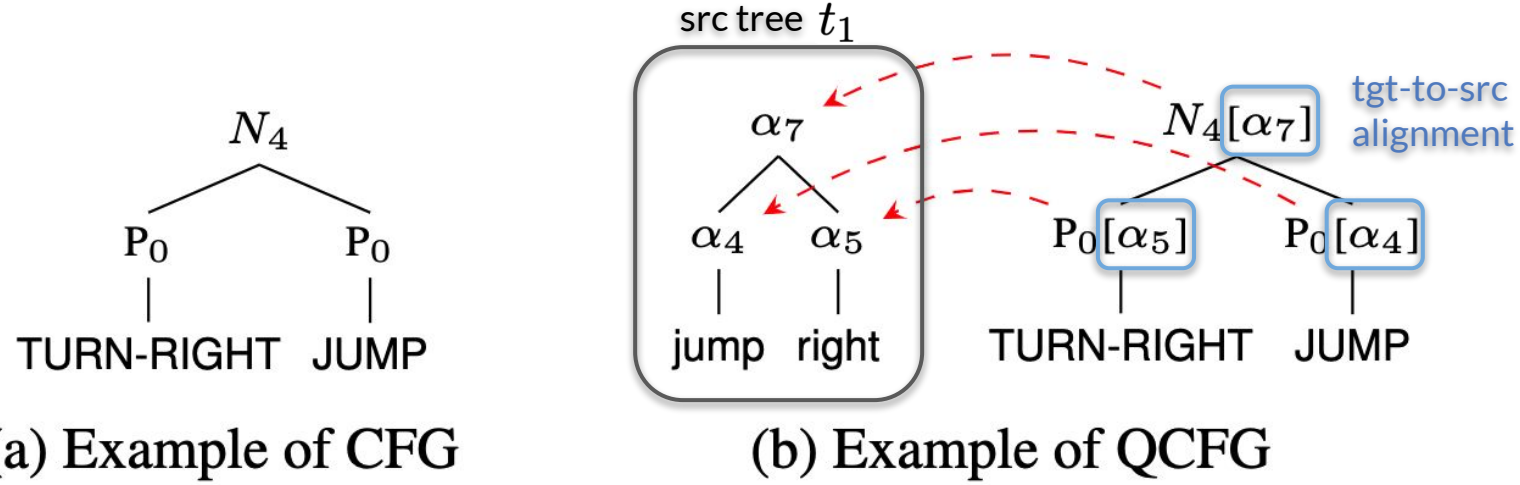
Improving Grammar-based Sequence-to-Sequence Modeling with Decomposition and Constraints

Chao Lou, Kewei Tu

School of Information Science and Technology, ShanghaiTech University
Shanghai Engineering Research Center of Intelligent Vision and Imaging

Background: Quasi-synchronous Context-free Grammar



(a) Example of CFG

(b) Example of QCFG

Background: Quasi-synchronous Context-free Grammar

Rules in QCFG:

$$E \rightarrow A[\alpha_i] \qquad \text{where } A \in \mathcal{N},\ \alpha_i \in t_1,$$

binary rules

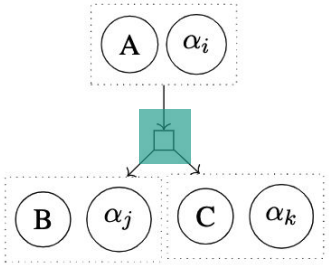
$$A[\alpha_i] \rightarrow B[\alpha_j]C[\alpha_k] \qquad \text{where } A \in \mathcal{N},\ B, C \in \mathcal{N} \cup \mathcal{P},\ \alpha_i, \alpha_j, \alpha_k \in t_1,$$

$$D[\alpha_i] \rightarrow w \qquad \text{where } A \in \mathcal{P},\ \alpha_i \in t_1,\ w \in \Sigma$$

- $|\mathcal{N}|(|\mathcal{N}| + |\mathcal{P}|)^2 S^3$ binary rules in total, *where S is the len of src sequences.*
- Cannot scale up ($|\mathcal{N}||\mathcal{P}|$)
- Not support long sequences (S)

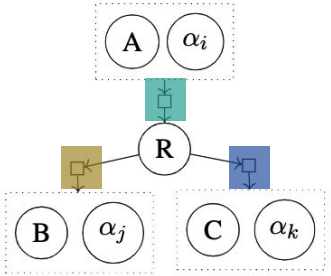
Low-rank Models

We propose the efficient model (E model) and the expressive model (P model) by decomposing binary rules.



