Non-Autoregressive Neural Machine Translation

Speaker: 邵晨泽

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

Outline

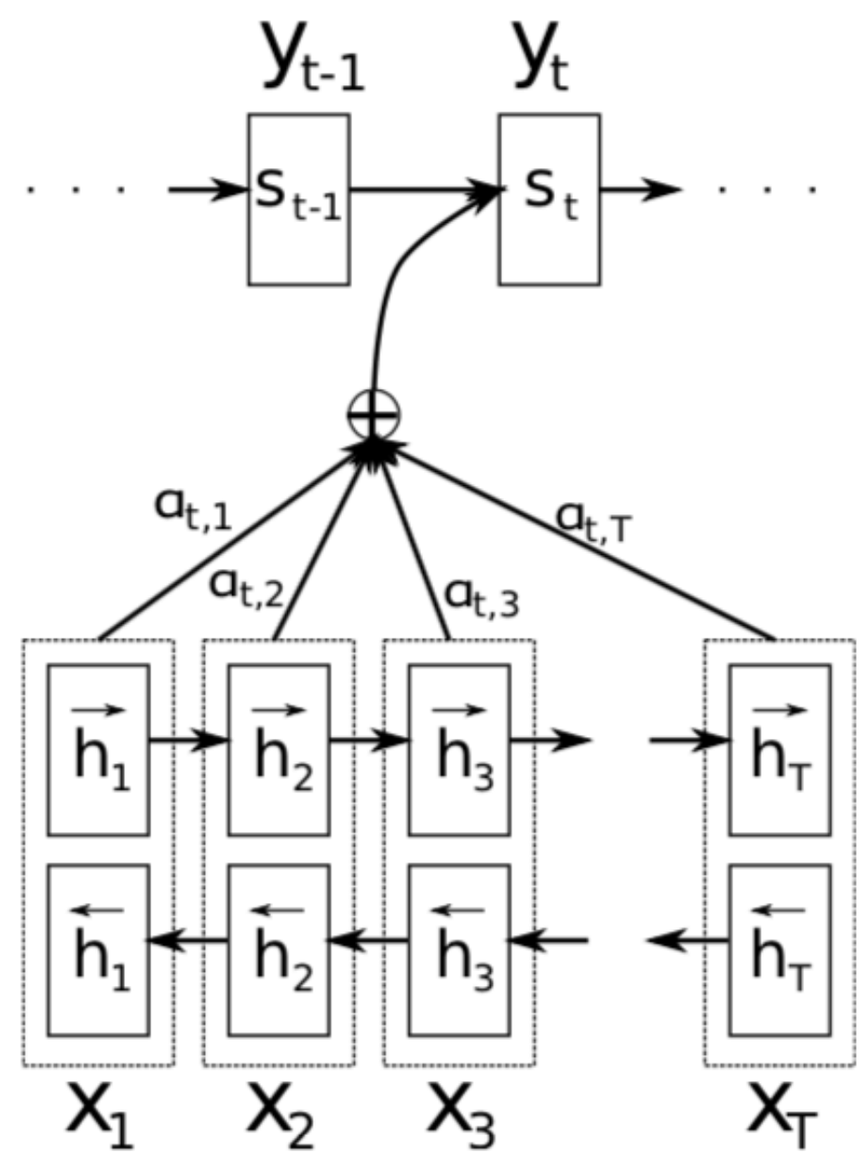
- **Introduction**
- Non-autoregressive Transformer (NAT)
- Progress in NAT
- Our work: Retrieving Sequential Information for Non-Autoregressive Neural Machine Translation

Autoregressive Mechanism

- Sequence prediction tasks: machine translation, image captioning, video description, etc.

- Sequence prediction models: RNN, CNN, Transformer, etc.

- Autoregressive mechanism: word-by-word generation, to predict the next word, previous predictions have to be fed to the model

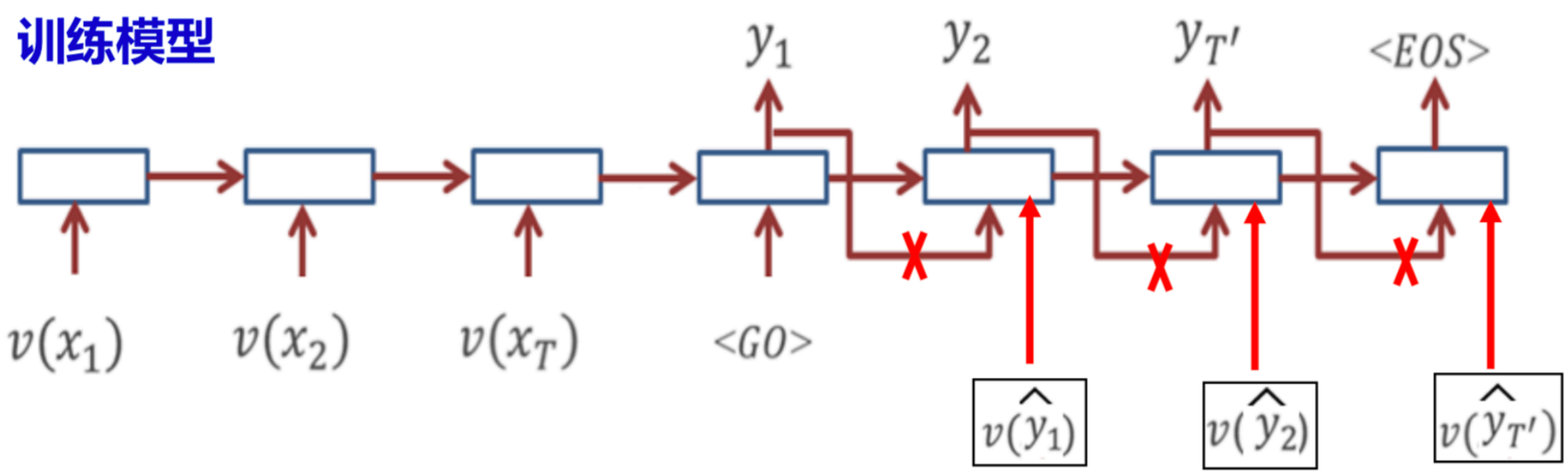


Autoregressive Mechanism

- Training: teacher forcing algorithm, the golden target word is fed to the next step.

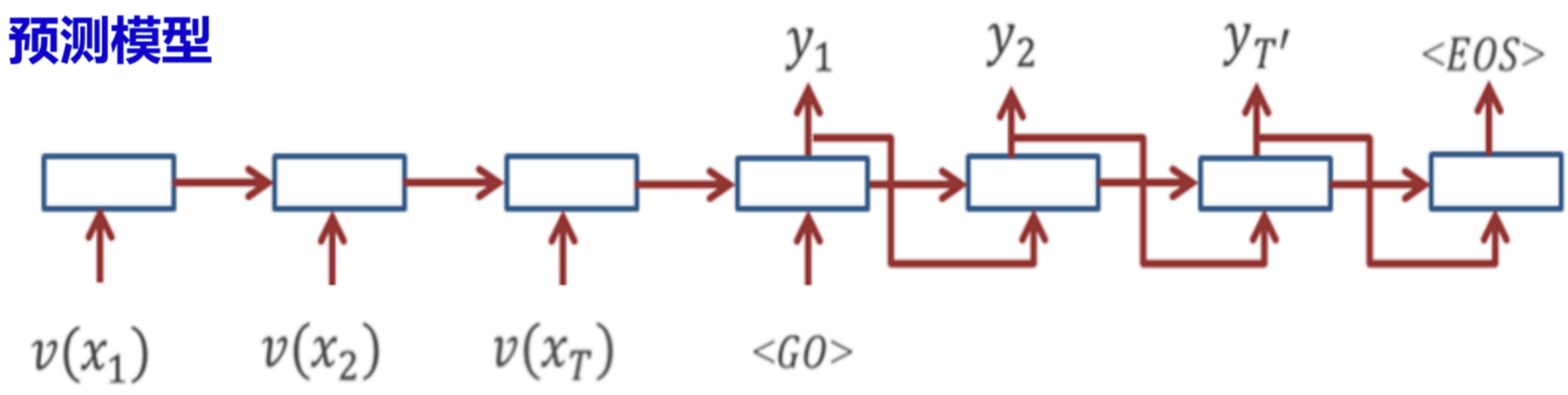
- Loss : word-level cross-entropy loss.

$$\mathcal{L}(\theta) = \sum_{m=1}^{M} \sum_{t=1}^{T} \log(p(y_t^m | \boldsymbol{y}_{<t}^m, \boldsymbol{X}^m, \theta)),$$



