples that `[MIN 0 1] = 0`

– nonstationary environment (i.e the same sequence of actions can receive different rewards)

Optimization challenges

Typically, the ***compositional function*** θ is learned faster than the ***parser*** φ.

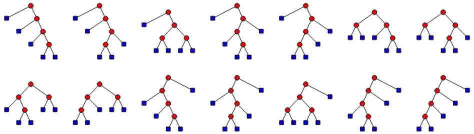
Optimization challenges

Typically, the ***compositional function*** θ is learned faster than the ***parser*** φ.

This fast coadaptation limits the exploration of the search space to parsing strategies similar to those found at the beginning of the training.

Optimization challenges

- High variance in the estimate of a parser's gradient ∇_{φ} has to be addressed.



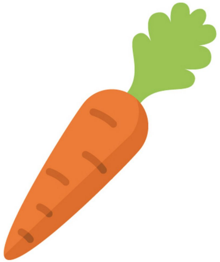
- Learning paces of a parser θ and a compositional function φ have to be levelled off.

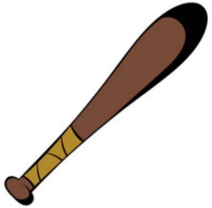
Variance reduction

$$\nabla_\phi \mathcal{L} \approx \ell(f_\theta(x,t), y) \frac{\partial \log p_\phi(t|x)}{\partial \phi}.$$

Variance reduction

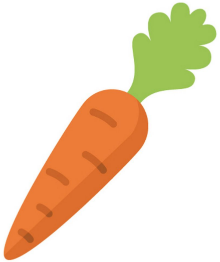
$$\nabla_\phi \mathcal{L} \approx \underbrace{\ell(f_\theta(x,t),y)}_{\textbf{reward}} \frac{\partial \log p_\phi(t|x)}{\partial \phi}.$$

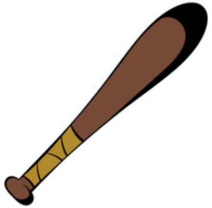


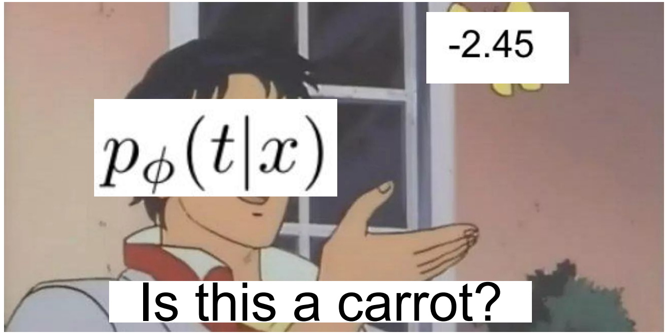


Variance reduction

$$\nabla_\phi \mathcal{L} \approx \underbrace{\ell(f_\theta(x,t),y)}_{\textbf{reward}} \frac{\partial \log p_\phi(t|x)}{\partial \phi}.$$







Variance reduction

$$\nabla_\phi \mathcal{L} \approx \ell(f_\theta(x,t),y)\frac{\partial \log p_\phi(t|x)}{\partial \phi}.$$

the moving average of recent rewards

$$\nabla_\phi \mathcal{L} \approx \underbrace{(\ell(f_\theta(x,t),y) - c)}_{\textbf{new reward}}\frac{\partial \log p_\phi(t|x)}{\partial \phi}.$$

Variance reduction

- `[MIN 1 [MAX [MIN 9 [MIN 1 0 ] 2 [MED 8 4 3 ] ] [MAX 7 5 ] 6 9 ] ]`
- `[MAX 1 0 ]`

Variance reduction

- `[MIN 1 [MAX [MIN 9 [MIN 1 0 ] 2 [MED 8 4 3 ] ] [MAX 7 5 ] 6 9 ] ]`
- `[MAX 1 0 ]`

$$\nabla_\phi \mathcal{L} \approx (\ell(f_\theta(x,t),y) - c(x)) \frac{\partial \log p_\phi(t|x)}{\partial \phi}$$

Variance reduction

- `[MIN 1 [MAX [MIN 9 [MIN 1 0 ] 2 [MED 8 4 3 ] ] [MAX 7 5 ] 6 9 ] ]`
- `[MAX 1 0 ]`

$$\nabla_\phi \mathcal{L} \approx (\ell(f_\theta(x,t),y) - c(x)) \frac{\partial \log p_\phi(t|x)}{\partial \phi}$$

Variance reduction

- `[MIN 1 [MAX [MIN 9 [MIN 1 0 ] 2 [MED 8 4 3 ] ] [MAX 7 5 ] 6 9 ] ]`
- `[MAX 1 0 ]`

$$\nabla_\phi \mathcal{L} \approx (\ell(f_\theta(x,t),y) - c(x)) \frac{\partial \log p_\phi(t|x)}{\partial \phi}$$

self-critical training (SCT) baseline Rennie et al. (2017)

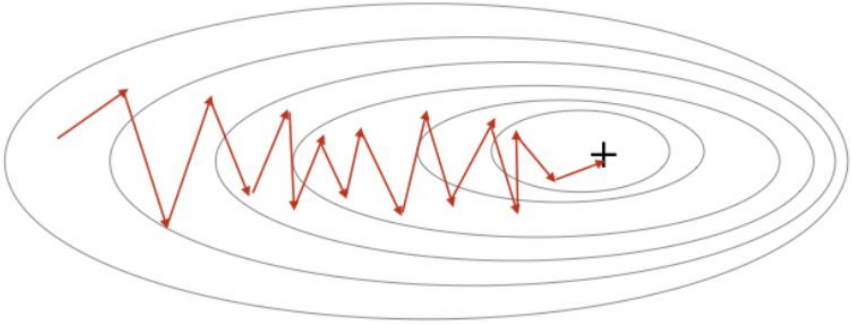
$$c(x) = \ell(f_\theta(x,\hat{t}),y)$$

$$\hat{t} = \arg\max p_\phi(t|x)$$

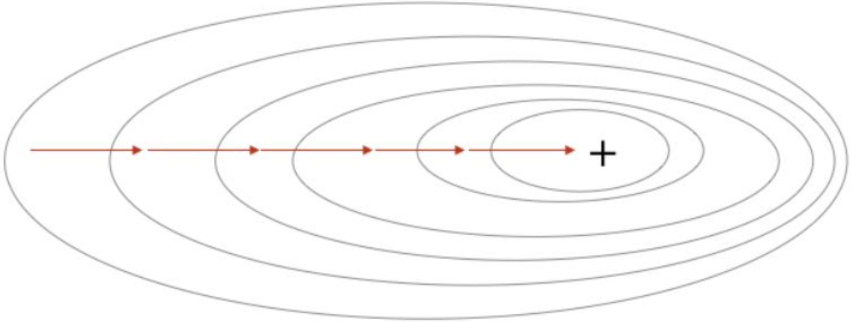
Synchronizing syntax and semantics learning

