Sequence Modeling with Unconstrained Generation Order

Dmitrii Emelianenko, Elena Voita, Pavel Serdyukov | NeurIPS 2019

Yandex Research

TL;DR

- Sequence model that can **insert** tokens at an arbitrary position;
- Implicitly learn the most convenient decoding order from data;
- Superior results on several datasets for Machine Translation, Im2Latex and Image Captioning.

Learning to Insert

Main idea: generate a sequence by inserting elements at arbitrary positions
Hence, we generate a sequence of insertions: $\tau = (\tau_0, \tau_1, \tau_2, ..., \tau_T)$

Insertion: $\tau_t = (pos_t, token_t)$ insert $token_t$ at position $pos_t \in [0; t]$

Model: predict next insertion into partial output formed by previous inserts

$$p(\tau|X,\theta) = \prod_t p(\tau_t|X, \tilde{Y}(\tau_{0:t-1}), \theta)$$

X – input (e.g. source sequence in MT)

$\tilde{Y}(\tau_{0:t-1})$ – partial output sequence

Log-likelihood: (infeasible) $$L = \sum_{\{X,Y\}\in D} \log p(Y|X,\theta) = \sum_{\{X,Y\}\in D} \log \sum_{\tau \in T^*(Y)} \prod_t p(\tau_t|X, \tilde{Y}(\tau_{0:t-1}), \theta) \quad \textbf{(1)}$$

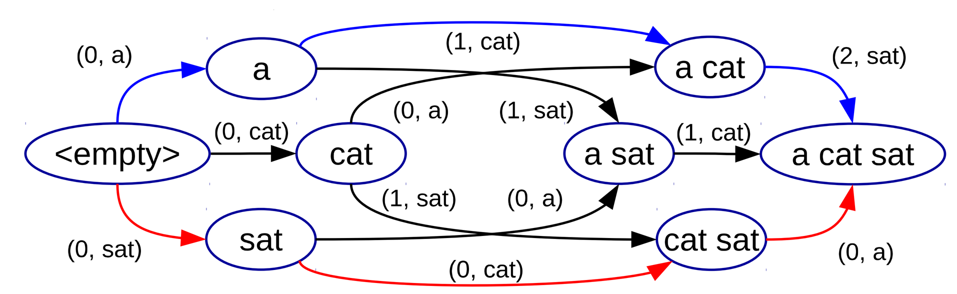
Lower bound: $$L \geq \sum_{\{X,Y\}\in D} E_{\tau\sim p(\tau|X,\tau\in T^*(Y),\theta)} \log \prod_t p(\tau_t|X, \tilde{Y}(\tau_{0:t-1}), \theta) \quad \textbf{(2)}$$

Training algorithm:

1) Sample mini-batch of sequences (X, Y);

2) Generate a trajectory τ that produces Y proportionally to model probabilities;

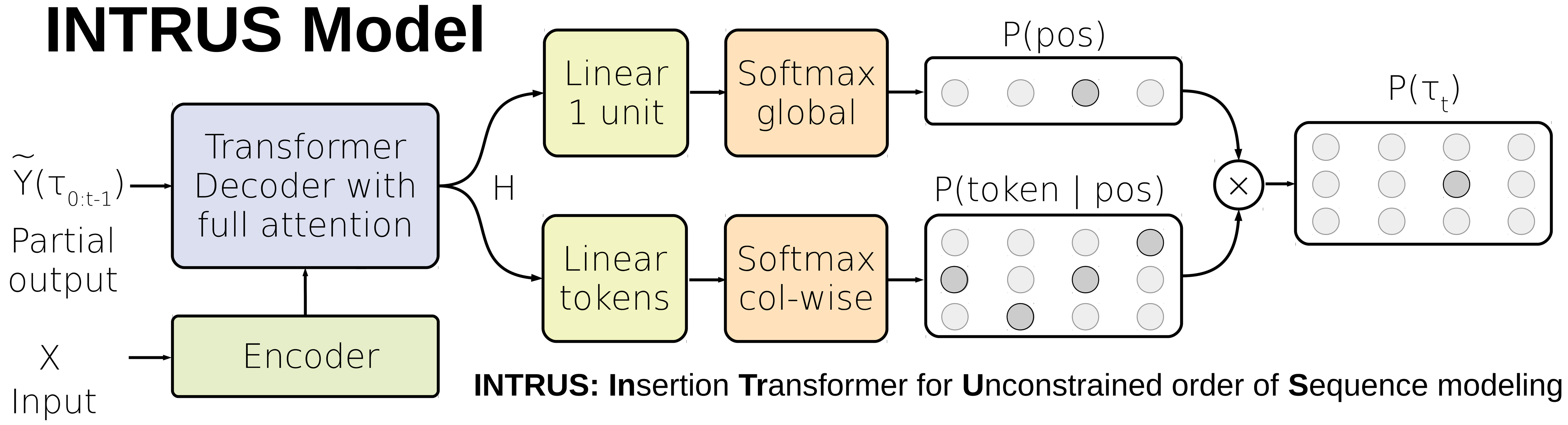
3) Maximize lower bound **(2)** by backprop;



All trajectories that produce "a cat sat". Left-to-right is blue and right-to-left is red.

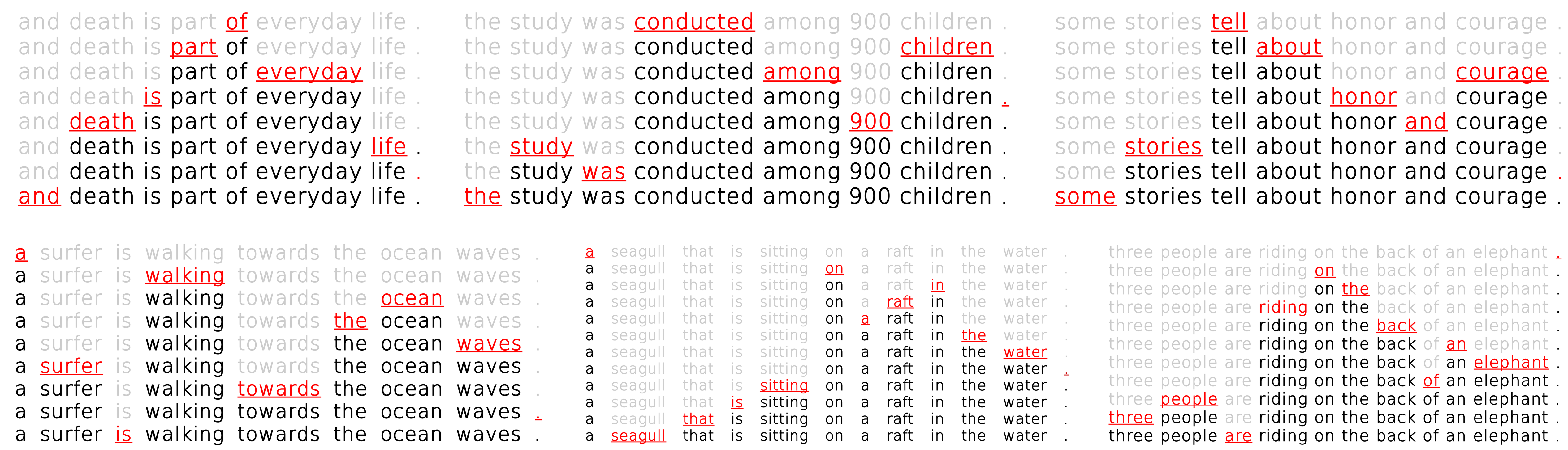
Pre-training: at step 2, sample generation order at random, ignoring model probabilities

INTRUS Model



INTRUS: Insertion **Tr**ansformer for **U**nconstrained order of **S**equence modeling

Learned Decoding Orders:



Experiments

Model	En-Ru	Ru-En	En-Ja	En-Ar	En-De	De-En	Im2Latex	MSCOCO	
	BLEU							BLEU	CIDEr
Left-to-right	31.6	35.3	47.9	**12.0**	28.04	**33.17**	89.5	18.0	56.1
Right-to-left	-	-	48.6	11.5	-	-	-	-	-
INTRUS	**33.2***	**36.4***	**50.3***	**12.2**	**28.36***	33.08	**90.3***	**25.6***	**81.0***

Tasks:

- Machine Translation WMT En-Ru, ASPEC En-Ja IWSLT En-De, En-Ar
- ImageToLatex-140K
- MSCOCO Image captioning

Ablation Analysis:

- Sampling lower bound is better than argmax
- Pre-training greatly affects performance

Training strategy	INTRUS	Argmax	Pretraining left-to-right	No pre-training	Only pretraining uniform	Only pretraining left-to-right	Baseline left-to-right
BLEU	27.5	26.6	26.3	27.1	24.6	25.5	25.8

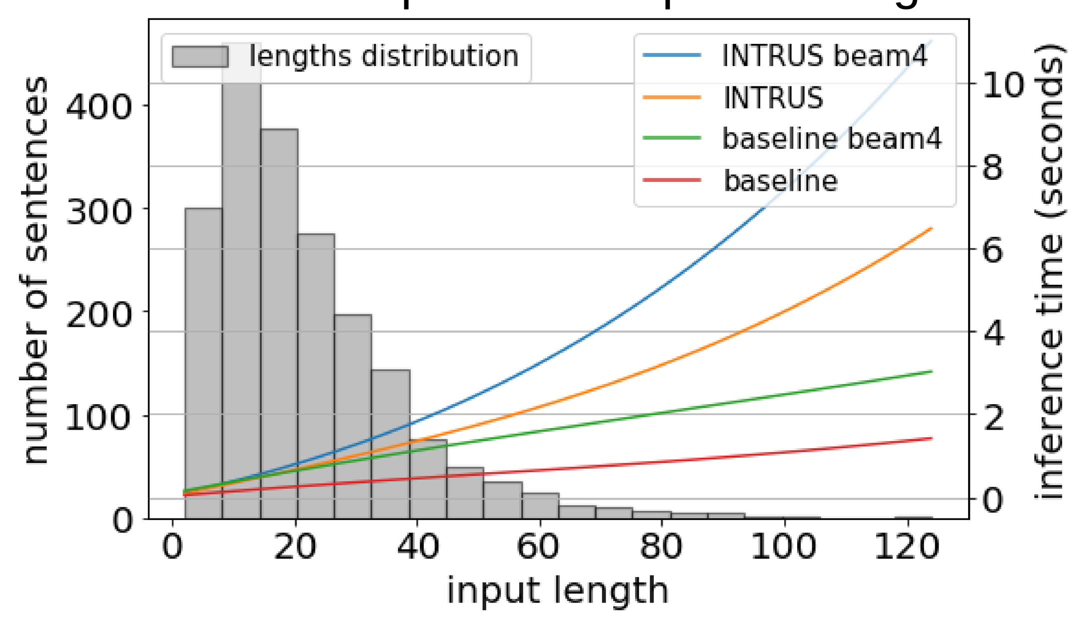
Ablation analysis on En-Ru translation task

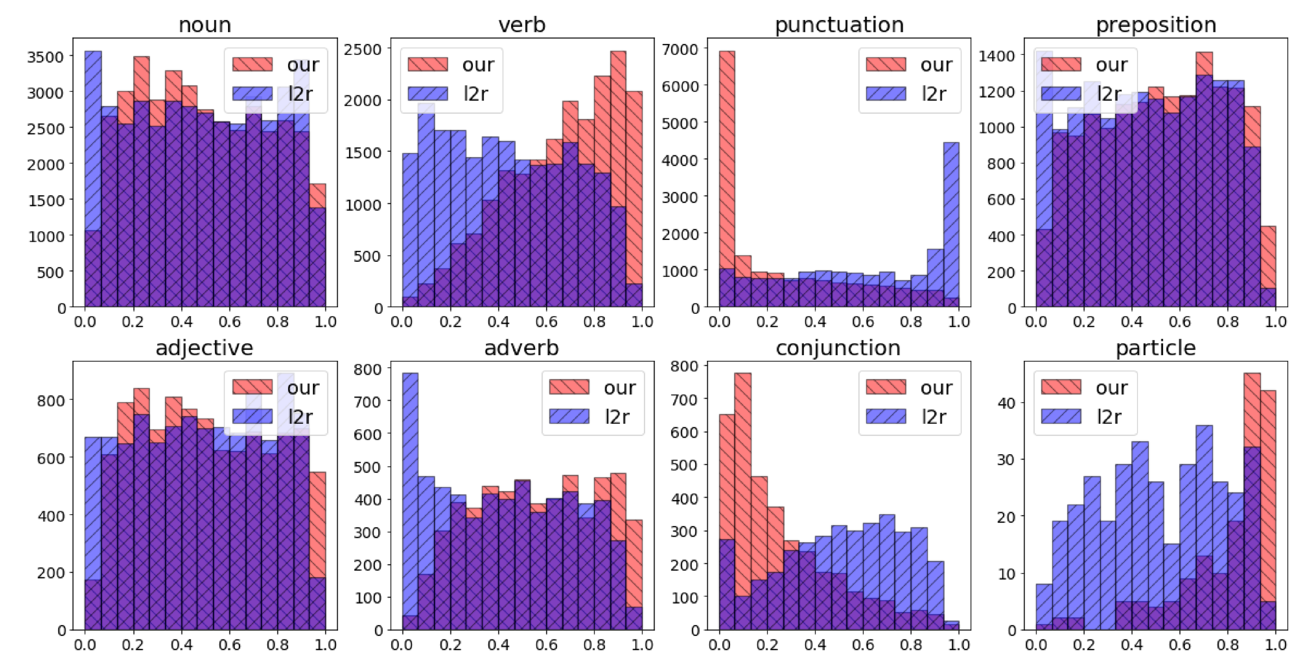
Analysis

Brief summary:

- Learned decoding order meaningfully differs from left-to-right (right);
- In general, the model learned to insert "easy" tokens first;
- INTRUS is ~50% slower than base Transformer for most MT tasks (below);

Inference speed vs sequence length





Step on which different POS tags appear in sequences generated by INTRUS (our) and Transformer (l2r)

Links:

Code: *https://github.com/TIXFeniks/neurips2019_intrus*

Yandex Research: *https://research.yandex.com*