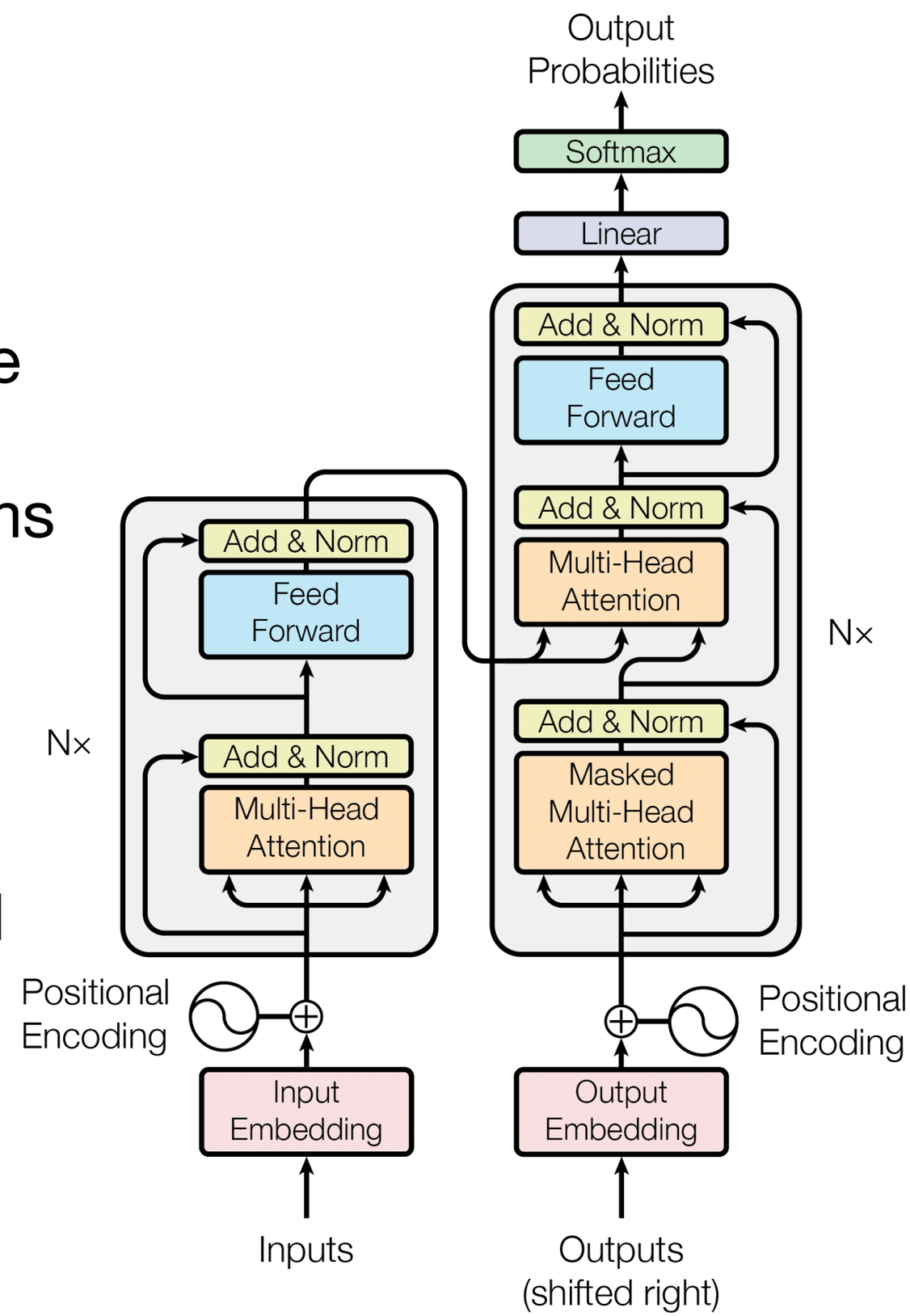
Autoregressive Mechanism

- Testing: word-by-word generation style, feed the model prediction to the next step
- Decoding algorithm: greedy search, beam search, etc.



Transformer

- Transformer: an autoregressive encoder-decoder architecture based on attention mechanisms
- Solve the problem of parallelization, fast in training
- Autoregressive: word-by-word generation, slow in decoding



Outline

- Introduction
- **Non-autoregressive Transformer (NAT)**
- Progress in NAT
- Our work: Retrieving Sequential Information for Non-Autoregressive Neural Machine Translation

Non-autoregressive Transformer

- NAT: Non-autoregressive Transformer、Non-autoregressive Translation

- Non-autoregressive mechanism: independent prediction of target words, generate the whole target sentence simultaneously

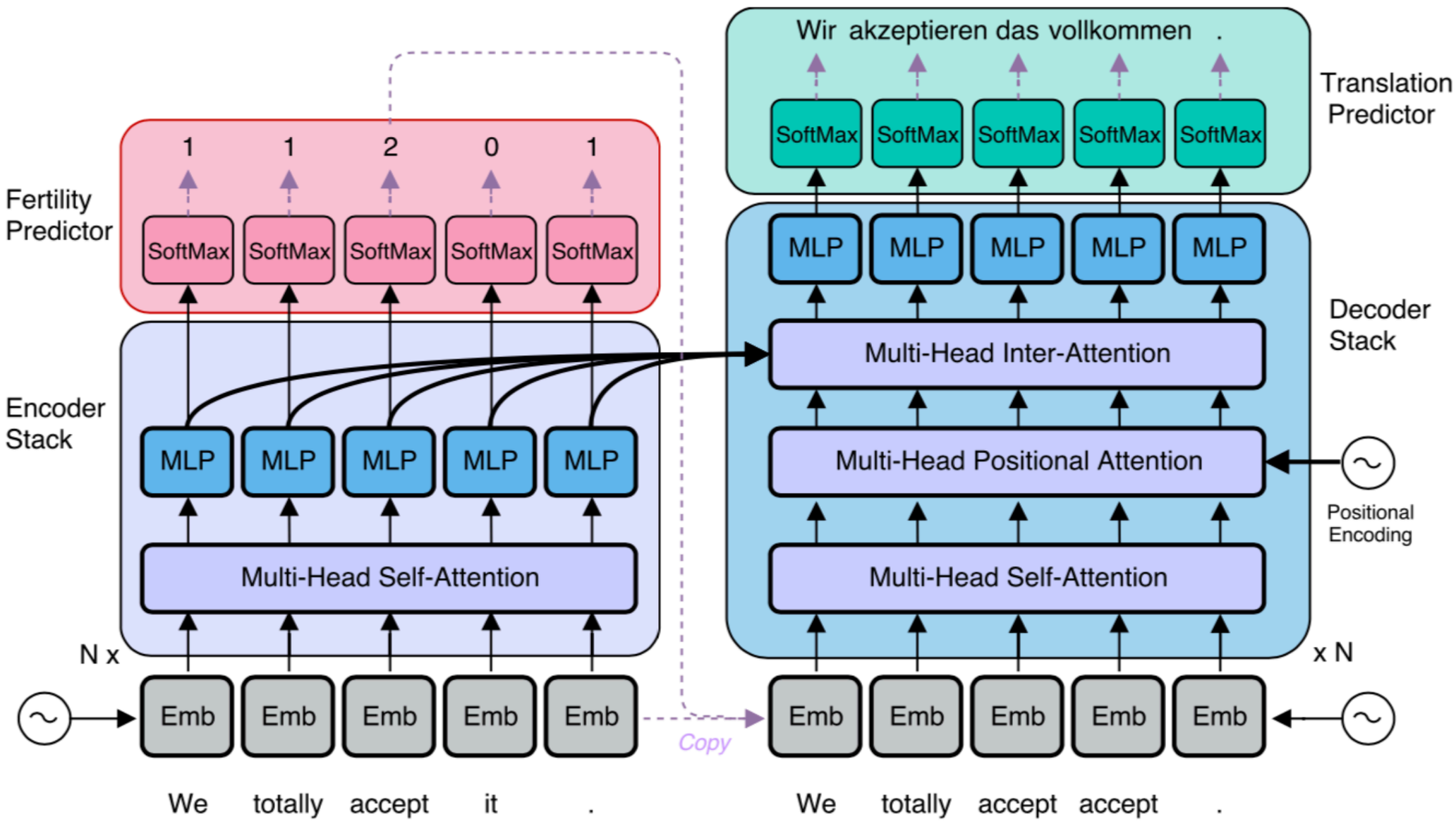
$$P(\boldsymbol{Y}|\boldsymbol{X},\theta) = \prod_{t=1}^{T} p(y_t|\boldsymbol{X},\theta).$$

- Proposed by Gu et al. in ICLR 2018

NON-AUTOREGRESSIVE
NEURAL MACHINE TRANSLATION

Jiatao Gu[†,*], **James Bradbury**[‡], **Caiming Xiong**[‡], **Victor O.K. Li**[†] **& Richard Socher**[‡]
[‡]Salesforce Research
{james.bradbury,cxiong,rsocher}@salesforce.com
[†]The University of Hong Kong
{jiataogu, vli}@eee.hku.hk

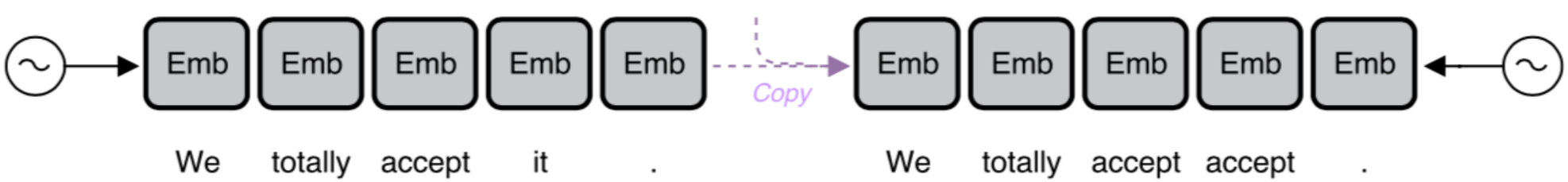
NAT: Model Overview



中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

NAT: Decoder Inputs

- Decoder inputs: source embedding, copied from encoder inputs (Transformer: target embedding)
- Problem: source length and target length is different
- Copy methods: uniform copy、fertility prediction