Syntax  $p_\phi(t|x)$

Semantics  $f_\theta(x, t)$

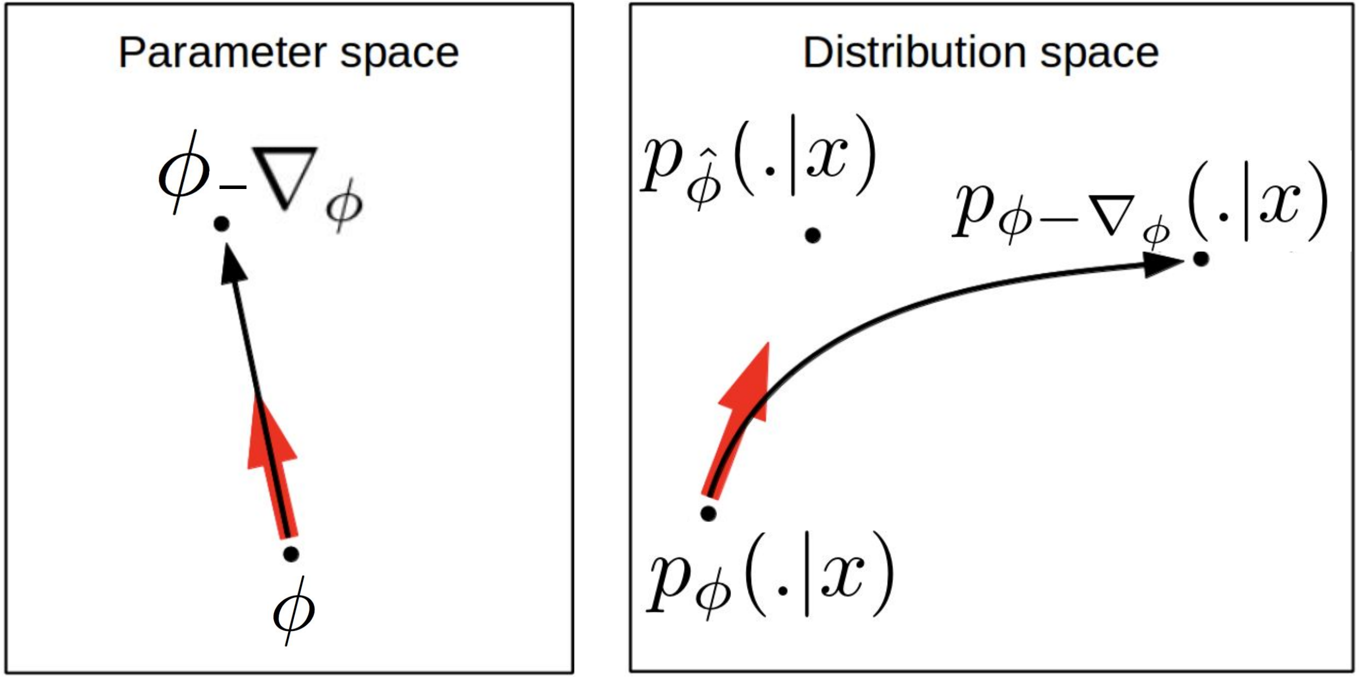


ϕ

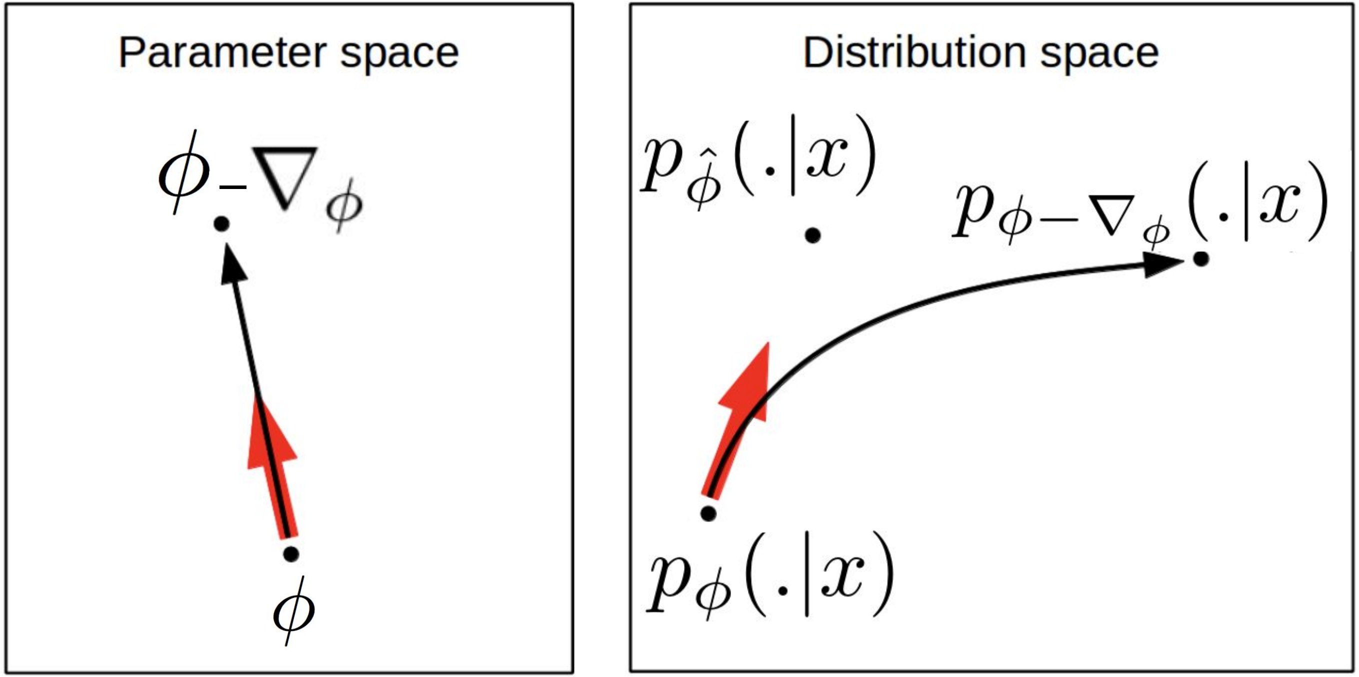


θ

Synchronizing syntax and semantics learning

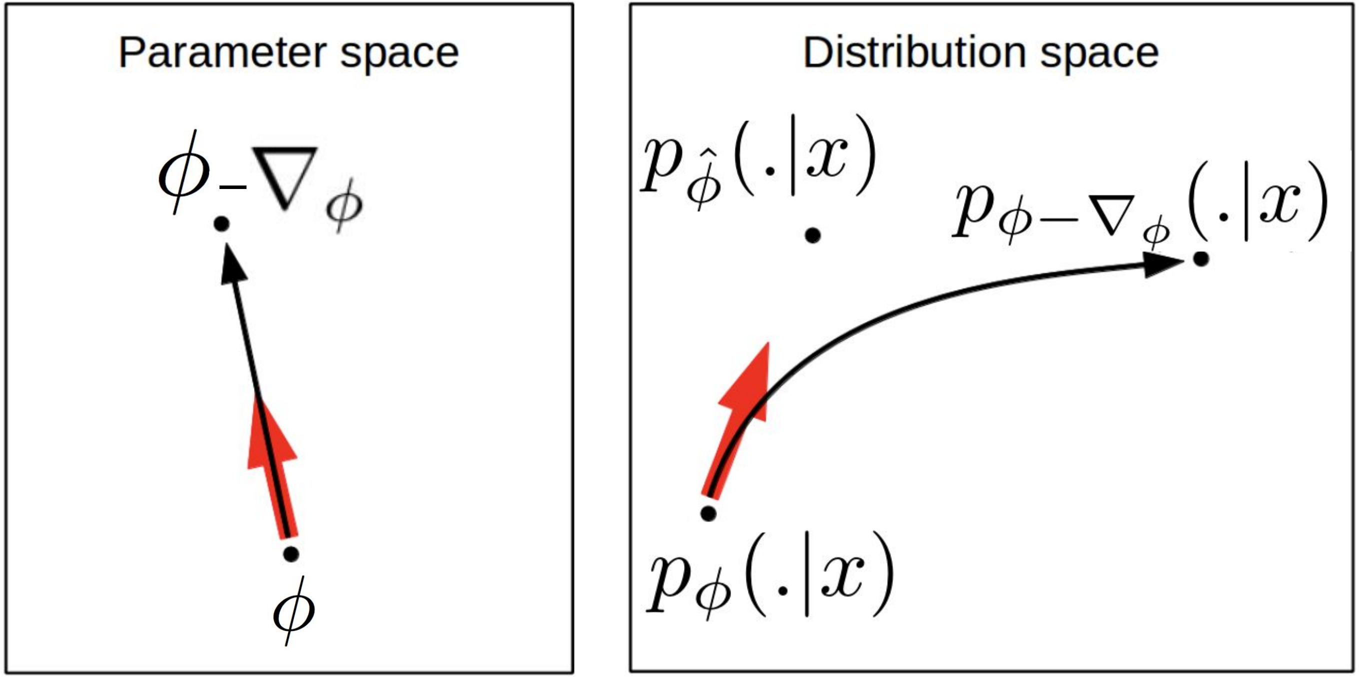


Synchronizing syntax and semantics learning



$$\frac{p_\phi(t|x)}{p_{\phi_{\text{old}}}(t|x)} \in [1-\epsilon; 1+\epsilon]$$

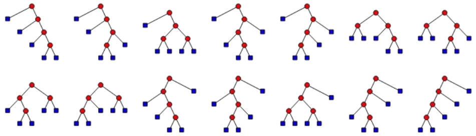
Synchronizing syntax and semantics learning



Proximal Policy Optimization (PPO) of Schulman et al. (2017)

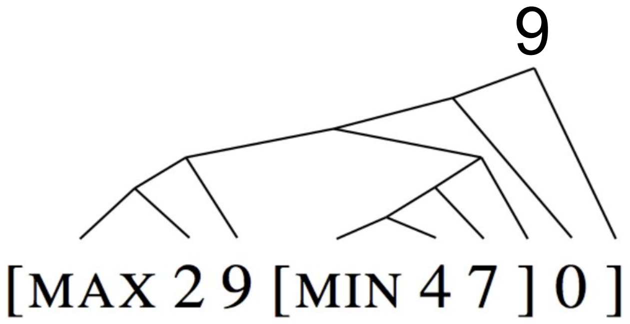
Optimization challenges

- High variance in the estimate of a parser's gradient ∇_{φ} is addressed by using **self-critical training** (SCT) baseline of Rennie et al. (2017).