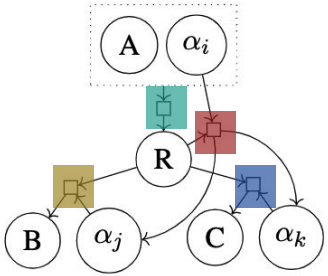
(a) Vanilla Neural QCFG

$$p(A[\alpha_i] \rightarrow B[\alpha_j]C[\alpha_k])$$



(b) E model

$$p(A[\alpha_i] \rightarrow B[\alpha_j]C[\alpha_k]) = \sum_R p(A[\alpha_i] \rightarrow R) \times p(R \rightarrow B[\alpha_j]) \times p(R \rightarrow C[\alpha_k])$$

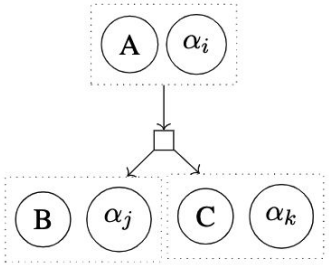


(c) P model

$$p(A[\alpha_i] \rightarrow B[\alpha_j]C[\alpha_k]) = \sum_R p(A[\alpha_i] \rightarrow R) \times p(R, \alpha_i \rightarrow \alpha_j, \alpha_k) \times p(R, \alpha_j \rightarrow B) \times p(R, \alpha_k \rightarrow C)$$

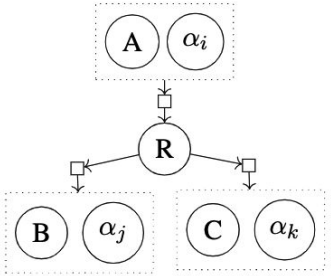
Low-rank Models

We propose the efficient model (E model) and the expressive model (P model) by decomposing binary rules.



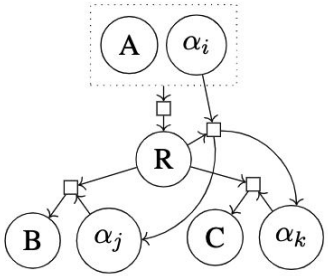
(a) Vanilla Neural QCFG

- **High** time and space complexity
- Support **short** sentences and **a small number** of nonterminals
- Support **all** constraints



(b) E model

- **Much lower** time and space complexity
- Support **much longer** sentences and **much more** nonterminals
- Support **few** constraints

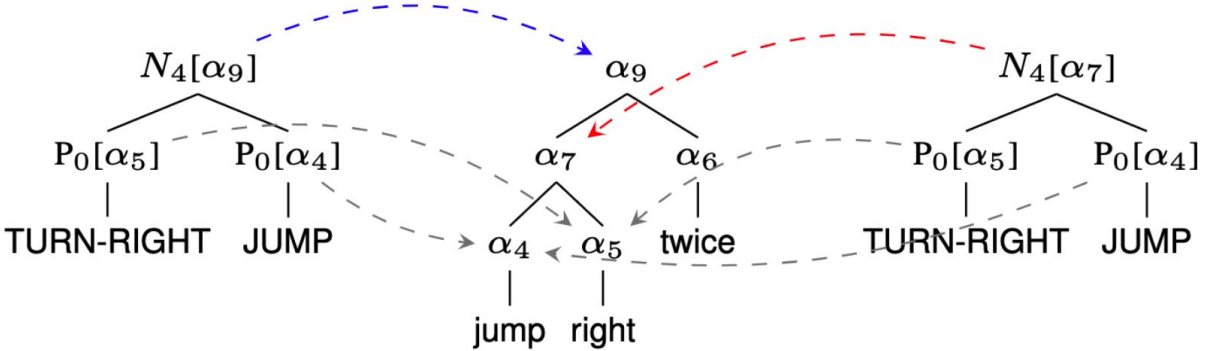


(c) P model

- **Lower** time and space complexity
- Support **longer** sentences and **more** nonterminals
- Support **all** constraints

Constraint I: Soft Tree Hierarchy Constraint

Rules with **smaller distances** between the src parent node and its children are more plausible.