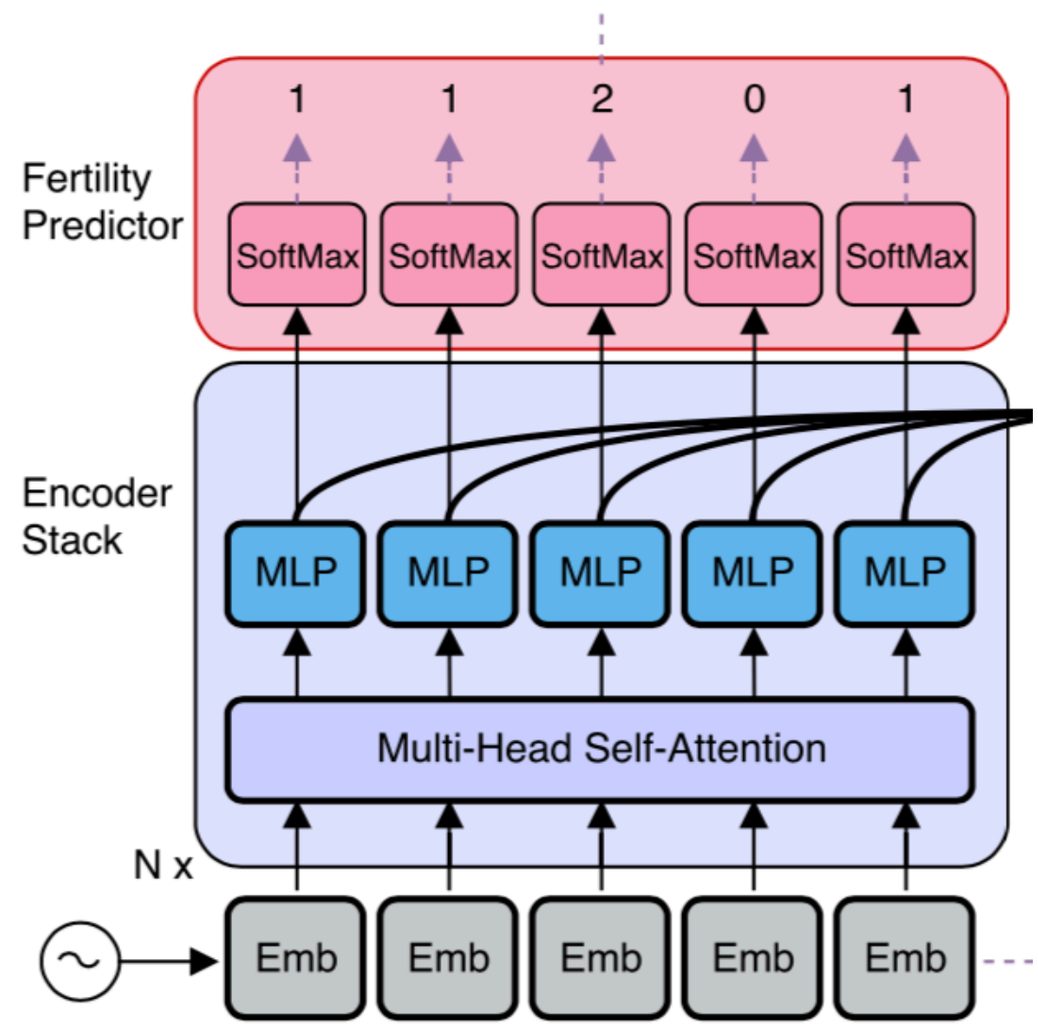
NAT: Decoder Inputs

- uniform: uniformly mapping the source side word embedding as the inputs to the decoder.

$$z_k = \mathcal{E}(x_i),\ i = \left[\frac{T_x}{T_y} t\right],\ t = 1, 2, ..., T_y.$$

- fertility: the copy times of a source word is predicted by the fertility predictor

- uniform copy is more popular



NAT: Length Prediction

- Uniform copy: has to know the target length

$$z_k = \mathcal{E}(x_i),\ i = \left[\frac{T_x}{T_y} t\right],\ t = 1, 2, ..., T_y.$$

- Length predictor: a softmax classifier, predict the target length based on encoder states

- Training: use the ground-truth target length

- Testing: use the predicted target length

NAT: Rescore

- Rescore for NAT: generate many candidates and find the best one, trade-off between quality and speed
- Uniform: target length ranging from [T-B,T+B] to generate 2B+1 candidates
- Fertility: sample from the fertility space to generate candidates
- Rescore: use the autoregressive Transformer to score candidates

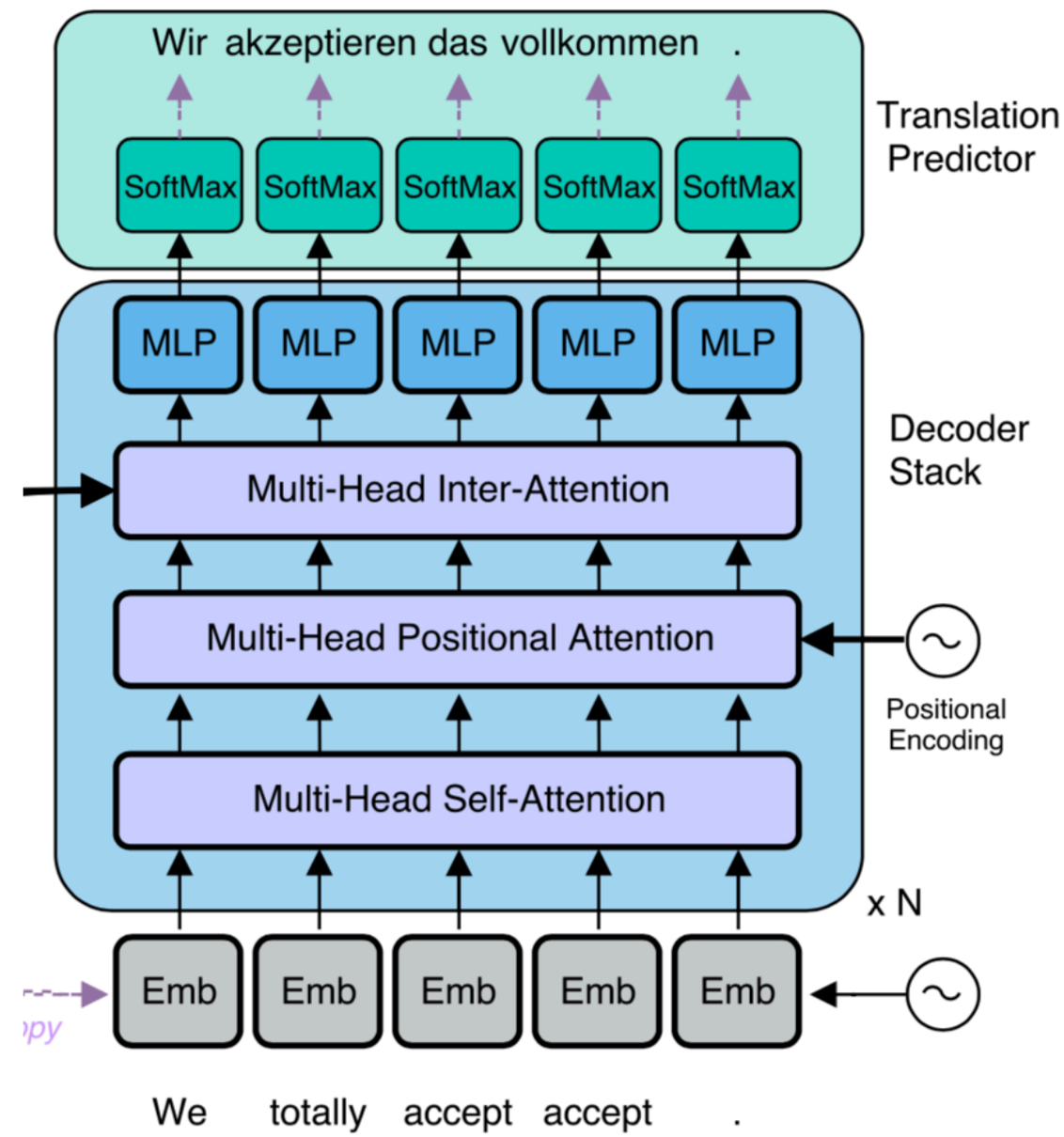
NAT: Positional attention

- Positional attention: an additional positional attention module in each decoder layer

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_{\text{model}}}}\right) \cdot V,$$

- Positional encoding: key and query; decoder states: value

- provides a stronger positional signal



NAT: Knowledge distillation

- Knowledge distillation: train the student model through distilling knowledge from the teacher model
- Sequence-level knowledge distillation: replace the target side of the training corpus by the output of the teacher model
- Student: NAT, Teacher: autoregressive Transformer

Sequence-Level Knowledge Distillation

Yoon Kim
yoonkim@seas.harvard.edu

Alexander M. Rush
srush@seas.harvard.edu

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

NAT: Performance

Models	WMT14		WMT16		IWSLT16		
	En→De	De→En	En→Ro	Ro→En	En→De	Latency / Speedup	
NAT	17.35	20.62	26.22	27.83	25.20	39 ms	15.6×
NAT (+FT)	17.69	21.47	27.29	29.06	26.52	39 ms	15.6×
NAT (+FT + NPD $s = 10$)	18.66	22.41	29.02	30.76	27.44	79 ms	7.68×
NAT (+FT + NPD $s = 100$)	19.17	23.20	29.79	**31.44**	28.16	257 ms	2.36×
Autoregressive ($b = 1$)	22.71	26.39	31.35	31.03	28.89	408 ms	1.49×
Autoregressive ($b = 4$)	23.45	27.02	31.91	31.76	29.70	607 ms	1.00×

Source:	I see wheelchairs bought and sold like used cars .
Target:	ich erlebe , dass Rollstühle gekauft und verkauft werden wie Gebrauchtwagen
AR:	ich sehe Rollstühlen , die wie Autos verkauft und verkauft werden .
NAT:	ich sehe , dass Stühle Stühle und verkauft wie Autos verkauft .
NAT+NPD:	ich sehe Rollühle kauften und verkaufte wie Autos .