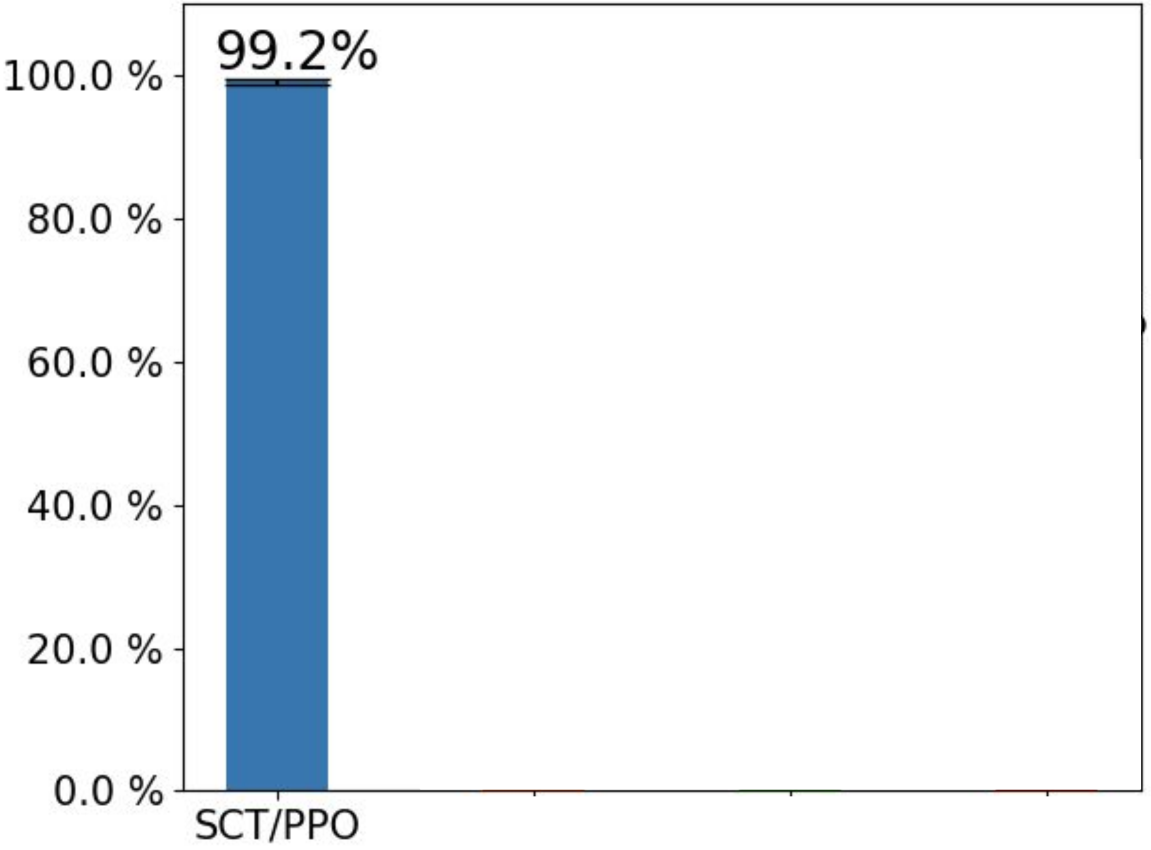


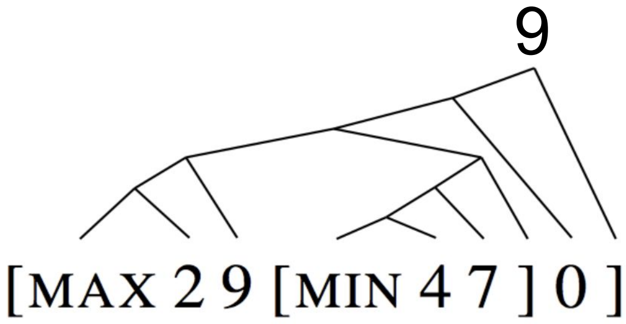
- Learning paces of a parser φ and a compositional function θ is levelled off by controlling parser's updates using **Proximal Policy Optimization** (PPO) of Schulman et al. (2017).

