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

① Overview

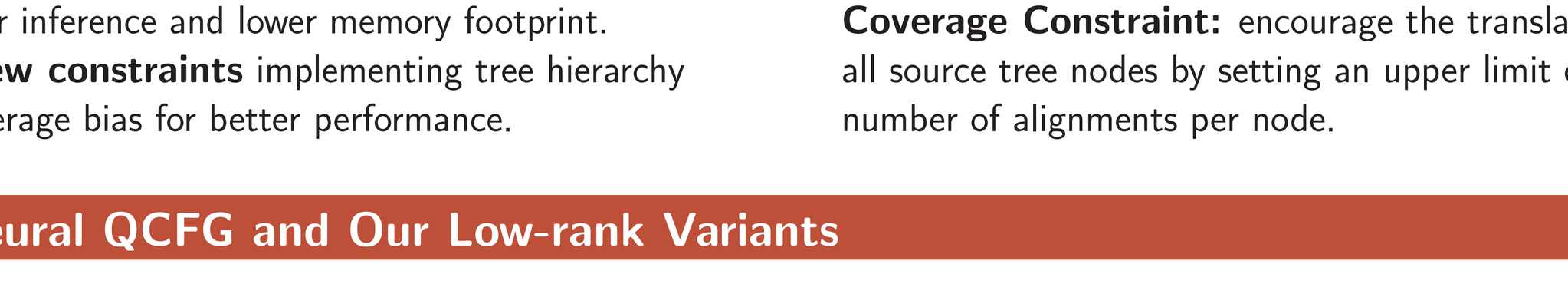
Background: **Neural QCFG** models the generation of target sequences and source-target node alignments.
Our work: **Two low-rank variants** of Neural QCFG for faster inference and lower memory footprint.
Two new constraints implementing tree hierarchy and coverage bias for better performance.

③ Constraints

Soft Tree Hierarchy Constraint: rules with smaller distances between the src parent node (i.e., α_i) and its children (i.e., α_j and α_k) are more plausible.
Coverage Constraint: encourage the translation of all source tree nodes by setting an upper limit on the number of alignments per node.

② Neural QCFG and Our Low-rank Variants

Neural QCFG has binary rules in the form of $A[\alpha_i] \to B[\alpha_j]C[\alpha_k]$. N^3S^3 rules in total, where N is the num. of nonterminals and S is the src sequence length. Difficult to scale up.
E model decomposes rules into $A[\alpha_i] \to r,\ r \to B[\alpha_j],\ r \to C[\alpha_k]$. $3NSR$ rules in total, where R is the num. of possible value of r. Support very efficient rank-space dynamic programming, but support few constraints.
P model decomposes rules into $A[\alpha_i] \to r,\ r\alpha_i \to \alpha_j\alpha_k,\ r\alpha_j \to B,\ r\alpha_k \to C$. $3NSR + RS^3$ rules in total. The relation among $\alpha_i, \alpha_j, \alpha_k$ is reserved, such that all constraints are supported.



(a) Example of QCFG (b) Vanilla Neural QCFG (c) E model (d) P model

Figure 1: (a): An example of QCFG. (b)-(d): Extended factor graph notation of decomposed binary rules. α means a source node, R is the rank variable and other capitalized letters mean target-side nonterminals. Red dashed lines are alignments. Each square □ represents a factor. Arrows indicate conditional probabilities.

④ Speed and memory benchmark

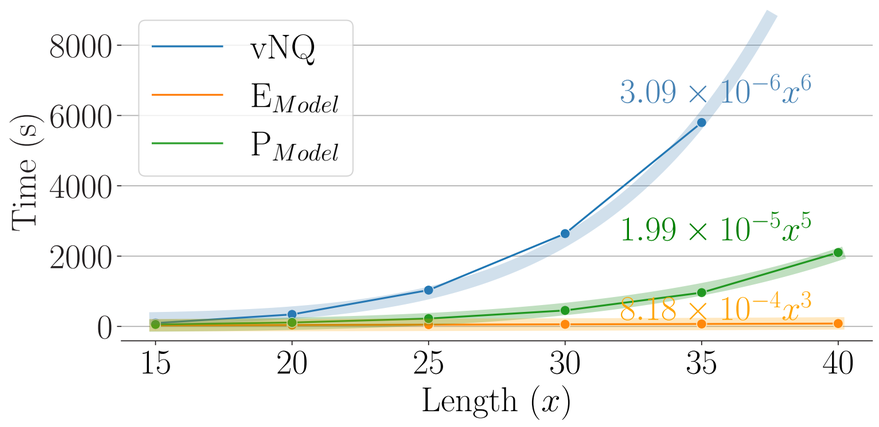


Figure 2: The one-epoch training time with different length ($x = S = T$).

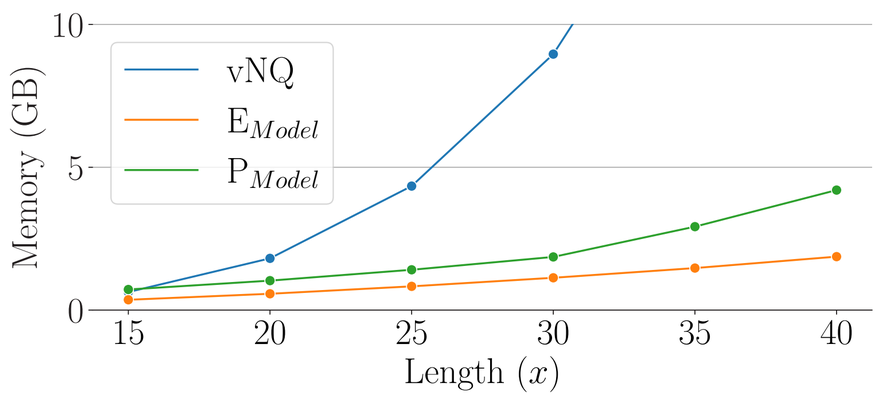


Figure 3: Memory usage for training with batch size 1 on synthetic datasets with different length ($x = S = T$).

⑤ Experiments

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

Table 1: Accuracy on the SCAN datasets.

Approach	*nil*	+H[1]	+H[2]	+S	+C
Active to passive (ATP)					
vNQ	71.42	71.56	–	71.62	73.86
E_{Model}	73.48	×	×	×	74.25
P_{Model}	75.06	69.88	–	73.11	75.44
En-Fr machine translation					
vNQ	28.63	–	29.10	30.45	31.87
E_{Model}	28.93	×	×	×	29.33
P_{Model}	29.27	–	29.76	30.51	29.69

Table 2: BLEU-4 for the ATP task from StylePTB (the top series) and BLEU for machine translation.