Outline

- Introduction
- Non-autoregressive Transformer (NAT)
- **Progress in NAT**
- Our work: Retrieving Sequential Information for Non-Autoregressive Neural Machine Translation

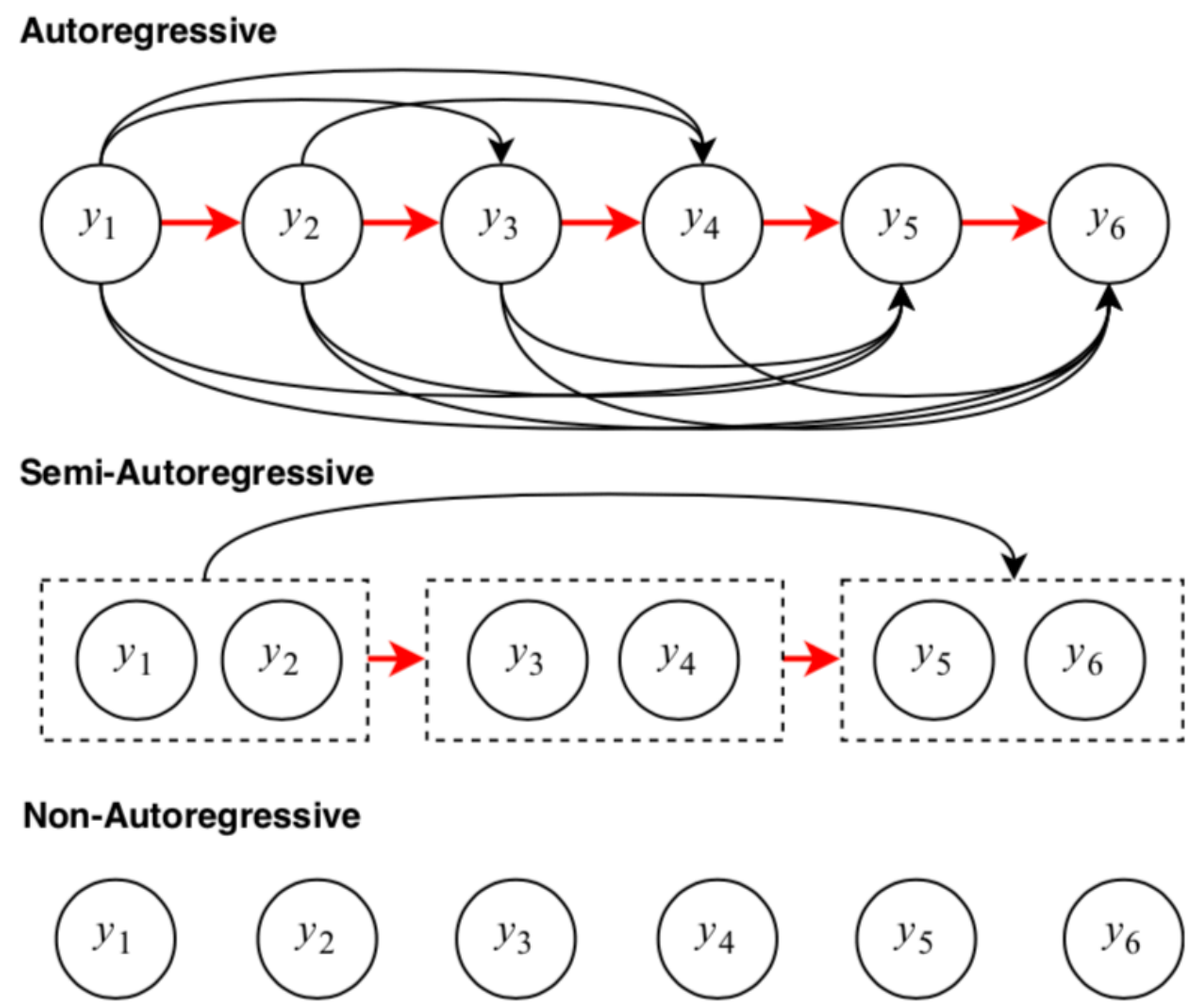
Progress in NAT

- NON-AUTOREGRESSIVE NEURAL MACHINE TRANSLATION (ICLR 2018)
- Fast Decoding in Sequence Models Using Discrete Latent Variables (ICML 2018)
- **Deterministic Non-Autoregressive Neural Sequence Modeling by Iterative Refinement (EMNLP 2018)**
- **Semi-Autoregressive Neural Machine Translation (EMNLP 2018)**
- **Non-Autoregressive Neural Machine Translation with Enhanced Decoder Input (AAAI 2019)**
- Non-Autoregressive Machine Translation with Auxiliary Regularization (AAAI 2019)
- HINT-BASED TRAINING FOR NON-AUTOREGRESSIVE TRANSLATION (openreview)

Progress in NAT

- Semi-Autoregressive Neural Machine Translation (EMNLP 2018)

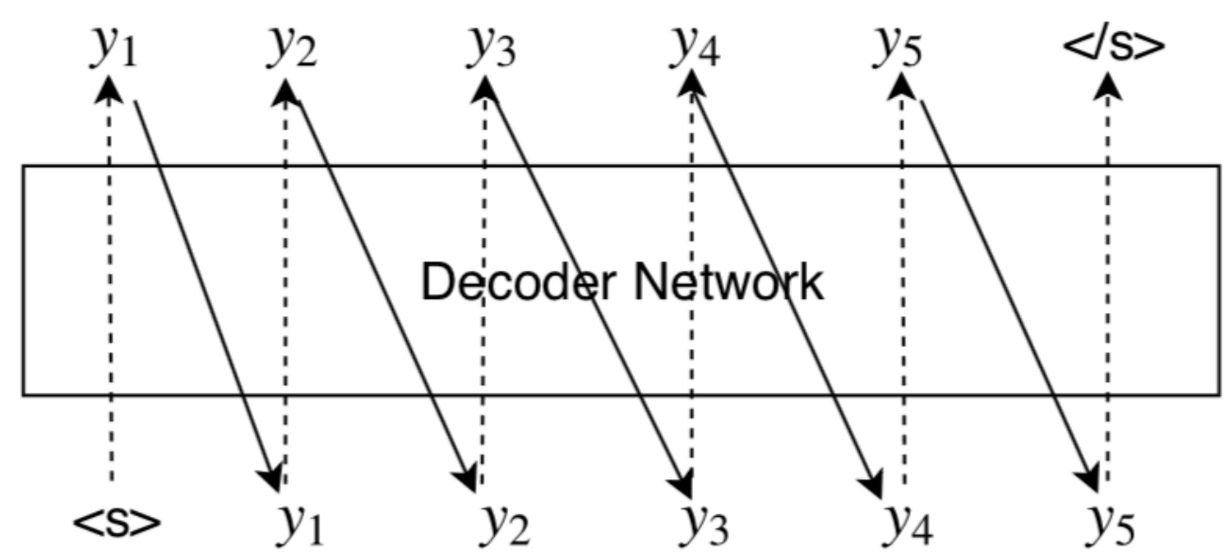
- In one running of the decoder
  - AT: generate one word
  - NAT: generate one sentence
  - SAT: generate K words

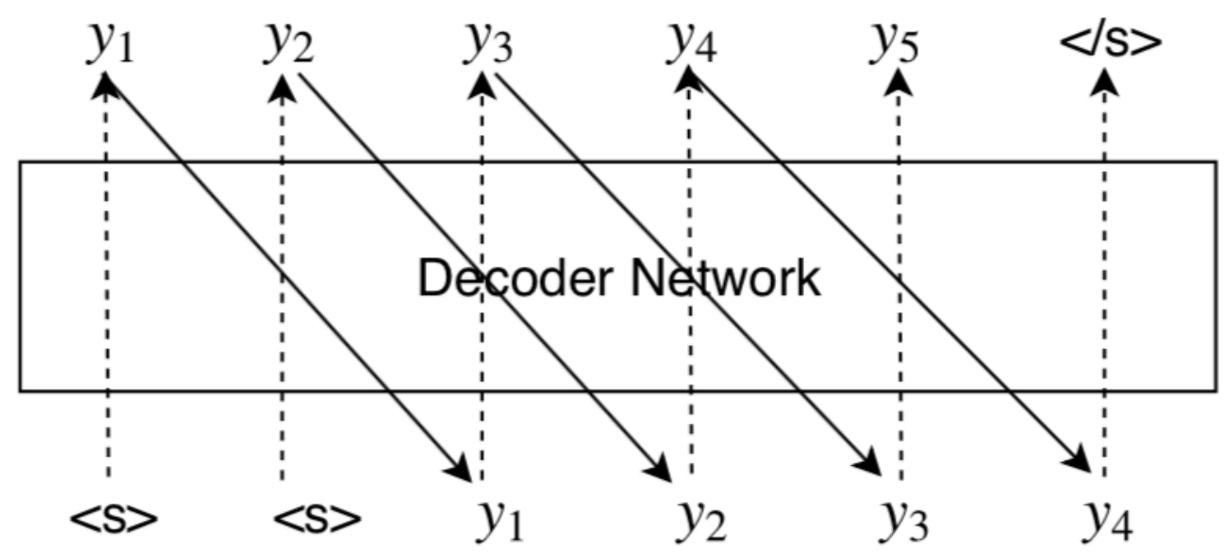


Progress in NAT

- Group-level chain rule
- A group contains K words

$$p(y_1 \dots y_n|\mathbf{x}) = \prod_{t=1}^{[(n-1)/K]+1} p(G_t|G_1 \dots G_{t-1}, \mathbf{x})$$