ListOps results

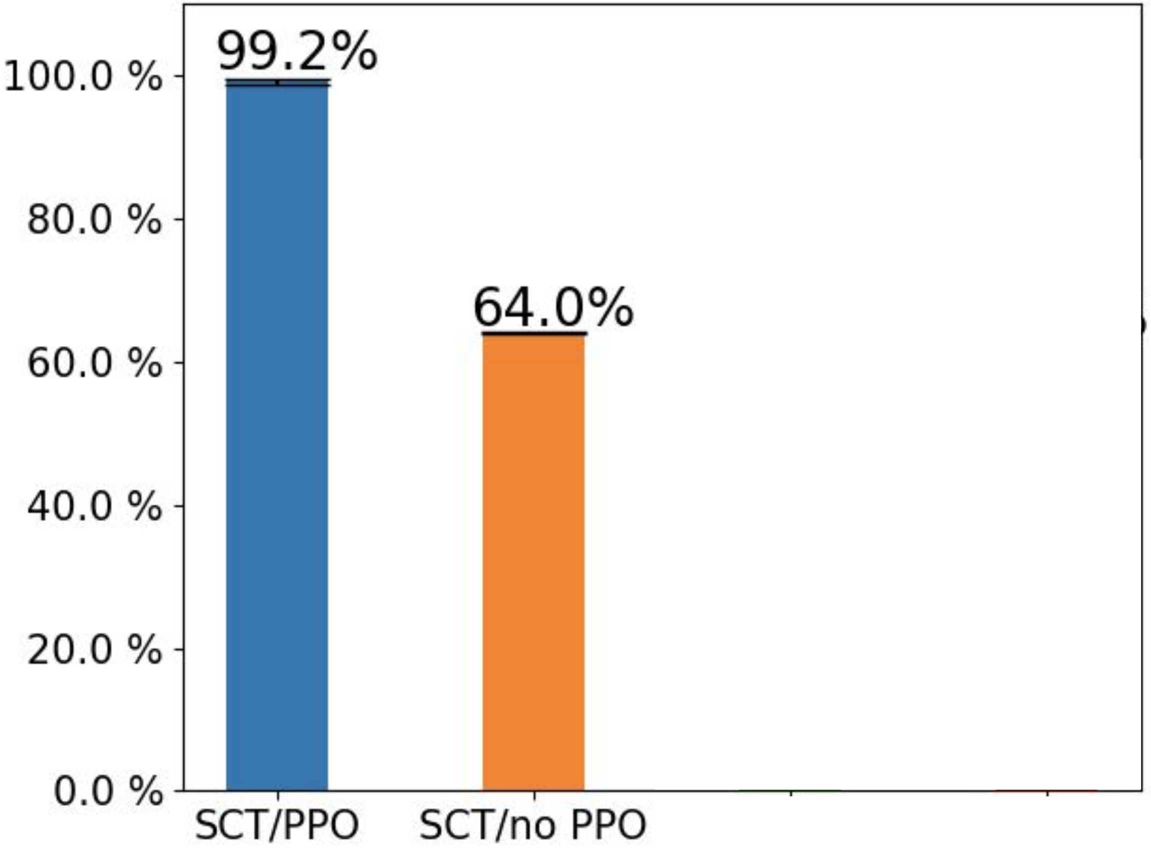


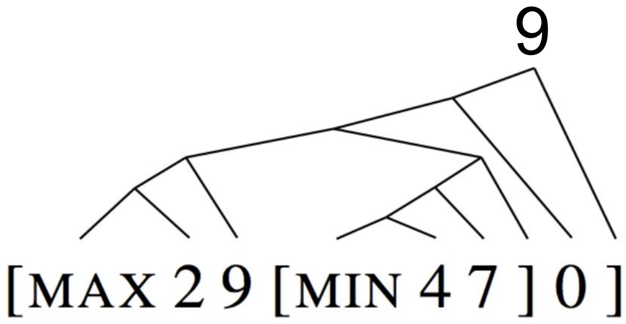
ListOps results



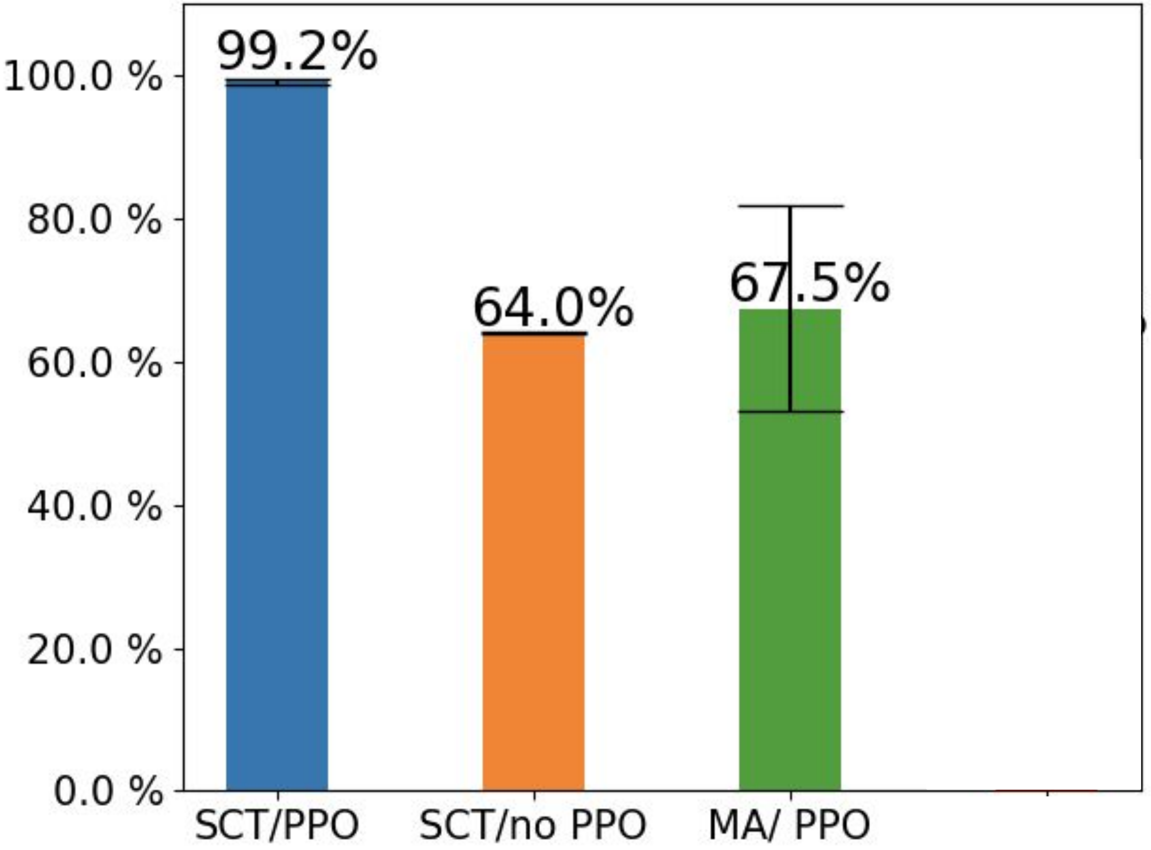


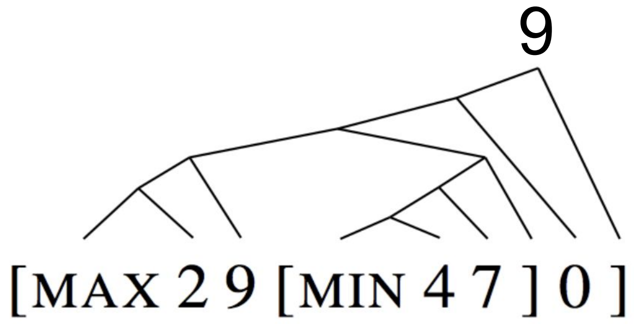
ListOps results



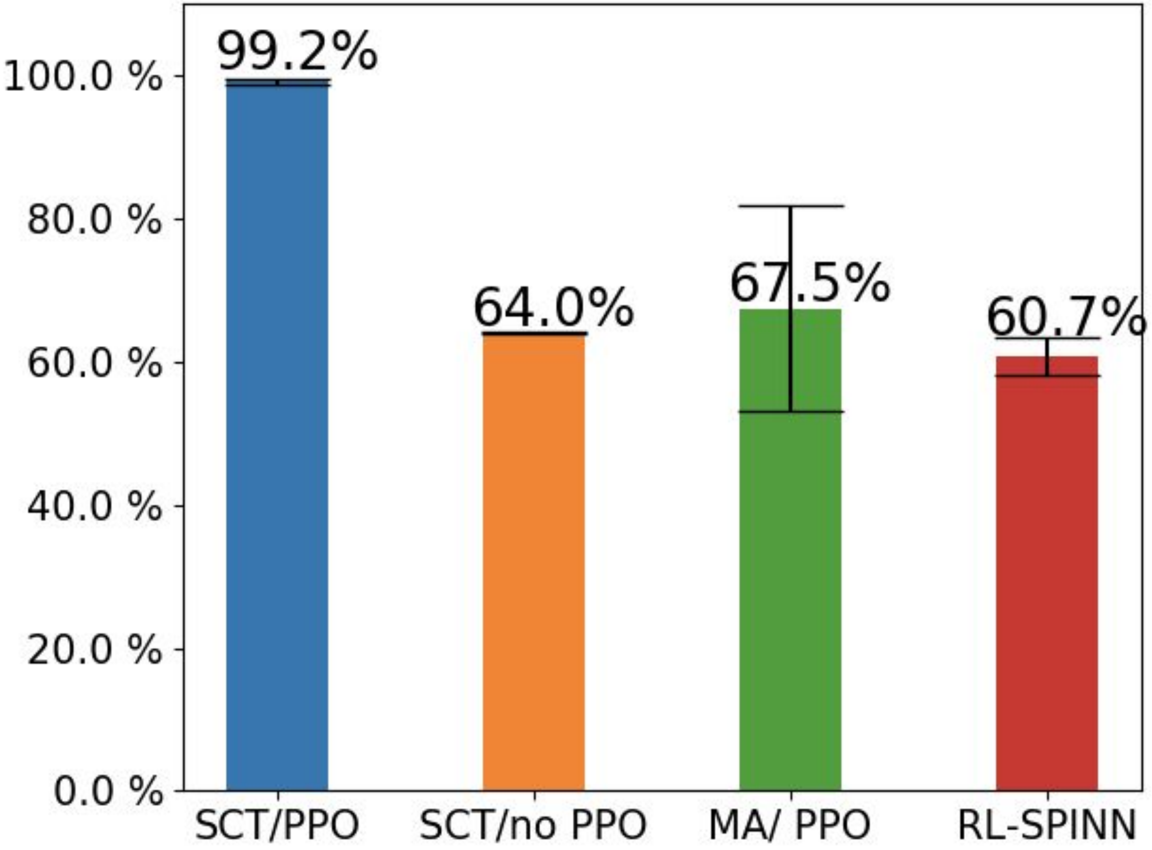


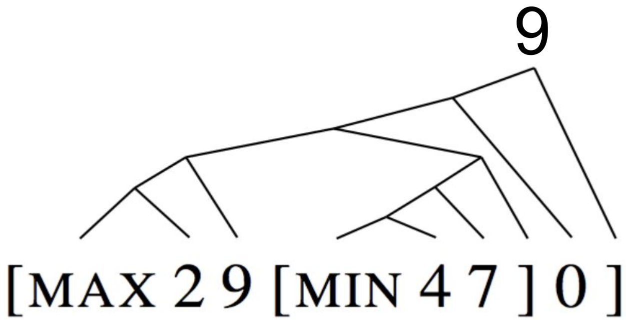
ListOps results



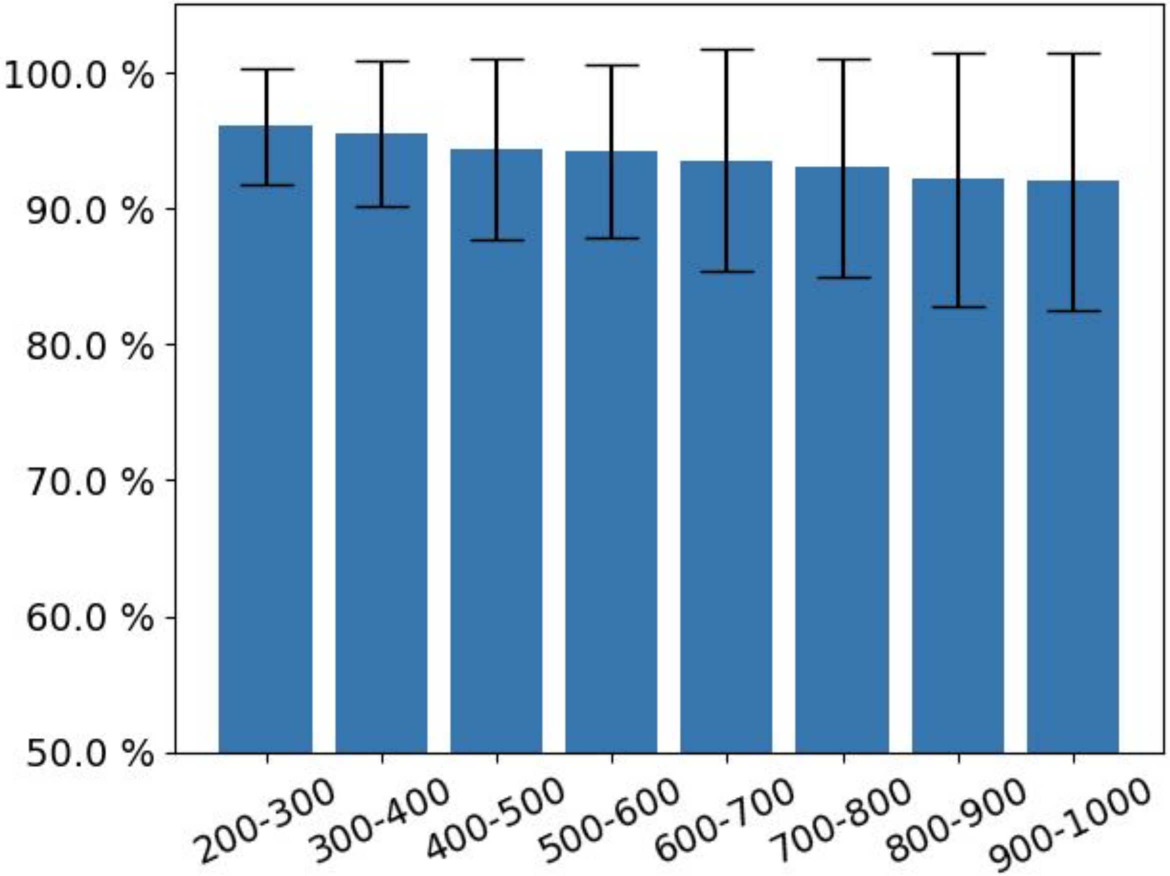