Case 1: $N_4[\alpha_9] \rightarrow P_0[\alpha_5]P_0[\alpha_4]$

The distance between α_9 and α_5, α_4 is 2.

Case 2: $N_4[\alpha_7] \rightarrow P_0[\alpha_5]P_0[\alpha_4]$

The distance between α_7 and α_5, α_4 is 1.

Constraint I: Soft Tree Hierarchy Constraint

Rules with **smaller distances** between the src parent node and its children are more plausible.

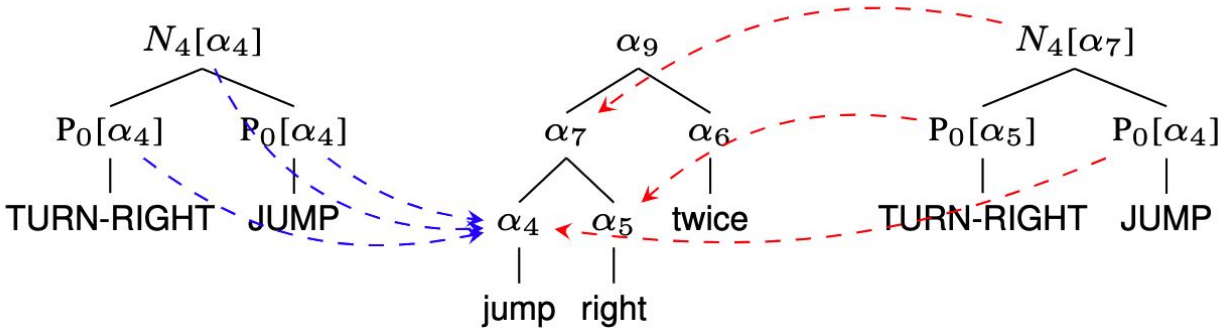
Implementation:

- Define the distance of a rule $d(r) = \max(d(\alpha_i, \alpha_j), d(\alpha_i, \alpha_k))$
- Define the reward function $\zeta(d(r)) \coloneqq d(r)e^{-d(r)}$
  and the reward of a tree $\zeta(t_2) = \prod_{r \in t_2} \zeta(d(r))$
- Optimize the expected rewards with a maximum entropy regularizer

$$\log \sum_{t_2 \in \mathcal{T}(s_2)} p_\theta(t_2|t_1)\zeta(t_2) + \tau \mathbb{H}\left(p_\theta(t_2|t_1, s_2)\right)$$

Constraint II: Coverage Constraint

Encourage the translation of all source tree nodes by setting **an upper limit** on the number of alignments per node, such that more source tree nodes will be aligned.



Case 1: max num. of alignments = 3
(α_4)

Case 2: max num. of alignments = 1
($\alpha_4\alpha_5\alpha_7$)

Constraint II: Coverage Constraint

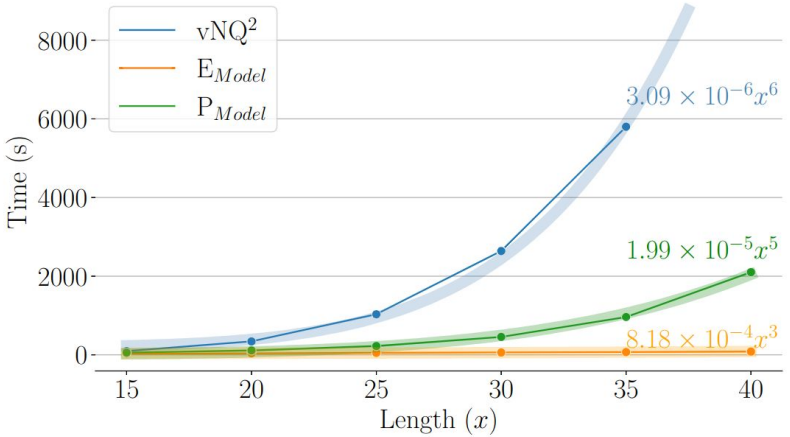
Encourage the translation of all source tree nodes by setting **an upper limit** on the number of alignments per node, such that more source tree nodes will be aligned.