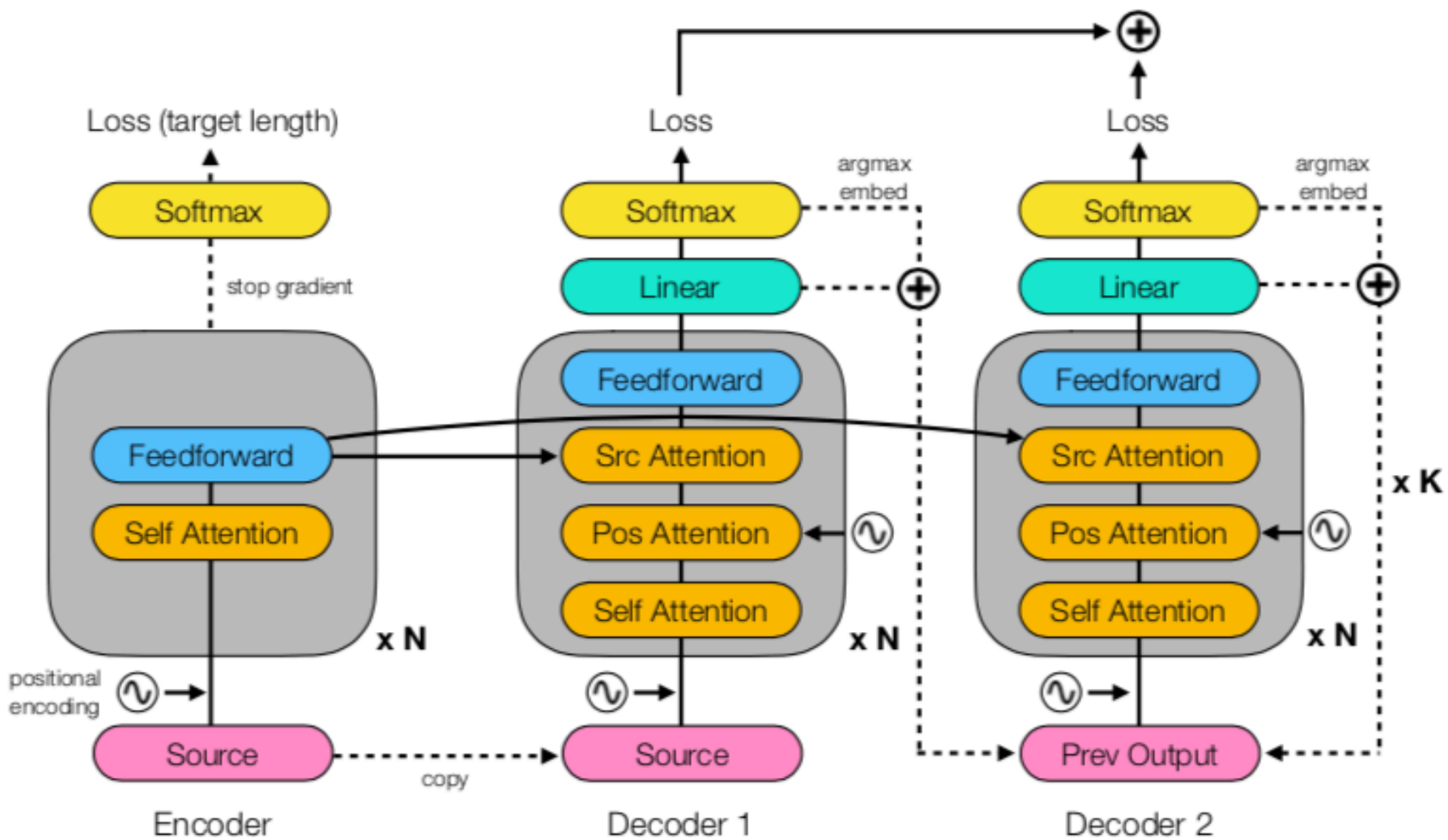
Progress in NAT

Model	Beam Size	BLEU				Degeneration	Lattency	Speedup
		NIST03	NIST04	NIST05	Averaged			
Transformer	4	40.74	40.54	40.48	40.59	0%	410ms	1.00×
	1	39.56	39.72	39.61	39.63	2%	302ms	1.36×
Transformer, N=2	4	37.30	38.55	36.87	37.57	7%	169ms	2.43×
	1	36.26	37.19	35.50	36.32	11%	117ms	3.50×
This Work								
SAT, K=2	4	39.13	40.04	39.55	39.57	3%	243ms	1.69×
	1	37.94	38.73	38.43	38.37	5%	176ms	2.33×
SAT, K=4	4	37.08	38.06	37.12	37.42	8%	152ms	2.70×
	1	35.77	36.43	35.04	35.75	12%	94ms	4.36×
SAT, K=6	4	34.61	36.29	35.06	35.32	13%	129ms	3.18×
	1	33.44	34.54	33.28	33.75	17%	64ms	6.41×

Source	大型校园文化展览也将在会议期间举行。
Transformer	the largescale exhibition of campus culture will also be held during the meeting .
SAT, k=2	the largescale cultural cultural exhibition on campus will also be held during the meeting .
SAT, k=4	the campus campus exhibition will also be held during the meeting .
SAT, k=6	a largescale campus culture exhibition will also be held on the sidelines of the meeting .
Reference	there will also be a large - scale campus culture show during the conference .

Progress in NAT

- Deterministic Non-Autoregressive Neural Sequence Modeling by Iterative Refinement (EMNLP 2018)



Progress in NAT

		IWSLT'16 En-De				WMT'16 En-Ro				WMT'14 En-De				MS COCO		
		En→	De→	GPU	CPU	En→	Ro→	GPU	CPU	En→	De→	GPU	CPU	BLEU	GPU	CPU
AR	$b = 1$	28.64	34.11	70.3	32.2	31.93	31.55	55.6	15.7	23.77	28.15	54.0	15.8	23.47	4.3	2.1
	$b = 4$	28.98	34.81	63.8	14.6	32.40	32.06	43.3	7.3	24.57	28.47	44.9	7.0	24.78	3.6	1.0
NAT	FT	26.52	–	–	–	27.29	29.06	–	–	17.69	21.47	–	–	–	–	–
	FT+NPD	28.16	–	–	–	29.79	31.44	–	–	19.17	23.30	–	–	–	–	–
Our Model	$i_{\text{dec}} = 1$	22.20	27.68	573.0	213.2	24.45	25.73	694.2	98.6	13.91	16.77	511.4	83.3	20.12	17.1	8.9
	$i_{\text{dec}} = 2$	24.82	30.23	423.8	110.9	27.10	28.15	332.7	62.8	16.95	20.39	393.6	49.6	20.88	12.0	5.7
	$i_{\text{dec}} = 5$	26.58	31.85	189.7	52.8	28.86	29.72	194.4	29.0	20.26	23.86	139.7	23.1	21.12	6.2	2.8
	$i_{\text{dec}} = 10$	27.11	32.31	98.8	24.1	29.32	30.19	93.1	14.8	21.61	25.48	90.4	12.3	21.24	2.0	1.2
	Adaptive	27.01	32.43	125.9	29.3	29.66	30.30	118.3	16.5	21.54	25.43	107.2	20.3	21.12	10.8	4.8

Src	er sah sehr glücklich aus , was damals ziemlich ungewöhnlich war , da ihn die Nachrichten meistens deprimierten .
Iter 1	he looked very happy , which was pretty unusual the , because the news was were usually depressing .
Iter 2	he looked very happy , which was pretty unusual at the , because the news was s depressing .
Iter 4	he looked very happy , which was pretty unusual at the , because news was mostly depressing .
Iter 8	he looked very happy , which was pretty unusual at the time because the news was mostly depressing .
Iter 20	he looked very happy , which was pretty unusual at the time , because the news was mostly depressing .
Ref	there was a big smile on his face which was unusual then , because the news mostly depressed him .

Progress in NAT

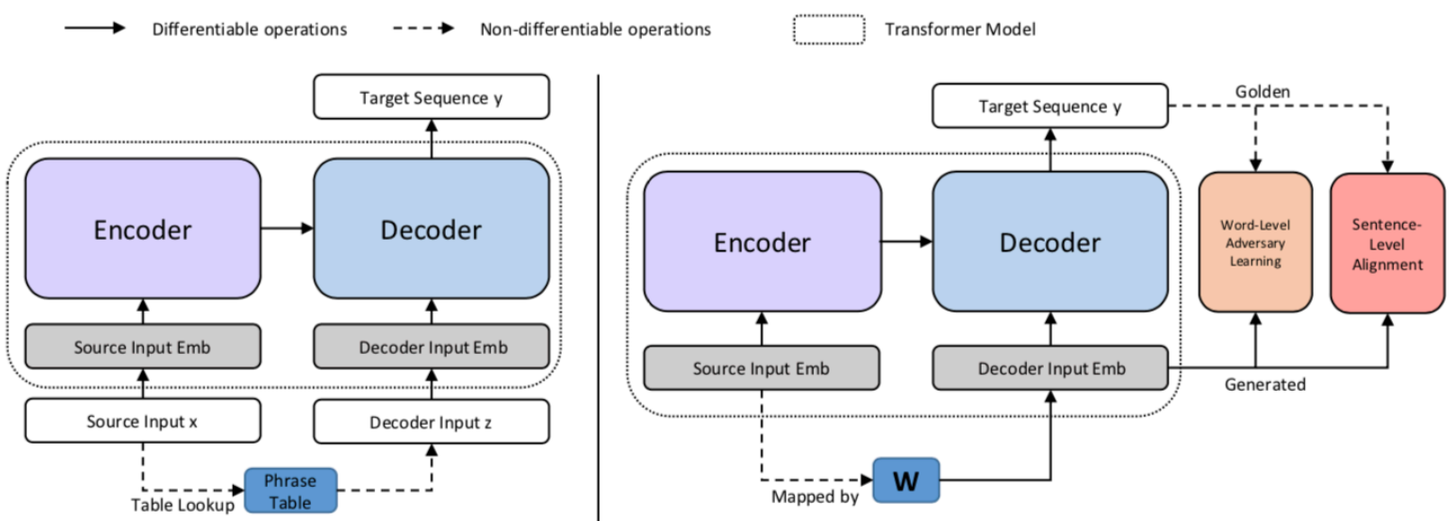
- Non-Autoregressive Neural Machine Translation with Enhanced Decoder Input (AAAI 2019)
- Decoder inputs of NAT: copied from source embeddings
- Enhanced decoder input: Directly feed target-side tokens as the inputs of the decoder
- Two methods: Phrase-Table Lookup、Embedding Mapping