ListOps results

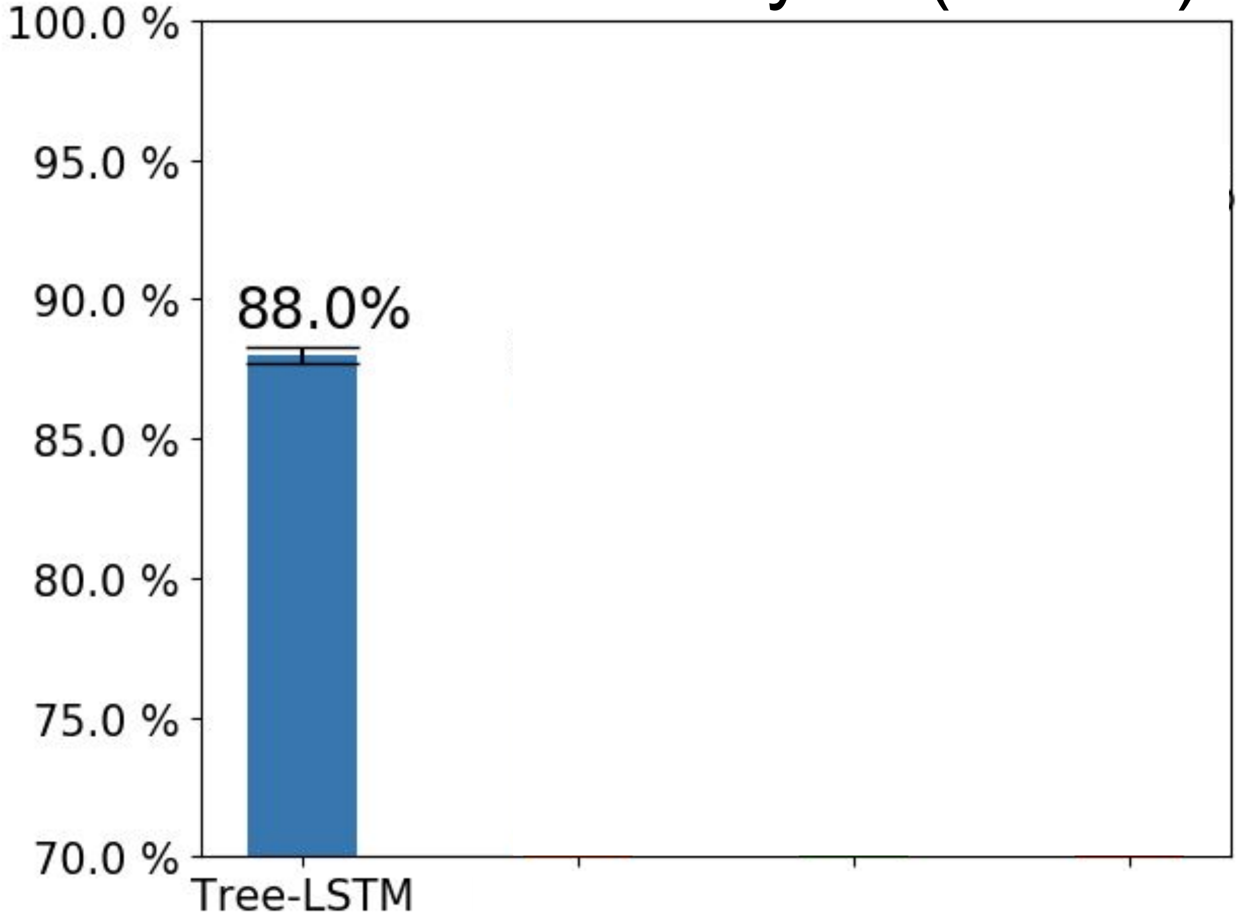




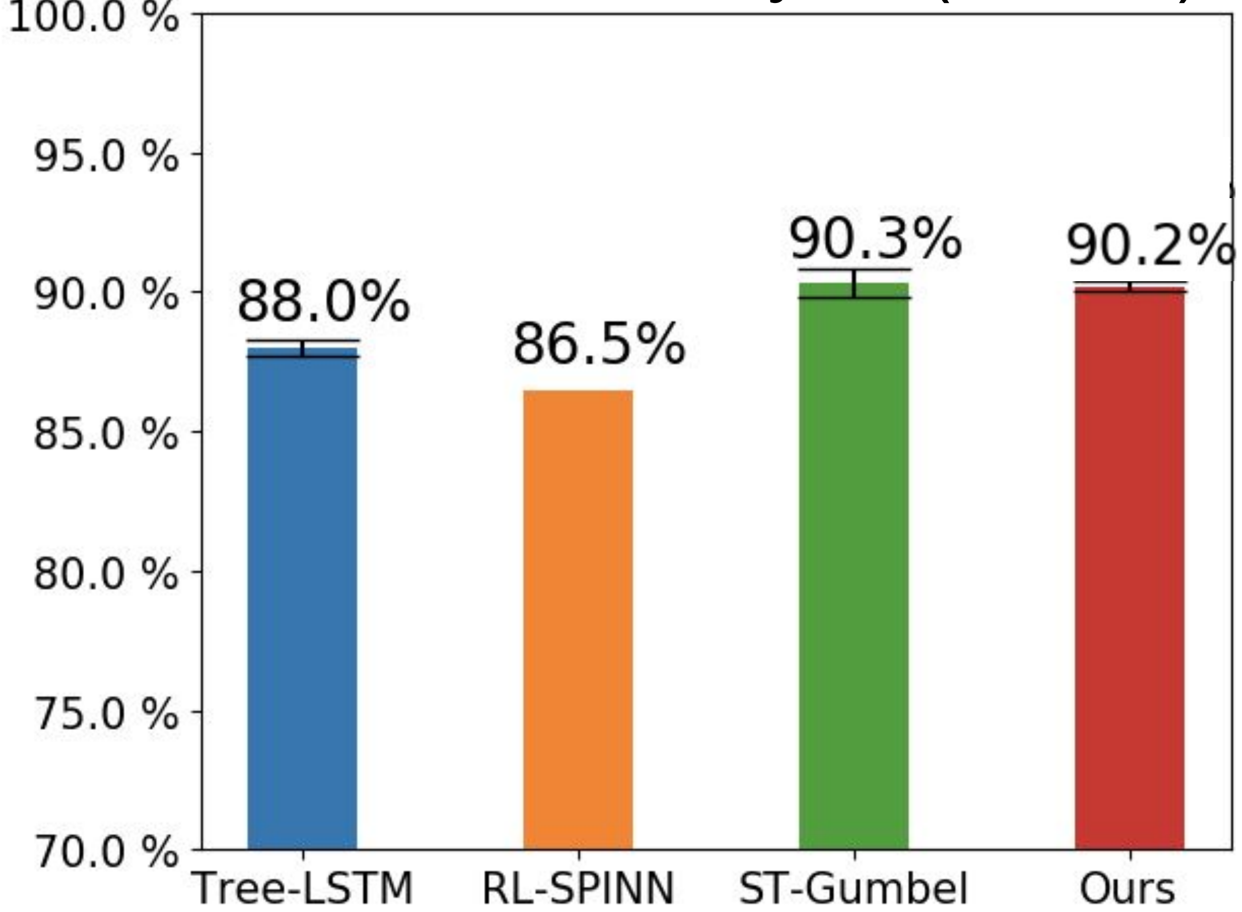
Extrapolation



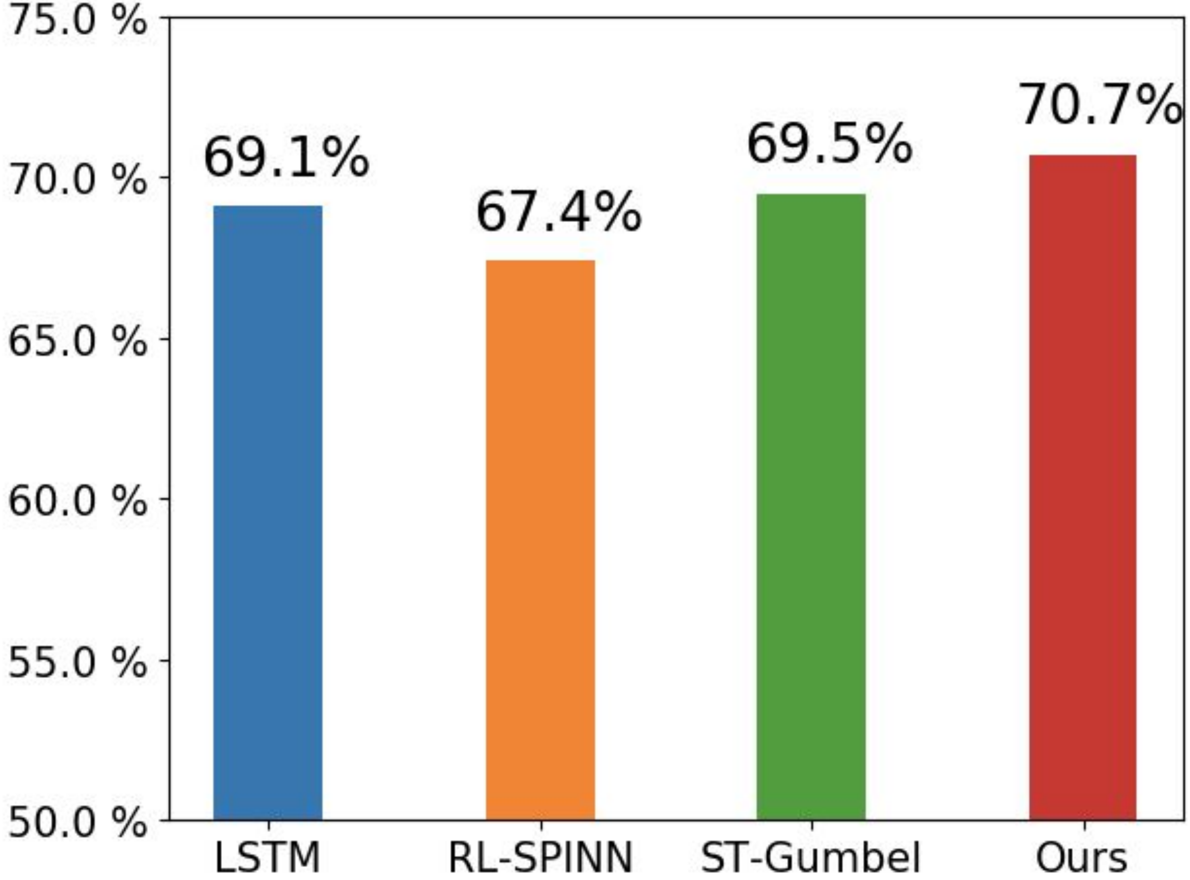
Sentiment Analysis (SST-2)



Sentiment Analysis (SST-2)



Natural language inference (MultiNLI)



Time and Space complexities

Method	Time complexity	Space complexity	ListOps
RL-SPINN: Yogatama et al., 2016	$O(nd^2)$	$O(nd^2)$	✗
Soft-CYK: Maillard et al., 2017	$O(n^3d+n^2d^2)$	$O(n^3d)$	✗
Gumbel Tree-LSTM: Choi et al., 2018	$O(n^2d+nd^2)$	$O(n^2d)$	✗
Ours	$O(Knd^2)$	$O(nd^2)$	✓

n – sentence length
d – tree-LSTM dimensionality
K – number of updates in PPO

Conclusions

- The *separation* between syntax and semantics allows **coordination** between optimisation schemes for each module.
- Self-critical training **mitigates credit assignment** problem by *distinguishing* “hard” and “easy” to solve datapoints.
- The model **can recover** a simple context-free grammar of mathematical expressions.
- The model **performs competitively** on several real natural language tasks.

github.com/facebookresearch/latent-treelstm