Implementation:

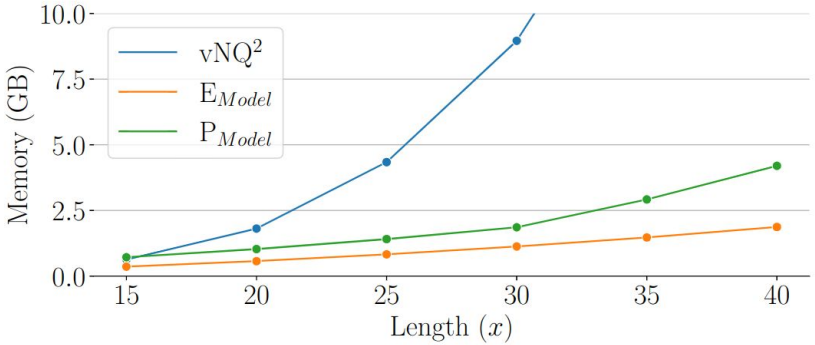
- Idea: posterior regularization (PR)
- Define the constraint set $\mathcal{Q} = \{q(t_2)|\mathbb{E}_{q(t)}\phi(t) \leq \xi\}$ (i.e., any distribution satisfying the upper limit)
- Optimize the PR objective

$$\mathbb{E}_{t_1}\left(\log p_\theta(s_2|t_1) + \gamma \min_{q\in\mathcal{Q}} \mathbb{KL}(q(t_2)||p_\theta(t_2|t_1, s_2))\right)$$

Efficiency: E model > P model > vanilla Neural QCFG



The one-epoch **training time** with different length (x = S = T)



Memory usage for training with batch size 1 on synthetic datasets with different length (x = S = T)

Experiments

Accuracy on SCAN

- *vNQ*[1] and *the P model* learn almost perfectly with constraints
- E model fails due to a lack of constraints

Approach	Simple	Jump	A. Right	Length
vNQ[1]	96.9	96.8	98.7	95.7
E_{Model}	9.01	-	1.2	-
P_{Model}	95.27	97.08	97.63	91.72

Experiments

BLEU scores for tasks (ATP, AEM, VEM) from StylePTB and the En-Fr machine translation.

- Our low-rank models outperforms *vNQ*[1]
- When using the newly proposed constraints, we can achieve better results

Approach	*nil*	+H[1]	+H[2]	+S	+C
Active to passive (ATP)					
vNQ[1]	–	66.2	–	–	–
vNQ[2]	71.42	71.56	–	71.62	73.86
E_{Model}	73.48	×	×	×	74.25
P_{Model}	75.06	69.88	–	73.11	75.44
Adjective Emphasise (AEM)					
vNQ[1]	–	31.6	–	–	–
vNQ[2]	28.82	31.52	–	36.77	30.81
E_{Model}	28.33	×	×	×	28.67
P_{Model}	31.81	29.14	–	35.91	30.12
Verb Emphasise (VEM)					
vNQ[1]	–	31.9	–	–	–
vNQ[2]	26.09	29.64	–	30.50	28.50
E_{Model}	25.21	×	×	×	24.67
P_{Model}	27.43	24.77	–	26.81	30.66
En-Fr machine translation					
vNQ[1]	–	–	26.8	–	–
vNQ[2]	28.63	–	29.10	30.45	31.87
E_{Model}	28.93	×	×	×	29.33
P_{Model}	29.27	–	29.76	30.51	29.69

Thank you