Progress in NAT

- Phrase-Table Lookup: pre-train a Phrase-Table using SMT, lookup the Phrase-Table to generate the translation

- Embedding Mapping: linearly map the source embeddings to target embeddings

$$\min_{\Theta} \max_{\theta_D} L(x, y) = L_{\text{neg}}(x, y; \theta_{\text{enc}}, \theta_{\text{dec}}) + \mu L_{\text{align}}(x, y; W) + \lambda L_{\text{adv}}(x, y; \theta_D, W),$$



Phrase-Table Lookup

Embedding Mapping

Progress in NAT

Models	WMT14 En−De	WMT14 De−En	WMT16 En−Ro	IWSLT14 De−En	Latency	Speedup
LSTM-based S2S (Wu et al. 2016)	24.60	/	/	28.53†	/	/
Transformer (Vaswani et al. 2017)	27.41†	31.29†	35.61†	32.55†	607 ms	1.00×
LT (Kaiser et al. 2018)	19.80	/	/	/	105 ms	5.78×
LT (rescoring 10 candidates)	21.00	/	/	/	/	/
LT (rescoring 100 candidates)	22.50	/	/	/	/	/
NART (Gu et al. 2017)	17.69	21.47	27.29	22.95†	39 ms	15.6×
NART (rescoring 10 candidates)	18.66	22.41	29.02	25.05†	79 ms	7.68×
NART (rescoring 100 candidates)	19.17	23.20	29.79	/	257 ms	2.36×
IR-NAT (Lee, Mansimov, and Cho 2018)	21.54	25.42	29.66	/	254† ms	2.39×
Phrase-Table Lookup	6.03	11.24	9.16	15.69	/	/
ENAT Phrase-Table Lookup	20.26	23.23	29.85	25.09	25 ms	24.3×
ENAT Phrase-Table Lookup (rescoring 9 candidates)	23.22	**26.67**	34.04	**28.60**	50 ms	12.1×
ENAT Embedding Mapping	20.65	23.02	30.08	24.13	**24** ms	**25.3**×
ENAT Embedding Mapping (rescoring 9 candidates)	**24.28**	26.10	**34.51**	27.30	49 ms	12.4×

Table 2: BLEU scores on WMT14 En-De, WMT14 De-En, WMT16 En-Ro and IWSLT14 De-En tasks. "/" indicates the corresponding result is not reported and "†" means results are produced by ourselves. We also list the inference latency compared with previous works. ENAT with rescoring 9 candidates indicates results when $B = 4$, otherwise $B = 0$.

Outline

- Introduction
- Non-autoregressive Transformer (NAT)
- Progress in NAT
- **Our work: Retrieving Sequential Information for Non-Autoregressive Neural Machine Translation**

Retrieving Sequential Information for Non-Autoregressive Neural Machine Translation

Chenze Shao[1 2 3†], Yang Feng[1 2⋆], Jinchao Zhang[3], Fandong Meng[3], Xilin Chen[1 2] and Jie Zhou[3]

[1] University of Chinese Academy of Sciences
[2] Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)
[3] Pattern Recognition Center, WeChat AI, Tencent Inc, China
{shaochenze18z, fengyang, xlchen}@ict.ac.cn
{dayerzhang, fandongmeng, withtomzhou}@tencent.com

Contents

- Problem: over-translation and under-translation errors due to the lack of target-side sequential information
- Solution: retrieve the target-side sequential information
    - Evaluation: Reinforce-NAT
    - Generation: FS-decoder
- Experiments and analysis

Problem: lack of target-side sequential information

- Generation: NAT generate target words independently
- Evaluation: the cross-entropy loss evaluates every generation independently
- No sequential information involved, inaccurate loss function

Source: 我 必须 解决 这个 问题 。

Refer: I must solve the problem .

Trans: I have to solve the problem

Problem: lack of target-side sequential information

- Weak in modeling sequential dependency
- over-translation and under-translation errors

Src	und noch tragischer ist , dass es Oxford war · · ·
Ref	even more tragic is that it was Oxford · · ·
NAT	and more more more more that it was Oxford · · ·
AR	and , more tragic , Oxford was · · ·

Solutions

- Retrieve the target-side sequential information for NAT models and meanwhile preserve the fast-decoding property
- Evaluation: sequence-level training based on a novel reinforcement algorithm for NAT (Reinforce-NAT)
- Generation: fuse the sequential information into the top layer of decoder (FS-decoder)

Reinforce-NAT

- Retrieve sequential information in evaluation: sequence-level training
- Training NAT models with sequence-level objectives (e.g., BLEU, GLEU, TER)
- Encourage NAT to generate high quality sentences rather than correct tokens
- Background: REINFORCE for autoregressive models Reinforce-NAT

Reinforce-NAT

- Background: REINFORCE for autoregressive models
- Loss function: the negative expected reward

$$\mathrm{L}_\theta = -\sum_{\mathbf{Y}} p(\mathbf{Y}|\mathbf{X},\theta) \cdot r(\mathbf{Y})$$

- Gradient of the loss function

$$\nabla_\theta \mathrm{L}_\theta = -\sum_{\mathbf{Y}} \nabla_\theta p(\mathbf{Y}|\mathbf{X},\theta) \cdot r(\mathbf{Y})$$

- Problem: exponential search space

Reinforce-NAT

- REINFORCE: estimate the gradient using the log-probability of the sampled sentence

$$
\begin{aligned}
\nabla_\theta \mathrm{L}_\theta &= -\sum_{\mathbf{Y}} \nabla_\theta p(\mathbf{Y}|\mathbf{X},\theta) \cdot r(\mathbf{Y}) \\
&= -\sum_{\mathbf{Y}} p(\mathbf{Y}|\mathbf{X},\theta) \cdot \frac{\nabla_\theta p(\mathbf{Y}|\mathbf{X},\theta)}{p(\mathbf{Y}|\mathbf{X},\theta)} \cdot r(\mathbf{Y}) \\
&= -\mathop{\mathbb{E}}_{\mathbf{Y}\sim p}[\nabla_\theta \log p(\mathbf{Y}|\mathbf{X},\theta) \cdot r(\mathbf{Y})] \\
&= -\mathop{\mathbb{E}}_{\mathbf{Y}\sim p}[\sum_{t=1}^{T} \nabla_\theta \log p(y_t|\boldsymbol{y}_{<t},\boldsymbol{X},\theta) \cdot r(\mathbf{Y})]
\end{aligned}
$$

Reinforce-NAT

- Directly applying REINFORCE in NAT

$$P(\boldsymbol{Y}|\boldsymbol{X},\theta) = \prod_{t=1}^{T} p(y_t|\boldsymbol{X},\theta).$$

$$\nabla_\theta \mathrm{L}_\theta = -\sum_{\mathbf{Y}} \nabla_\theta \prod_{t=1}^{T} p(y_t|\boldsymbol{X},\theta) \cdot r(\mathbf{Y})$$

$$\nabla_\theta \mathrm{L}_\theta = -\mathop{\mathbb{E}}_{\mathbf{Y}}[\sum_{t=1}^{T} \nabla_\theta \log(p(y_t|\mathbf{X},\theta)) \cdot r(\mathbf{Y})].$$

- The same reward for every token, high variance, unstable

- improve the algorithm with some unique properties of NAT

Reinforce-NAT

- Gradient: target probabilities weighted by their corresponding rewards

$$\nabla_\theta \mathrm{L}_\theta = -\sum_{\mathbf{Y}} \nabla_\theta \prod_{t=1}^{T} p(y_t|\boldsymbol{X},\theta)\cdot r(\mathbf{Y}) = -\sum_{t=1}^{T}\sum_{y_t} \nabla_\theta p(y_t|\mathbf{X},\theta)\cdot r(y_t).$$

$$r(y_t) = \mathop{\mathbb{E}}_{y_{1:t-1}} \mathop{\mathbb{E}}_{y_{t+1:T}} r(\mathbf{Y})$$

- REINFORCE: log-probability estimation

$$\nabla_\theta \mathrm{L}_\theta = -\sum_{t=1}^{T} \mathop{\mathbb{E}}_{y_t}[\nabla_\theta \log(p(y_t|\mathbf{X},\theta))\cdot r(y_t)]$$

- Reward $r(y_t)$: Monte-carlo sampling

Algorithm 1 Estimation of $r(y_t)$

Input: the output probability distribution $p(\cdot|\mathbf{X},\theta))$, t, y_t, T, sampling times n
Output: estimate of $r(y_t)$

```
r = 0, i = 0
for i < n do
    sample ỹ_{1:t-1}, ỹ_{t+1:T} from p(·|X, θ))
    Ỹ = {ỹ_{1:t-1}, y_t, ỹ_{t+1:T}}
    r += r(Ỹ)
    i += 1
r = r/n
return r
```

$$
\begin{aligned}
\nabla_\theta \mathrm{L}_\theta &= \sum_{\mathbf{Y}} \nabla_\theta \prod_{t=1}^{T} p(y_t|\boldsymbol{X},\theta) \cdot r(\mathbf{Y}) \\
&= \sum_{\mathbf{Y}} \sum_{t=1}^{T} \nabla_\theta p(y_t|\boldsymbol{X},\theta) \cdot \prod_{i=1}^{t-1} p(y_i|\boldsymbol{X},\theta) \cdot \prod_{j=t+1}^{T} p(y_j|\boldsymbol{X},\theta) \cdot r(\mathbf{Y}) \\
&= \sum_{t=1}^{T} \sum_{\mathbf{Y}} \nabla_\theta p(y_t|\boldsymbol{X},\theta) \cdot \prod_{i=1}^{t-1} p(y_i|\boldsymbol{X},\theta) \cdot \prod_{j=t+1}^{T} p(y_j|\boldsymbol{X},\theta) \cdot r(\mathbf{Y}) \\
&= \sum_{t=1}^{T} \sum_{y_t} \nabla_\theta p(y_t|\boldsymbol{X},\theta) \cdot \sum_{\mathbf{y_{1:t-1}}} \sum_{\mathbf{y_{t+1:T}}} \prod_{i=1}^{t-1} p(y_i|\boldsymbol{X},\theta) \cdot \prod_{j=t+1}^{T} p(y_j|\boldsymbol{X},\theta) \cdot r(\mathbf{Y}) \\
&= \sum_{t=1}^{T} \sum_{y_t} \nabla_\theta p(y_t|\boldsymbol{X},\theta) \cdot \mathop{\mathbb{E}}_{y_{1:t-1}} \mathop{\mathbb{E}}_{y_{t+1:T}} r(\mathbf{Y}). \\
&= \sum_{t=1}^{T} \sum_{y_t} \nabla_\theta p(y_t|\mathbf{X},\theta) \cdot r(y_t)
\end{aligned}
$$

Reinforce-NAT

- Estimate the gradient: $\nabla_\theta \mathrm{L}_\theta = \sum_{t=1}^{T} \sum_{y_t} \nabla_\theta p(y_t|\mathbf{X}, \theta) \cdot r(y_t)$
- REINFORCE: sample one word from the vocabulary, high variance; the vocabulary is too large to traverse
- In neural machine translation, top-ranking words usually occupy the central part of the probability distribution
- Motivation of Reinforce-NAT: traversing the top-ranking words, estimating the rest through REINFORCE

Reinforce-NAT

- Reinforce-NAT: compute the gradients of the top-ranking words accurately and estimate the rest via the REINFORCE algorithm

- Define T_k by the set of top-k words, P_k by the sum of probabilities in T_k, $\tilde{p}$ by the normalized probability

- An unbiased estimation of the gradient:

$$\nabla_\theta \mathrm{L}_\theta = -\sum_{t=1}^{T}\Big(\sum_{y_t \in T_K} \nabla_\theta p(y_t|\mathbf{X},\theta)\cdot r(y_t) + (1-P_k)\cdot \mathop{\mathbb{E}}_{y_t\sim\tilde{p}}[\nabla_\theta \log(p(y_t|\mathbf{X},\theta))\cdot r(y_t)]\Big)$$

Reinforce-NAT

- unbiased gradient estimation of Reinforce-NAT: traversing and sampling.

$$\nabla_\theta \mathrm{L}_\theta = -\sum_{t=1}^{T}\Big(\sum_{y_t \in T_K} \nabla_\theta p(y_t|\mathbf{X},\theta)\cdot r(y_t) + (1-P_k)\cdot \mathop{\mathbb{E}}_{y_t\sim\tilde{p}}[\nabla_\theta \log(p(y_t|\mathbf{X},\theta))\cdot r(y_t)]\Big)$$

- Traverse the important words that dominate the gradient estimation, and estimate the gradient of less important words via one sampling.

- Efficiency of NAT: do not need to re-run the decoder.

Algorithm 2 Reinforce-NAT

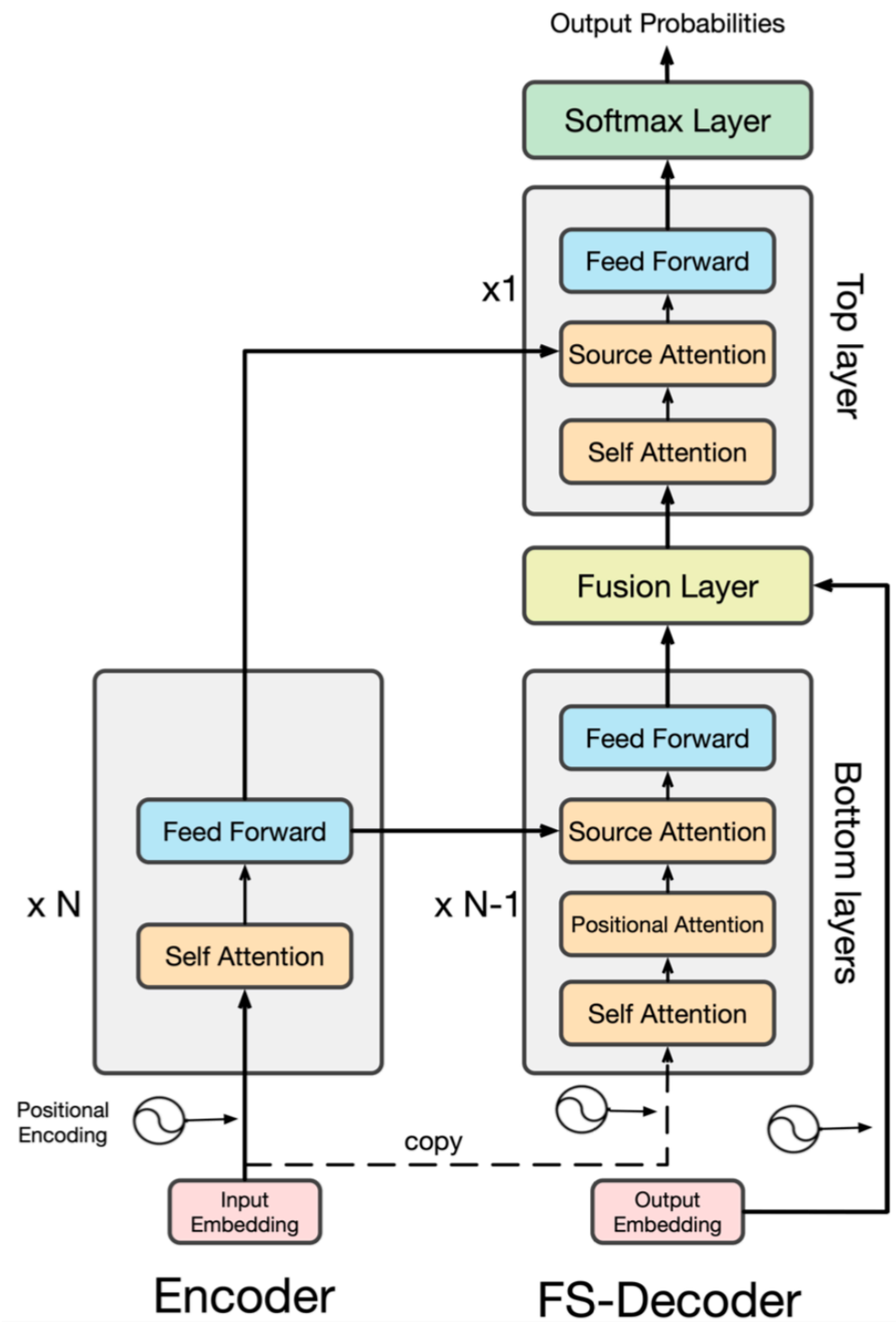
Input: the output probability distribution $p(\cdot|\mathbf{X},\theta))$, traversing count k, sample times n

Output: estimate of $\nabla_\theta \mathrm{L}_\theta$ in position t according to Eq.(12)

```
T_K = {words ranking top-k in p(·|X, θ))}
∇_θ L_θ = 0, p̃ = p, P_k = 0
for y_t in T_K do
    estimate r(y_t) by algorithm 1 with sample times n
    ∇_θ L_θ += ∇_θ p(y_t|X, θ) · r(y_t)
    p̃(y_t|X, θ) = 0
    P_k += p(y_t|X, θ)
normalize p̃(·|X, θ)
sample y_t from p̃(·|X, θ)
estimate r(y_t) by algorithm 1 with sample times n
∇_θ L_θ += (1 − P_k) · ∇_θ log(p(y_t|X, θ)) · r(y_t)
return ∇_θ L_θ
```

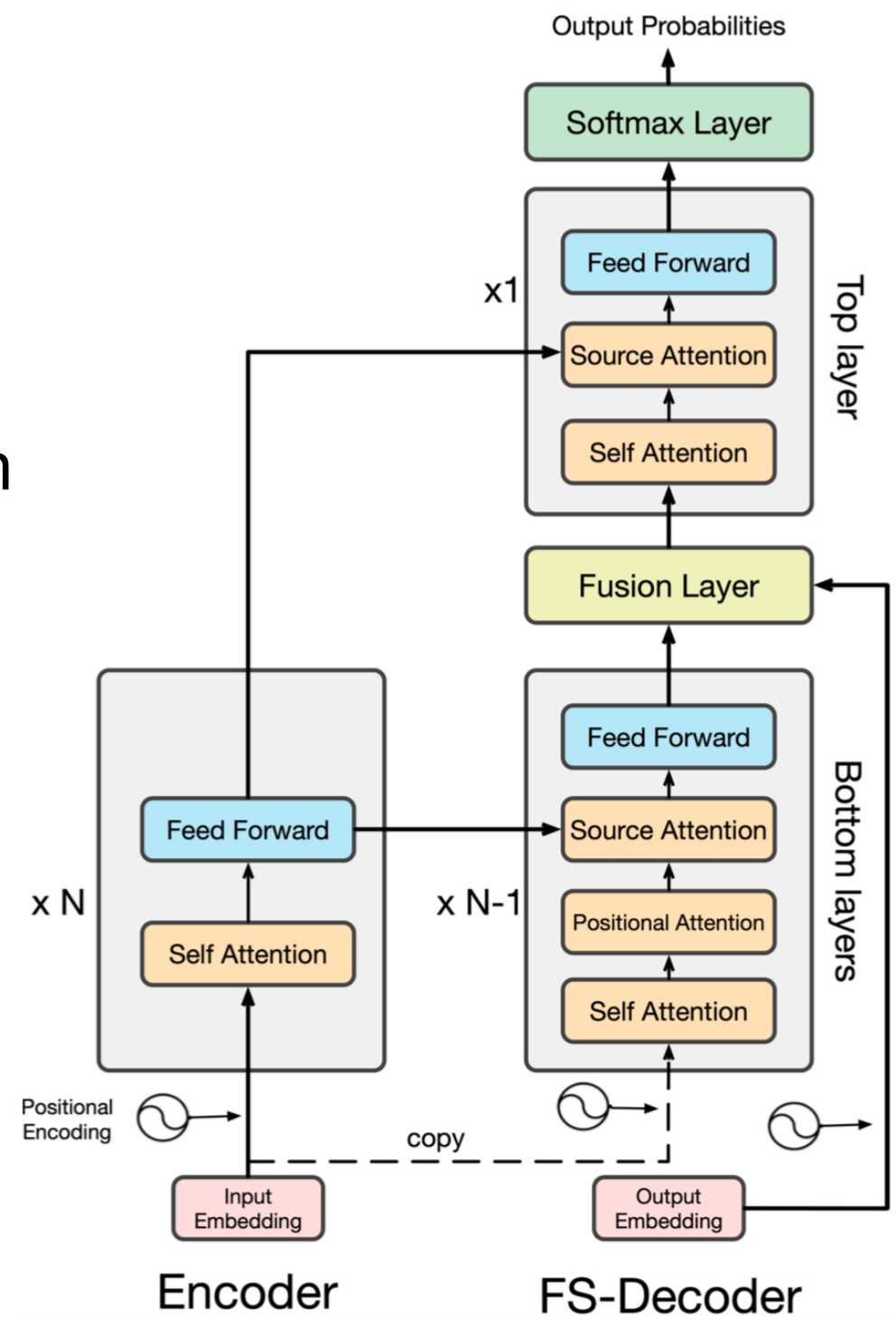
FS-Decoder

- Retrieve sequential information in evaluation: FS-Decoder
- Four parts: bottom layers, the fusion layer, the top layer and the softmax layer.
- Target-side sequential information is fused into the fusion layer.
- Bottom layers run in parallel
- the fusion layer and top layer runs autoregressively



FS-Decoder

- Bottom layers: identical to the NAT decoder layer
- Fusion layer: a linear transformation with a ReLU activation
- Fuses the outputs from bottom layers and the target embedding
- Top layer: identical to the Transformer decoder layer, runs autoregressively during decoding



Experiments

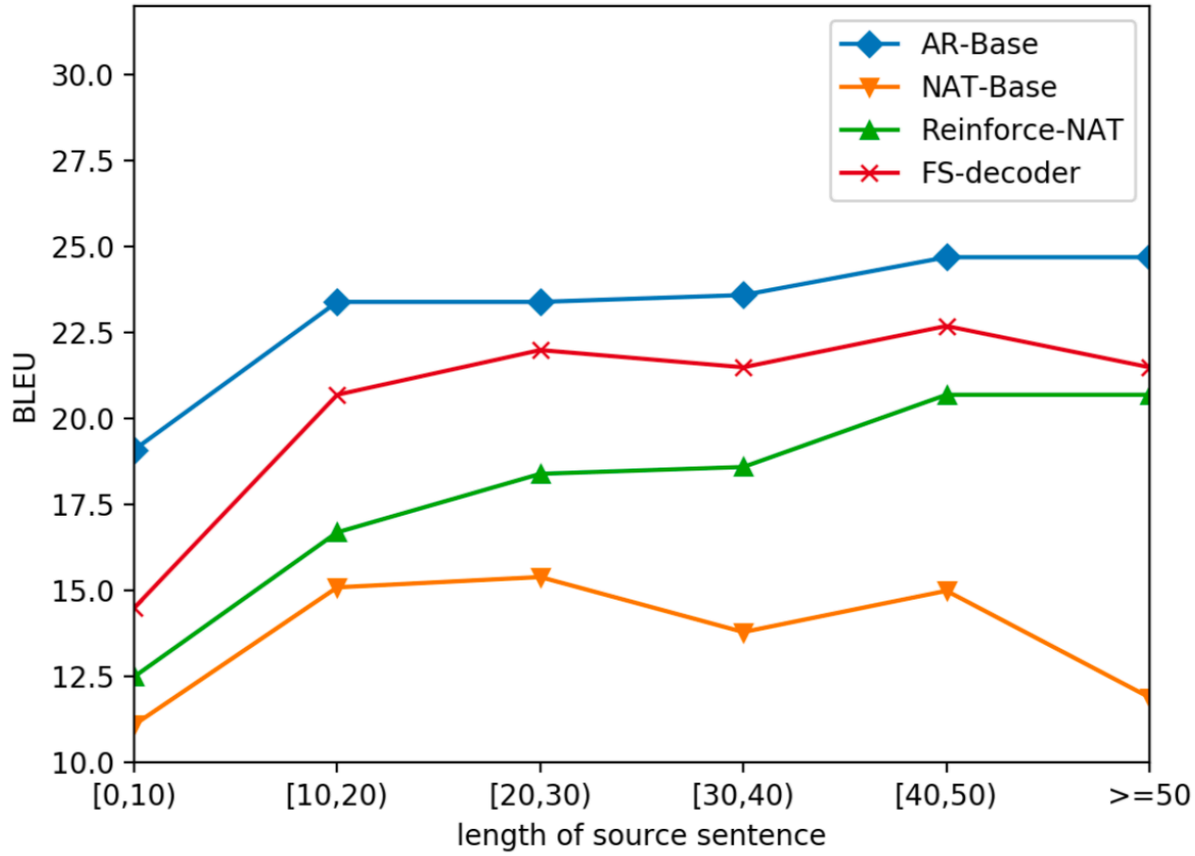
		IWSLT'16 En-De				WMT'16 En-Ro				WMT'14 En-De			
		En→	toks/s	speedup	secs/b	En→	Ro→	toks/s	speedup	En→	De→	toks/s	speedup
AR	b=1	28.13	45.3	1.09×	0.20	31.53	31.35	45.6	1.23×	23.67	28.04	33.7	1.13×
	b=4	28.25	41.6	1.00×	0.20	31.85	31.60	37.1	1.00×	24.29	28.86	29.9	1.00×
NAT	FT	26.52	–	15.6 ×	–	27.29	29.06	–	–	17.69	21.47	–	–
	FT+NPD	28.16	–	2.36 ×	–	29.79	31.44	–	–	19.17	23.20	–	–
IRNAT	iter=2	24.82	423.8	6.64 ×	–	27.10	28.15	332.7	7.68 ×	16.95	20.39	393.6	8.77 ×
	adaptive	27.01	125.9	1.97 ×	–	29.66	30.30	118.3	2.73 ×	21.54	25.43	107.2	2.39 ×
Our Models	NAT-base	24.13	350.2	8.42×	0.62	25.96	26.49	349.0	9.41×	16.05	19.46	321.7	10.76×
	+REINFORCE	24.30	354.1	8.51×	2.51	26.49	27.20	346.7	9.35×	18.47	21.89	323.2	10.81×
	+Reinforce-NAT	25.18	350.6	8.43×	13.40	27.09	27.93	350.3	9.44×	19.15	22.52	320.9	10.73×
	FS-decoder(b=1)	27.58	168.7	4.06×	0.241	30.53	30.68	170.5	4.60×	21.53	27.20	143.3	4.79×
	FS-decoder(b=4)	27.78	140.8	3.38×	0.241	30.57	30.83	137.1	3.70×	22.27	27.25	112.2	3.75×

- Reinforce-NAT: 1.05 BLEU improvements on IWSLT16 En→De, about 1.3 BLEU improvements on WMT16 En<->Ro, about 3.0 BLEU improvements on WMT14 En<->De

- FS-decoder: comparable to the autoregressive transformer , 4 times faster

Experiments

- Performance over Different Lengths
- NAT models are weak in translating long sentences
- Reinforce-NAT and FS-decoder can effectively improve the performance of NAT models in long sentences



Experiments

- Case Study
- NAT-base: over-translation and under-translation
- efficiently alleviated by Reinforce-NAT and FS-Decoder

Source	und noch tragischer ist , dass es Oxford war - eine Universitt , die nicht nur 14 Tory-Premierminister hervorbrachte , sondern sich bis heute hinter einem unverdienten Ruf von Gleichberechtigung und Gedankenfreiheit versteckt .
Target	even more tragic is that it was Oxford , which not only produced 14 Tory prime ministers , but , to this day , hides behind an ill-deserved reputation for equality and freedom of thought .
NAT-Base	and more more more more that it was Oxford - a university that not not only only TTory Prime Minister , but has has to hidden hidden behind an unfounded reputation of equality and freedom of thought .
Reinforce-NAT	and more more tragic is that it was Oxford - a university that did not only produce 14 Tory Prime Minister , but has still to be hidden behind an unfied reputation of equality and freedom of thought .
FS-decoder	and even more tragic , it was Oxford - a university that produced not only 14 Tory Prime Minister , but still hidden behind an unbridled reputation of equality and freedom of thought .
AR-Base	and , more tragic , Oxford was - a university that not only produced 14 Tory Prime Minister , but still hidden behind an unprecedented reputation for equality and freedom of thought .

Thanks

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES