Revisiting Text Generation Methods in Neural Machine Translation

Long Zhou (周龙)
Natural Language Computing Group at Microsoft Research Asia
lozhou@microsoft.com
CCMT, 10/10/2020

Microsoft

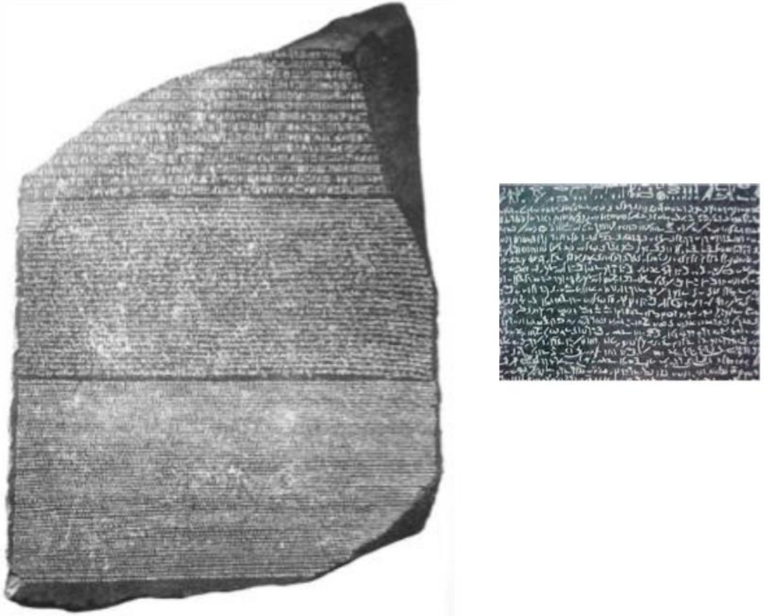
Content

- Background: Machine Translation
- **Autoregressive Neural Machine Translation**
- **Non-Autoregressive Neural Machine Translation**
- **Bidirectional Neural Machine Translation**
- Summary & Future Work

Section 1:
Background: Machine Translation

Language & Translation

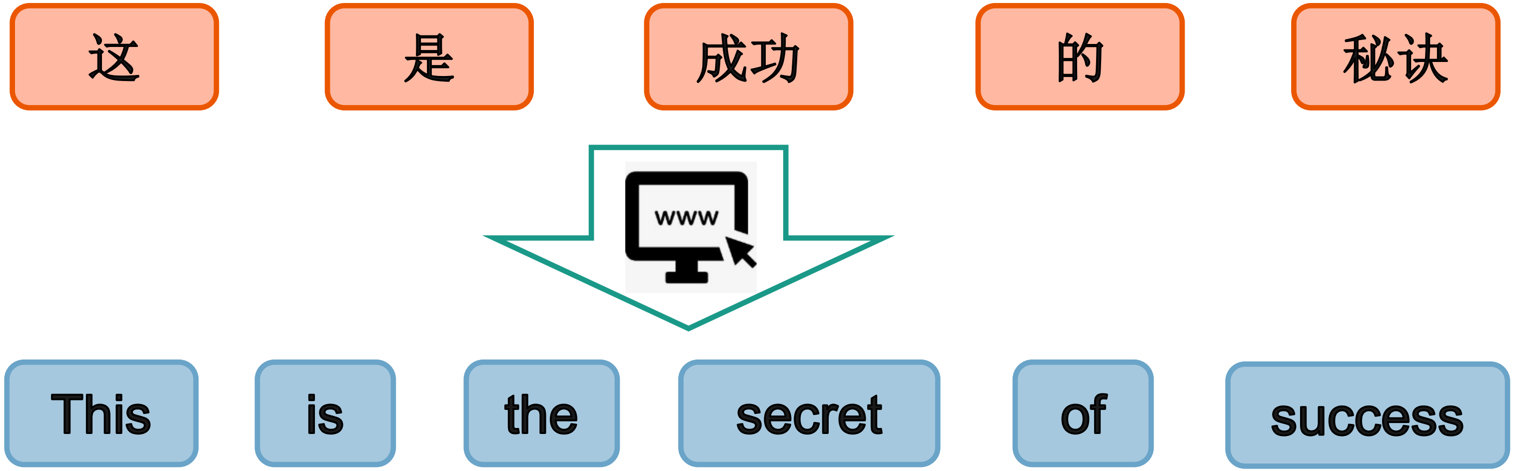
Tower of Babel (巴别塔)



Rosetta Stone (罗塞塔石碑)

Machine Translation

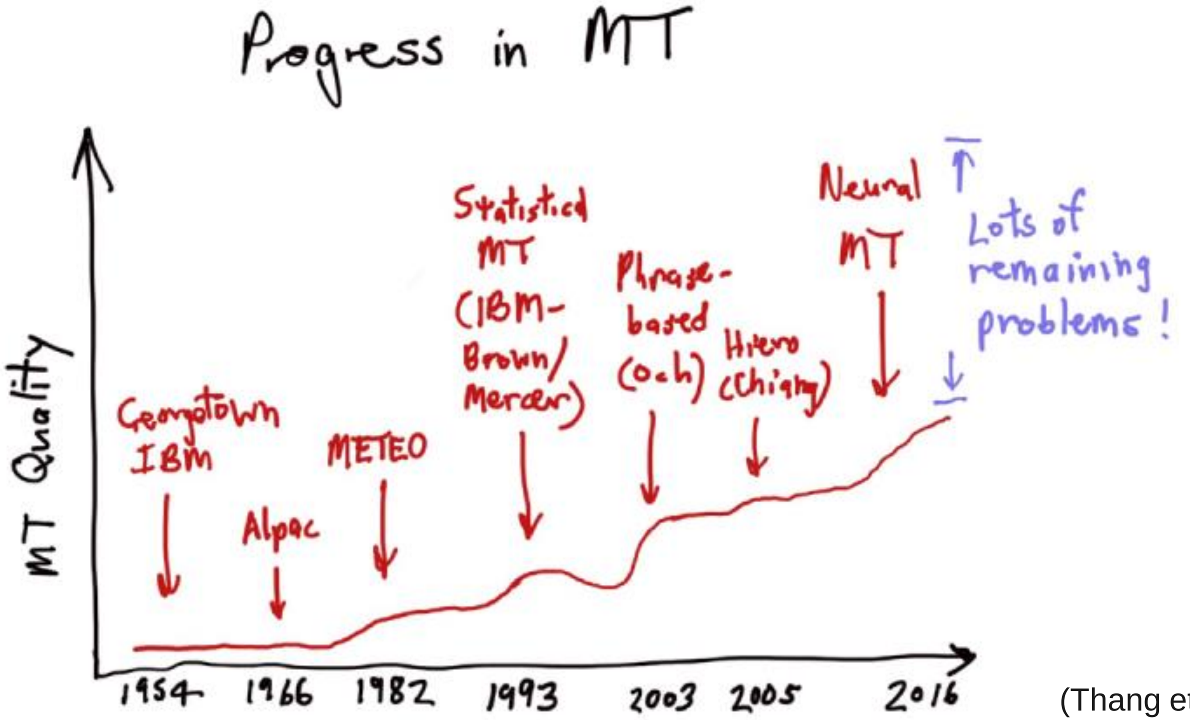
- **Automatically translate language by computer**



Machine Translation

- **Automatically translate language by computer**

Machine Translation



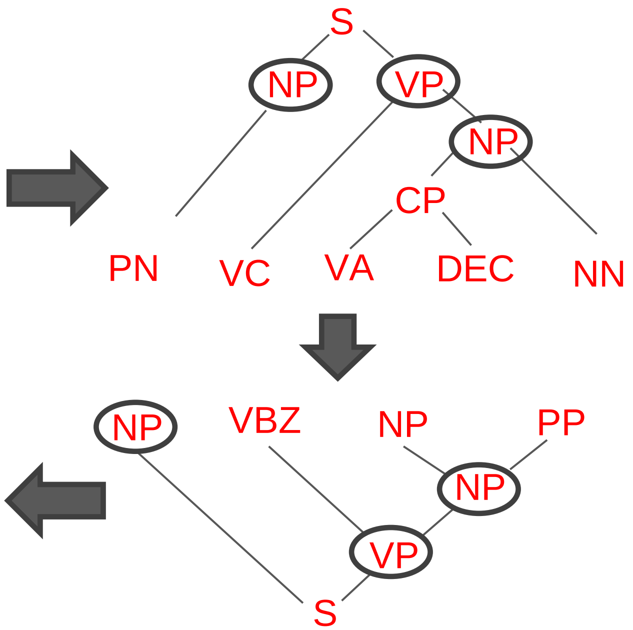
(Thang et al., 2016)

Rule based Machine Translation

- **Source: 这是成功的秘诀**

Tagging:

这/PN 是/VC 成功/VA 的/DEC 秘诀/NN



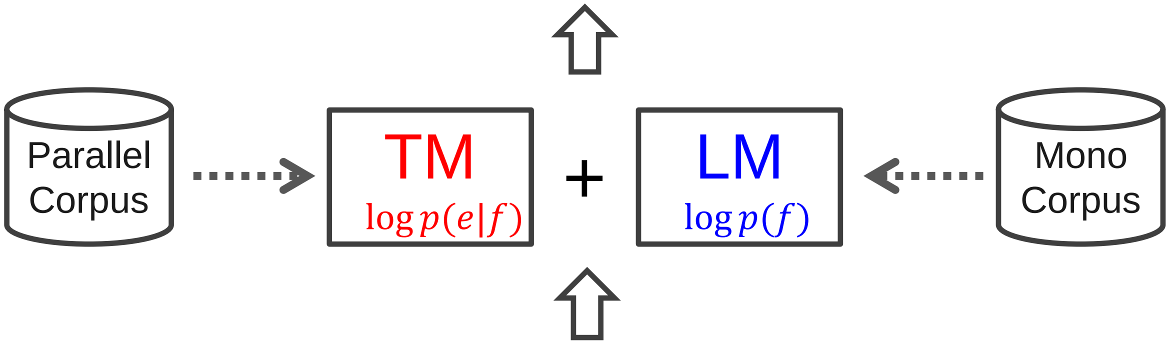
Translate:

#这 PN: this ||| #是 VC: is …

- **Target: This is the secret of success**

Statistical Machine Translation

$$f = (\text{this},\ \ is,\ \ success,\ \ of,\ \ secret)$$

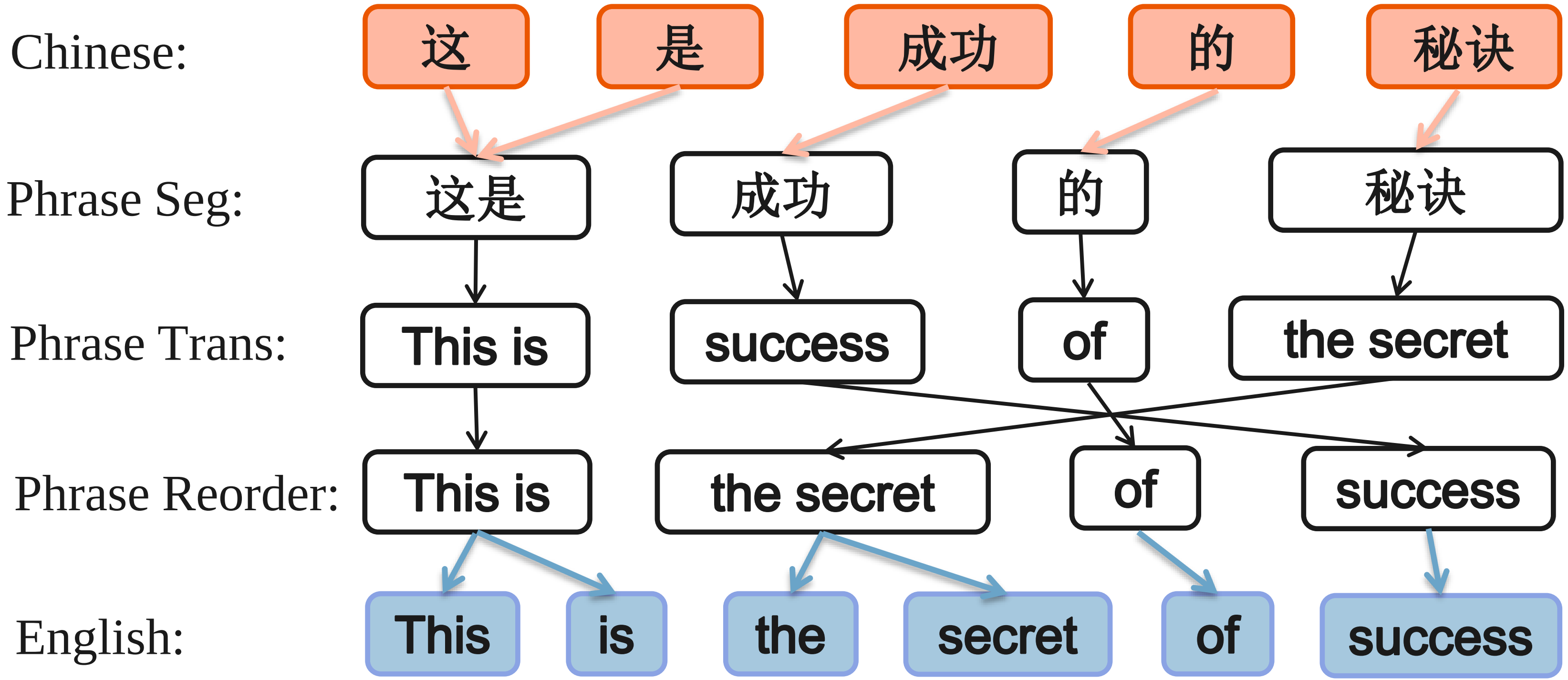


$$e = (这，是，成功，的，秘诀)$$

$$\log p(f|e) = \log p(e|f) + \log p(f)$$

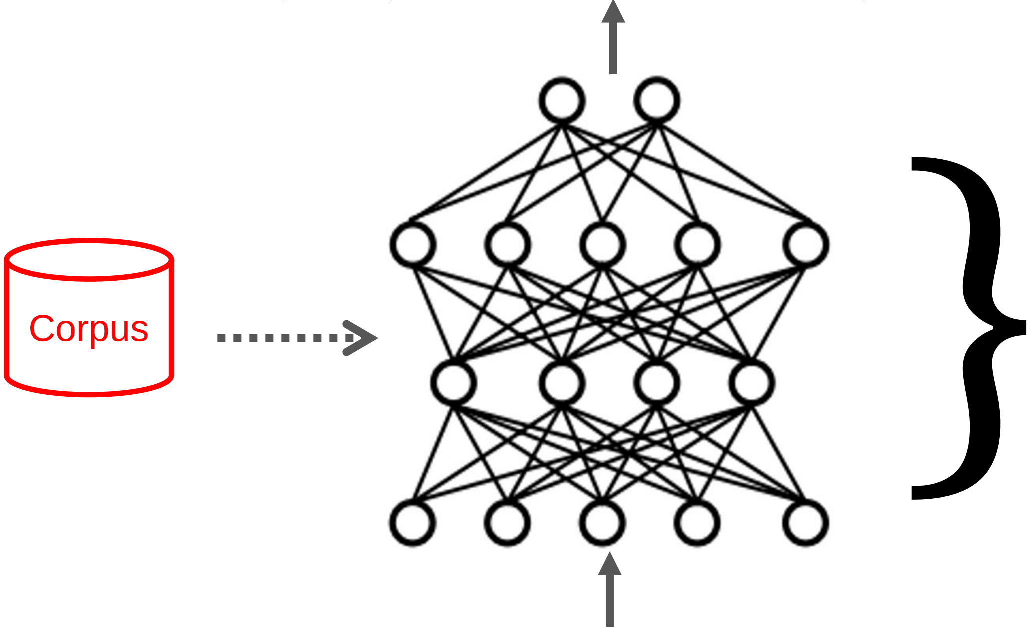
Translation Model Language Model

Statistical Machine Translation



Neural Machine Translation

$f = (\text{this},\ \ is,\ \ success,\ \ of,\ \ secret)$

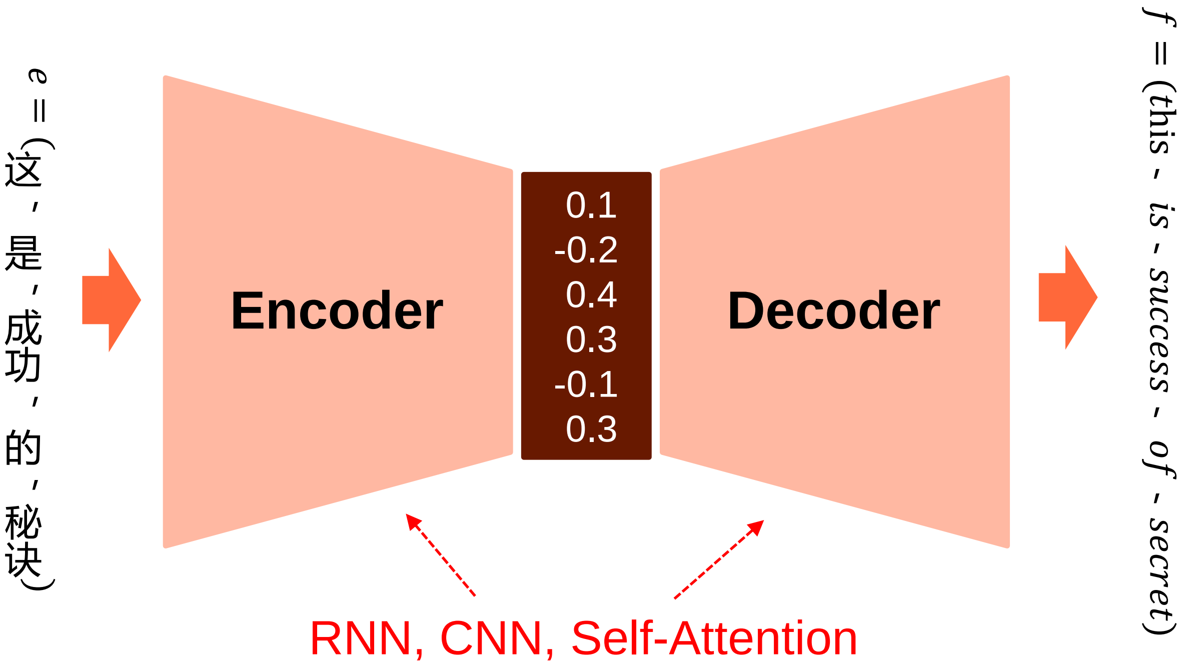


One big parametric approximation of

$\log p(f|e)$

$e =$ (这，是，成功，的，秘诀)

Encoder-Decoder Framework

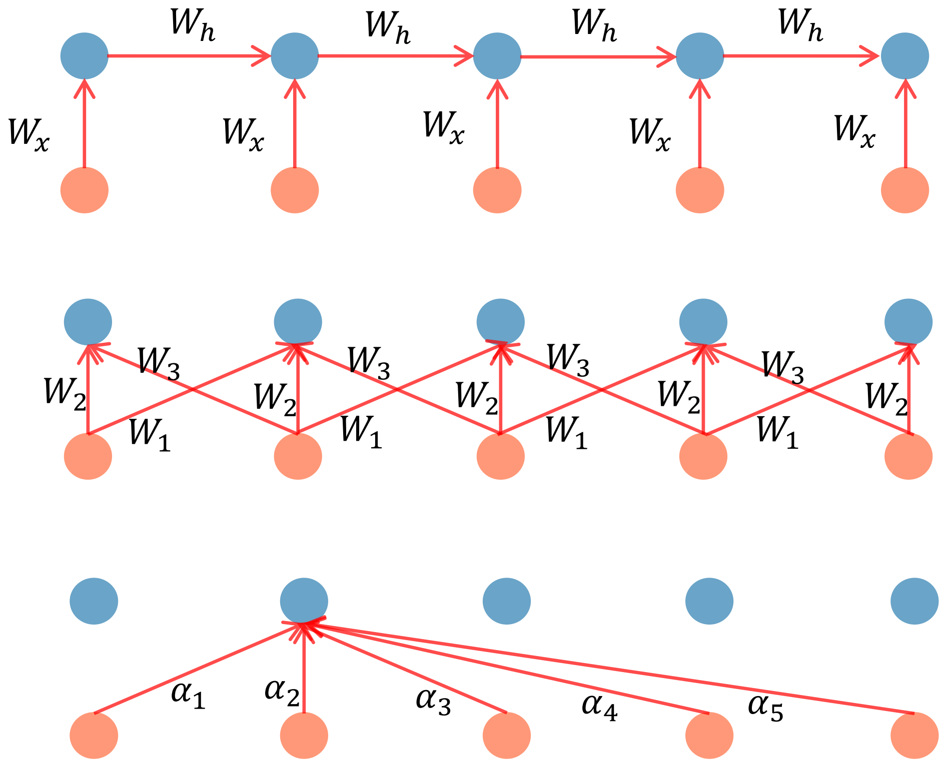


Build Unit

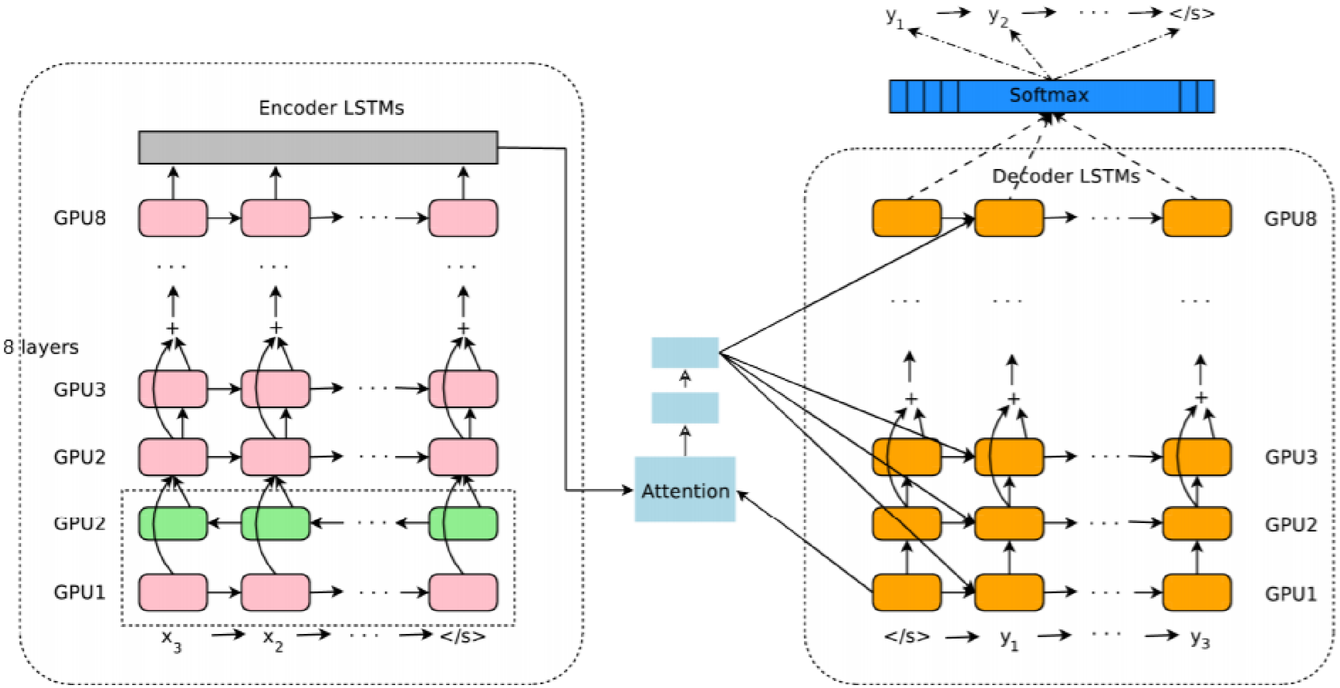
RNN:

CNN:

Self-Attention:



Google's Neural Machine Translation

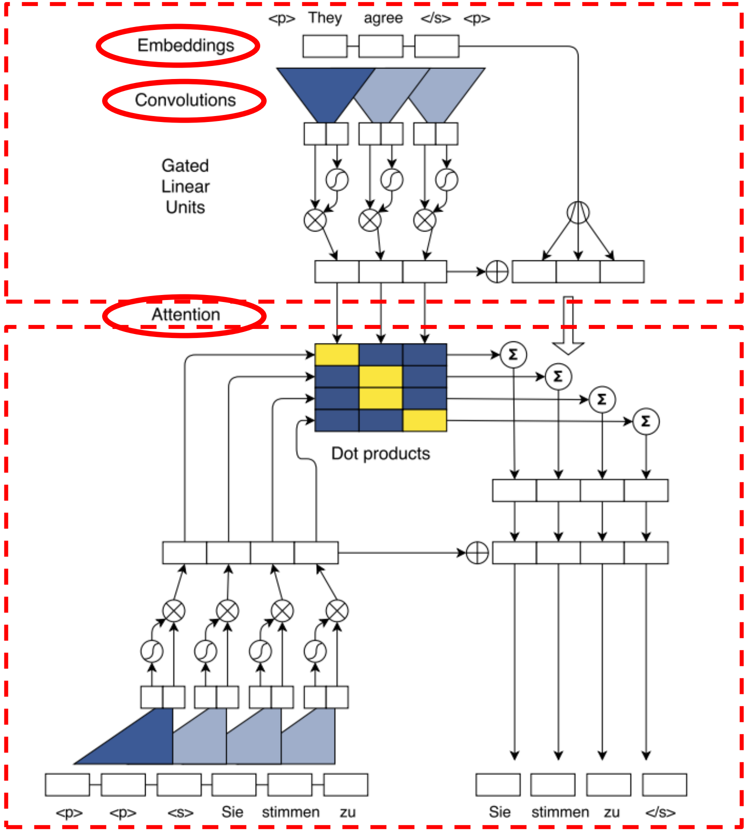


[Wu et al., 2016] Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. Arxiv.

Convolutional Neural Machine Translation

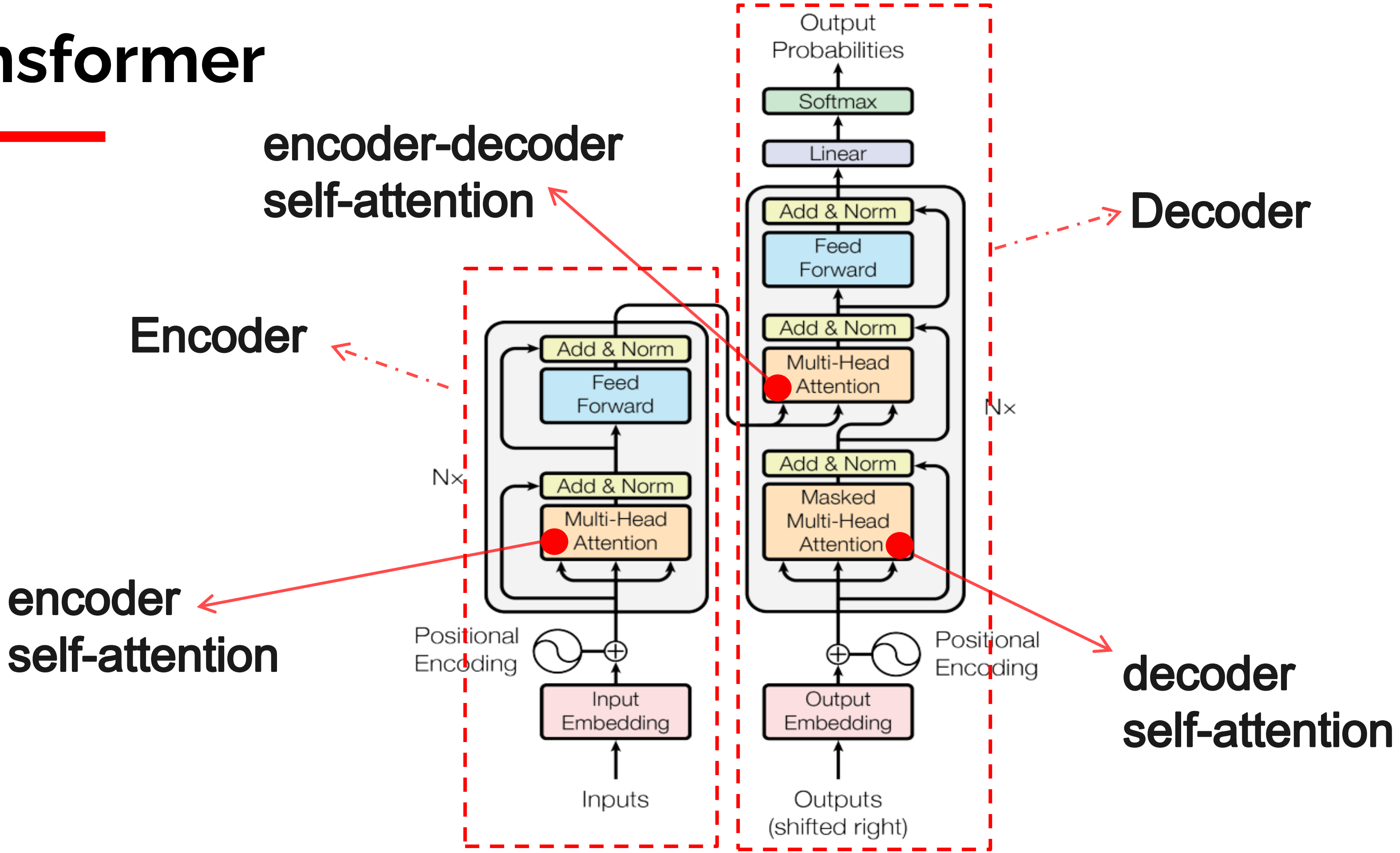
Encoder

Decoder



[Gehring et al., 2017] Convolutional Sequence to Sequence Learning. ICML.

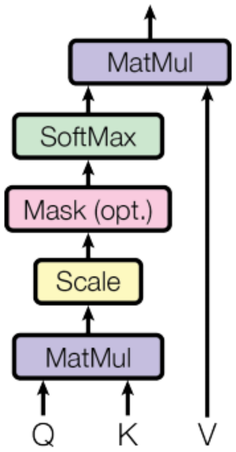
Transformer



[Vaswani et al., 2017] Attention is all you need. NIPS.

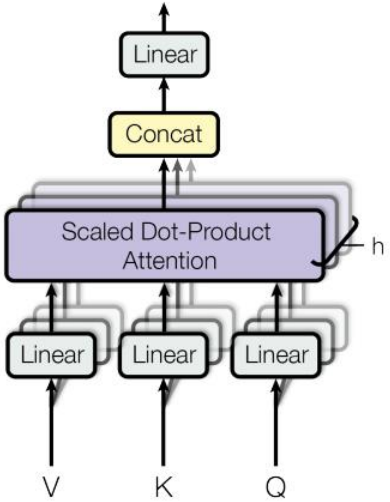
Self-Attention

Scaled Dot-Product Attention



$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

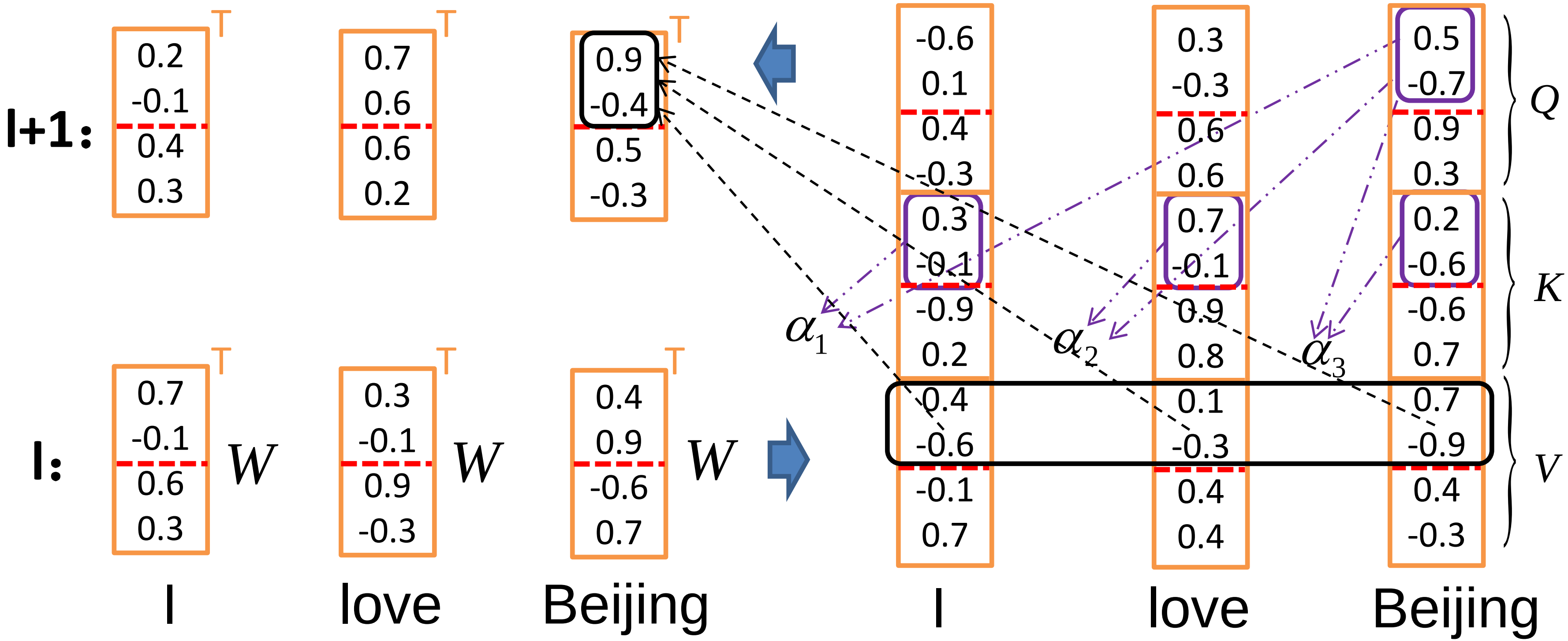
Multi-Head Attention



$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

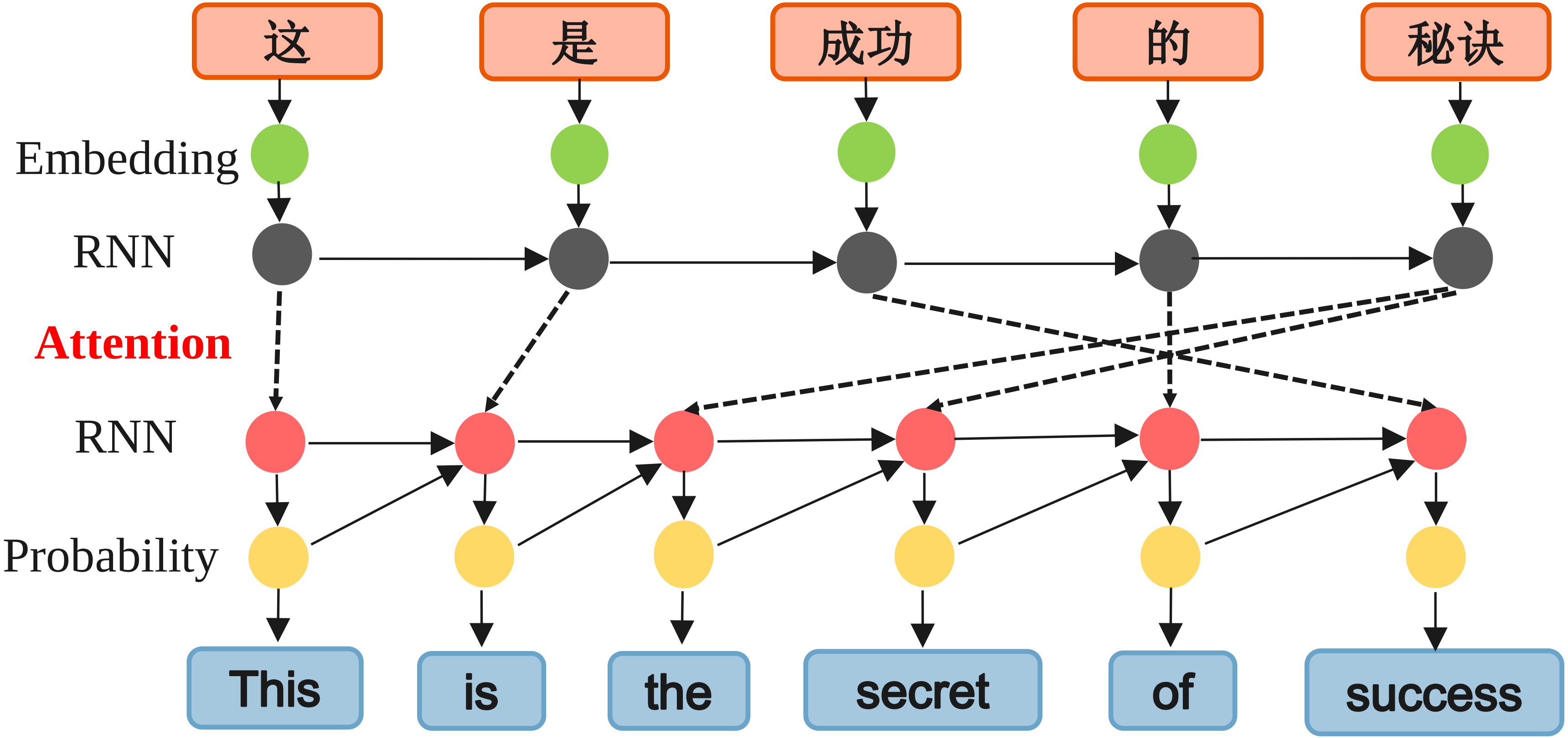
Self-Attention

- **Assume: hidden size=4, head=2, $W \in R^{4\times 12}$**

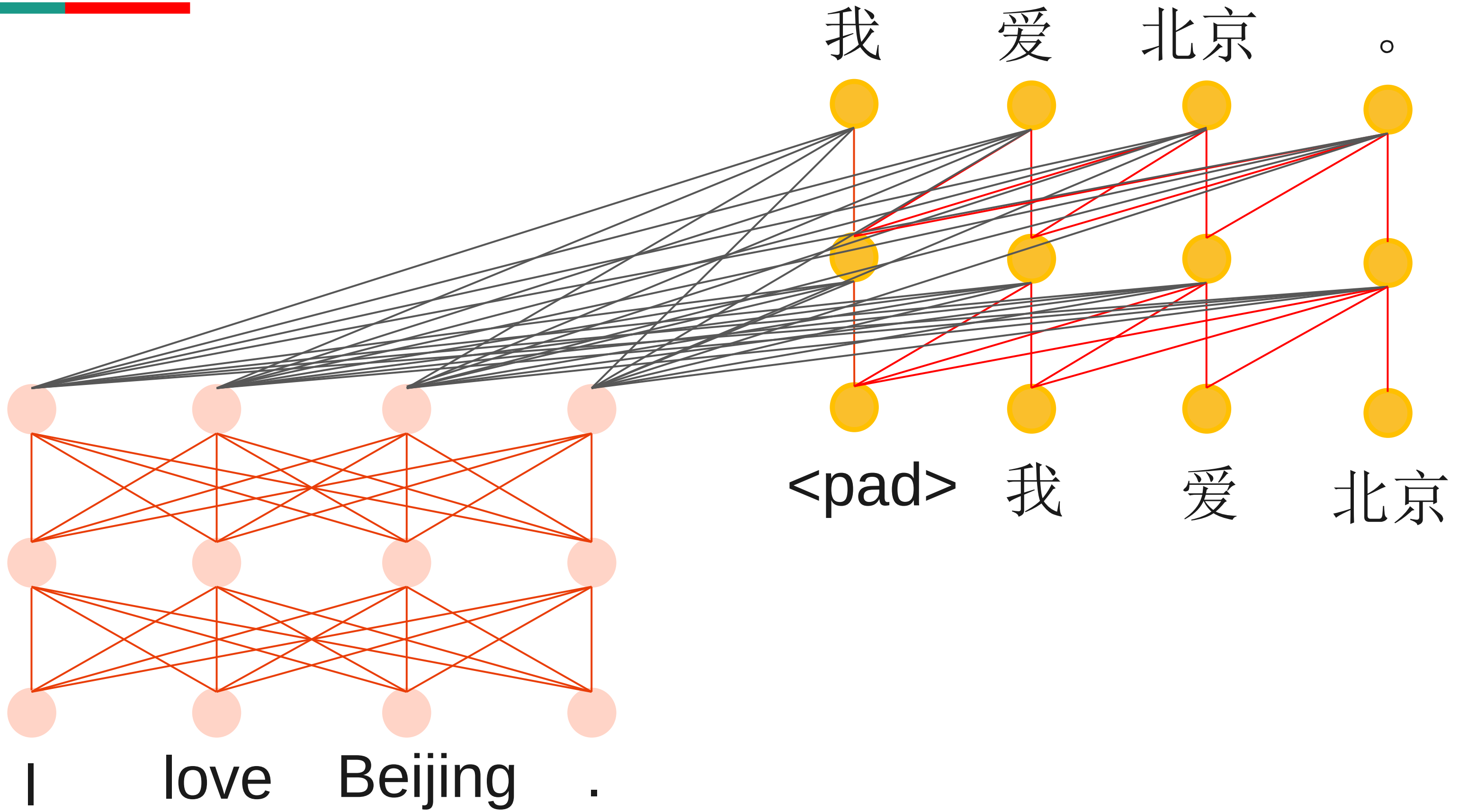


Section 2: Autoregressive Neural Machine Translation

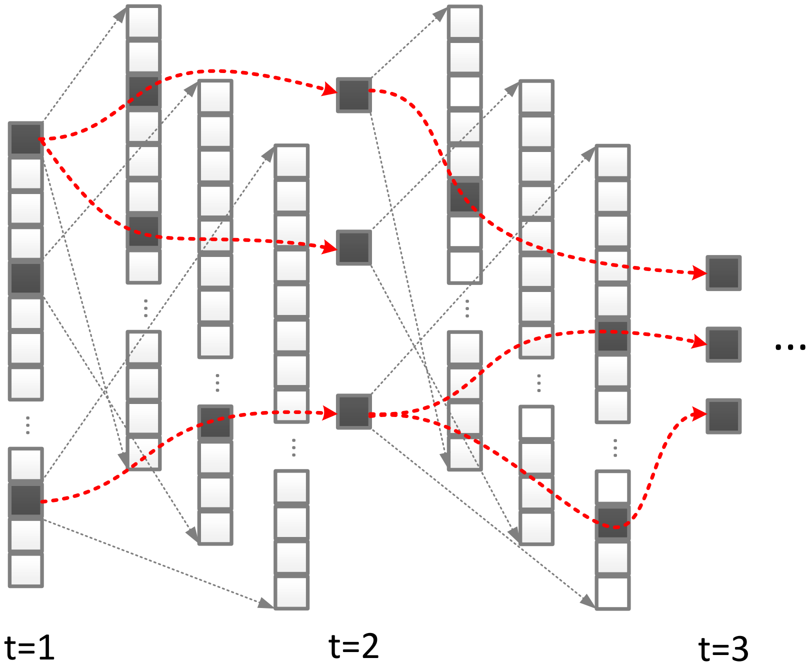
Inference: Autoregressive RNMT



Inference: Autoregressive Transformer



Inference (Beam Search)



Algorithm 1 Standard Beam Search.

Input: enc; dec; x, y;
Output: y;

define stack s
define set c
create initial hypo and put in into s[0]
i ← 0
N ← beam size
for $i = 1$ to max_len **do**
  for all $h \in s[i]$ **do**
    extend new hypos from h
    put new hypos into $s[i+1]$
  end for;
  prune $s[i+1]$ to keep N hypos
  move complete hypo in $s[i+1]$ to c
  if $len(c) > N$ **then**
    prune c to keep N hypos
  end if
  if $max\_point(s) < min\_point(c)$ **then**
    break
  end if
end for
y trace back from best $h \in c$

Autoregressive Generation Methods

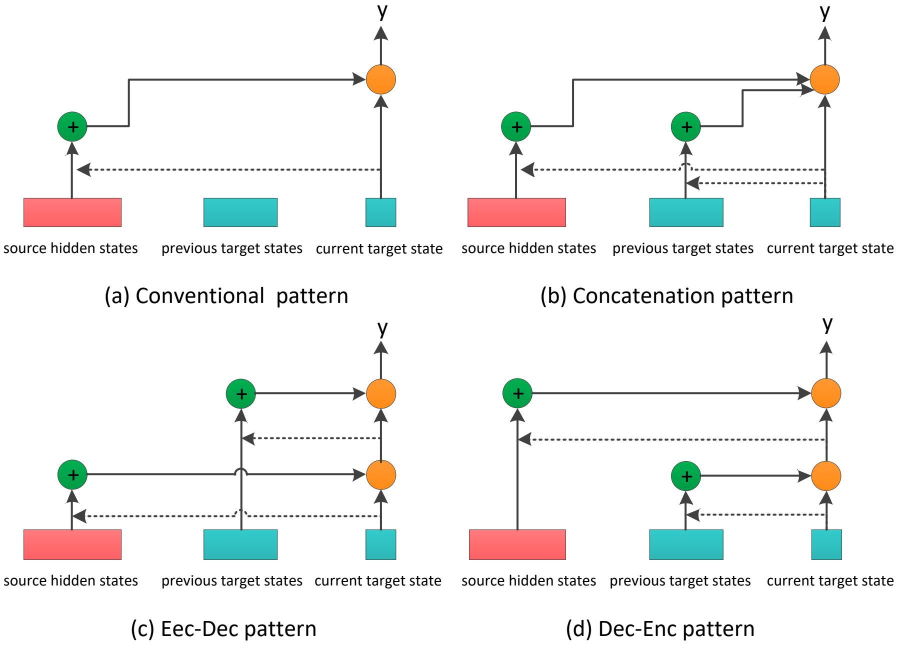
- History Enhanced Decoding
- Future Enhanced Decoding
- Constrained Decoding
- Memory Enhanced Decoding
- Structured Decoding
- Multi-Pass decoding
- Fast Decoding
- …

Autoregressive Neural Machine Translation

- **History Enhanced Decoding**
    - Look-ahead Attention for Generation in Neural Machine Translation. NLPCC 2017.
    - Sequence Generation with Target Attention. ECML PKDD 2017.
    - Self-Attentive Residual Decoder for Neural Machine Translation. NAACL 2018.
    - Neural Machine Translation with Decoding History Enhanced Attention. COLING 2018.

Autoregressive Neural Machine Translation

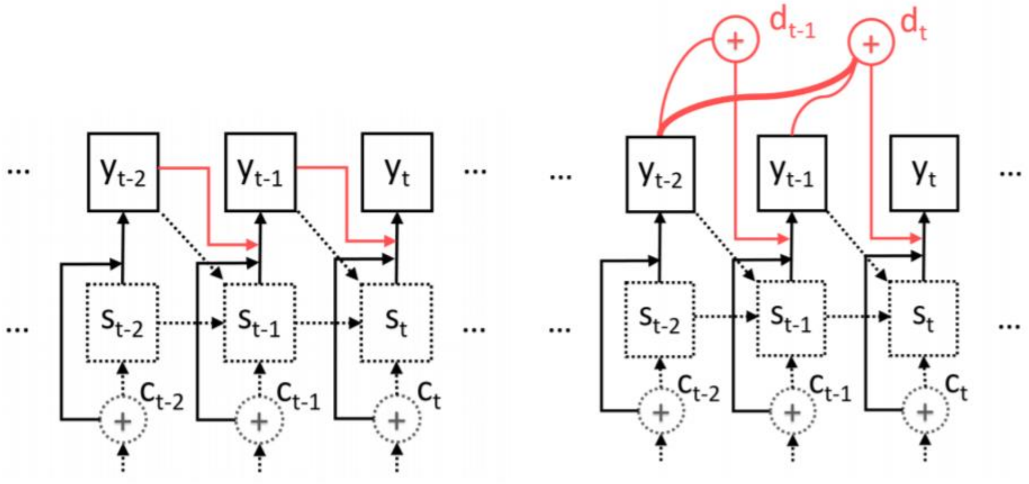
- ## History Enhanced Decoding



[Zhou et al., 2017] Look-ahead Attention for Generation in Neural Machine Translation. NLPCC.

Autoregressive Neural Machine Translation

- **History Enhanced Decoding**



(a) Baseline NMT decoder (b) Self-attentive residual dec.

[Werlen et al., 2018] Self-Attentive Residual Decoder for Neural Machine Translation. NAACL.

Autoregressive Neural Machine Translation

- **Future Enhanced Decoding**
  - Learning to Decode for Future Success. Arxiv 2017.
  - An actor-critic algorithm for sequence prediction. ICLR 2017.
  - Decoding with value networks for neural machine translation. NIPS 2017.
  - Modeling Past and Future for Neural Machine Translation. TACL 2018.
  - Target Foresight based Attention for Neural Machine Translation. NAACL 2018.
  - Future-Prediction-Based Model for Neural Machine Translation. Arxiv 2018.
  - Synchronous Bidirectional Neural Machine Translation. TACL 2019.
  - Dynamic Past and Future for Neural Machine Translation. EMNLP 2019.
  - Attending to Future Tokens for Bidirectional Sequence Generation. EMNLP 2019.
  - Modeling Future Cost for Neural Machine Translation. Arxiv 2020.

Autoregressive Neural Machine Translation

- **Future Enhanced Decoding**

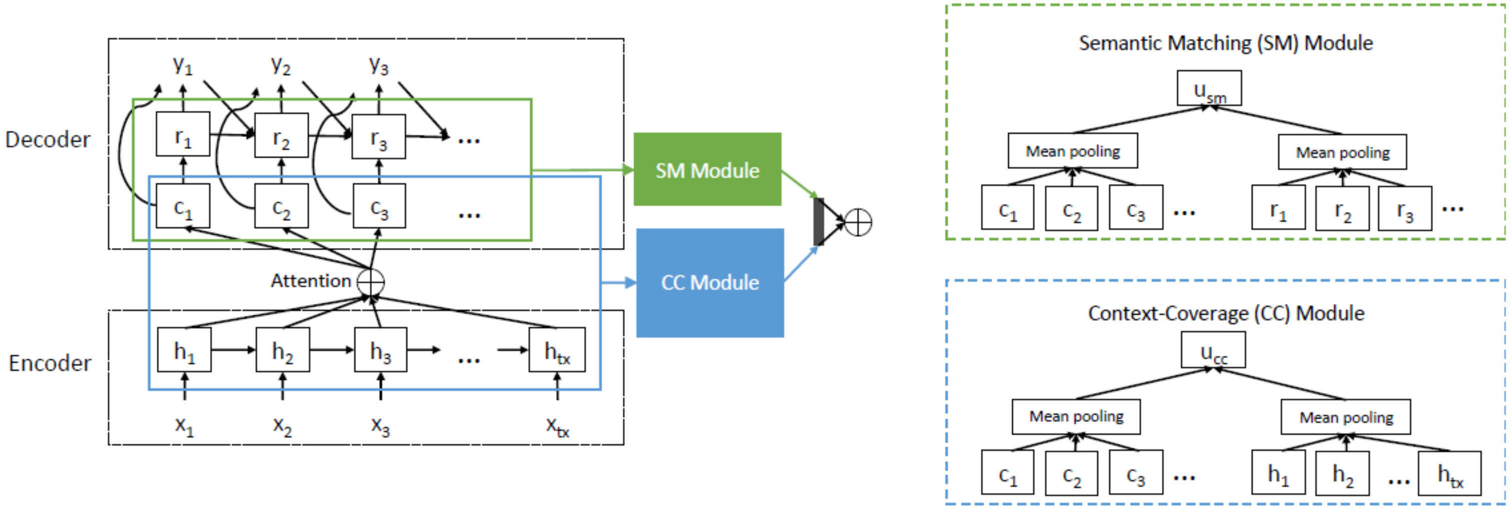


Figure: Architecture of Value Network

[He et al., 2017] Decoding with value networks for neural machine translation. NIPS.

Autoregressive Neural Machine Translation

- **Future Enhanced Decoding**

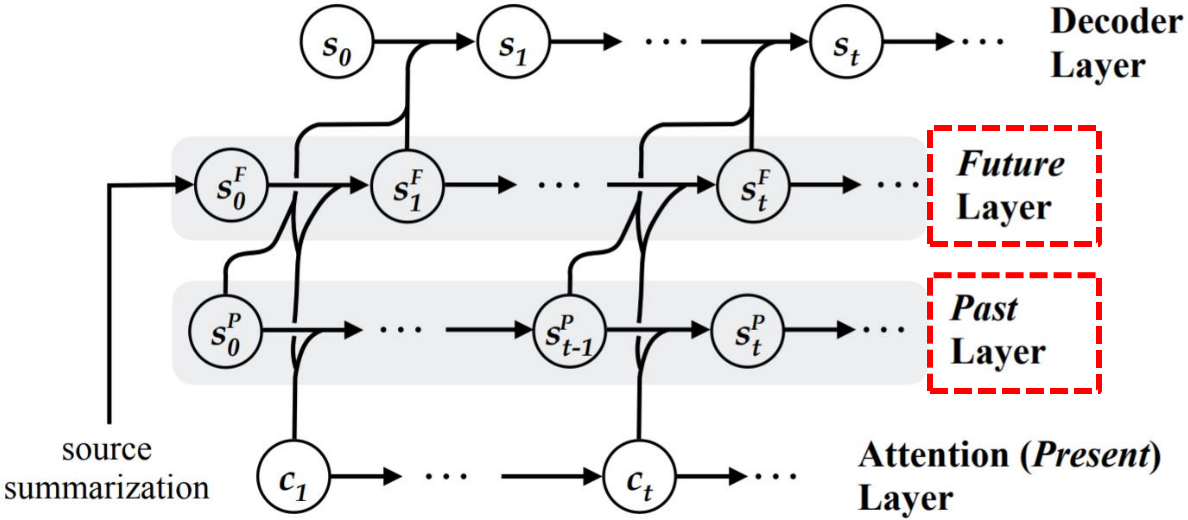


Figure: NMT decoder augmented with PAST and FUTURE layers.

[Zheng et al., 2018] Modeling Past and Future for Neural Machine Translation. TACL.

Autoregressive Neural Machine Translation

- **Memory Enhanced Decoding**
  - Memory-enhanced Decoder for Neural Machine Translation. EMNLP 2016.
  - Memory-augmented Neural Machine Translation. EMNLP 2017.
  - Encoding Gated Translation Memory into Neural Machine Translation. EMNLP 2018.
  - Phrase Table as Recommendation Memory for Neural Machine Translation. IJCAI 2018.
  - Neural Machine Translation with External Phrase Memory. Arxiv 2016
  - Guiding Neural Machine Translation with Retrieved Translation Pieces. NAACL 2018.
  - Learning to Remember Translation History with a Continuous Cache. TACL 2018.

Autoregressive Neural Machine Translation

- **Memory Enhanced Decoding**

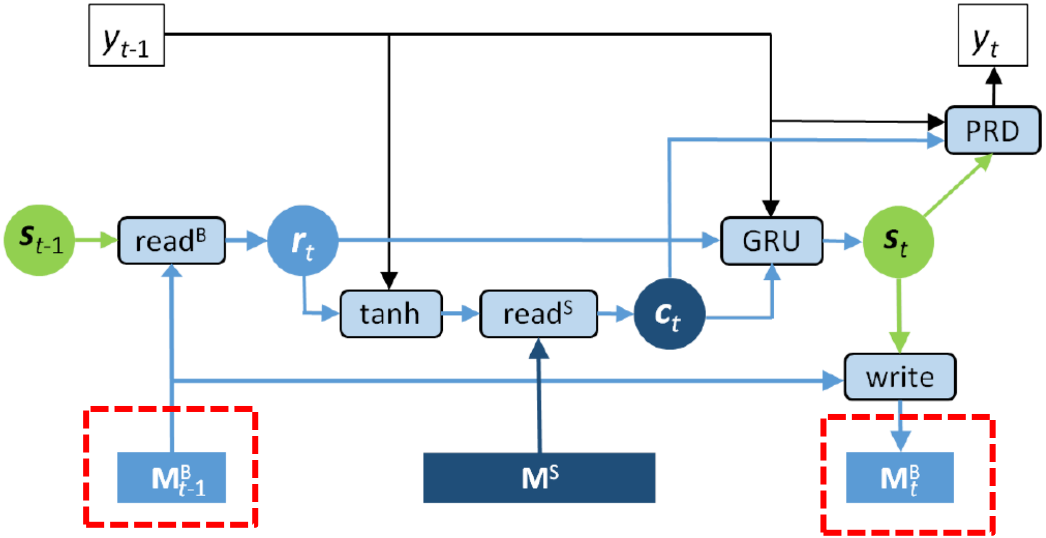


Figure: Diagram of the proposed memory-enhanced decoder

[Wang et al., 2016] Memory-enhanced Decoder for Neural Machine Translation. EMNLP.

Autoregressive Neural Machine Translation

- **Memory Enhanced Decoding**

Input: **jinkou dafu xiahua** shi **maoyi shuncha** zengzhang de zhuyao yuanyin

Reference: **the sharp decline in imports** was the main reason for the increase of the **trade surplus**

NMT: **import of imports** is the main reason for the growth in **trade**

SMT: **the sharp decline in imports** was mainly due to the growth of the **trade surplus**

Import of luxury cars
Import of foods
Import of vegetable
Import of toxic waste
…

trade deficit
trade surplus
trade with you
trade mask
…

Add bonus to words worthy of recommendation:

$$p(y_i \mid c_i, y_{<i}) = p(y_i \mid c_i, y_{<i}, R_i) = \mathrm{softmax}\left(\boxed{(1+\lambda V(R_i))}\,\mathrm{score}_i^N\right)$$

Attention weight   Phrase translation probability

$$V(R_i) = \begin{cases} \sum_{m=1}^{M} a_{(i,E_t)}^{m}\, p_{pht}^{m}(E_t \mid F_t) & \text{if } t \in R_i \\ 0 & \text{else} \end{cases}$$

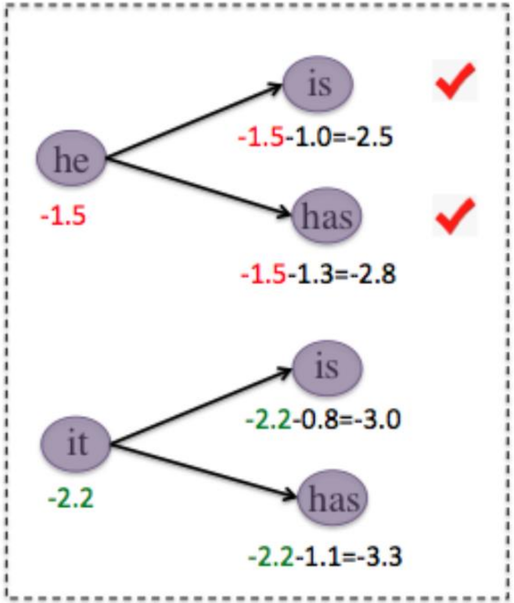
[Zhao et al., 2018] Phrase Table as Recommendation Memory for Neural Machine Translation. IJCAI.

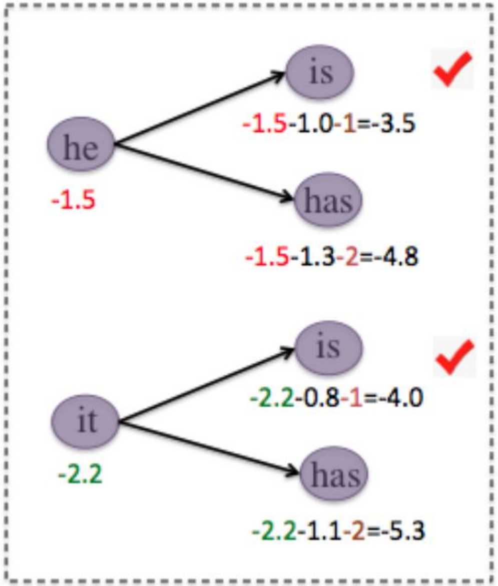
Autoregressive Neural Machine Translation

- **Constrained Decoding**
  - Mutual Information and Diverse Decoding Improve Neural Machine Translation. Arxiv 2016.
  - Neural Name Translation Improves Neural Machine Translation. CWMT 2018.
    Lexically Constrained Decoding for Sequence Generation Using Grid Beam Search. ACL 2017.
  - Neural Machine Translation Decoding with Terminology Constraints. NAACL 2018.
  - Fast Lexically Constrained Decoding with Dynamic Beam Allocation for Neural Machine Translation. NAACL 2018.
  - Improving Lexical Choice in Neural Machine Translation. NAACL 2018.
  - Sequence to Sequence Mixture Model for Diverse Machine Translation. CoNLL 2018.
  - Controlling Text Complexity in Neural Machine Translation. EMNLP 2019.
  - Generating Diverse Translation by Manipulating Multi-Head Attention. AAAI 2020.

Autoregressive Neural Machine Translation

- **Constrained Decoding**





[Li and Jurafsky, 2016] Mutual Information and Diverse Decoding Improve Neural Machine Translation. Arxiv.

Autoregressive Neural Machine Translation

- **Constrained Decoding**

Source:	Caroline went to Japan in April .
Target:	四月 卡罗琳 去 了 日本 。
Replaced Source:	PER1 went to LOC1 in April .
Replaced Target:	四月 PER1 去 了 LOC1 。

Table: Example of replacing entities with labels.

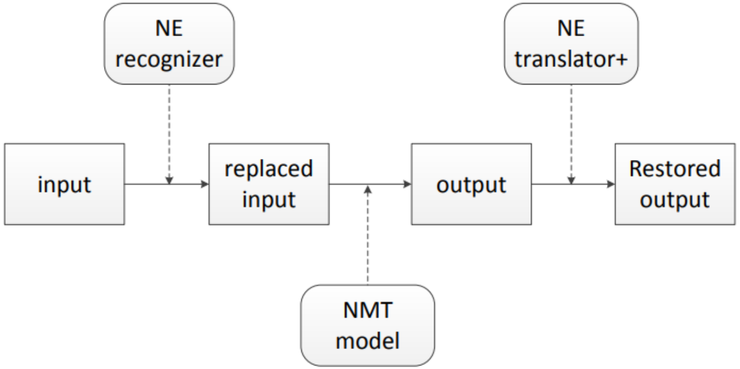


Figure: Replace-translate-restore framework

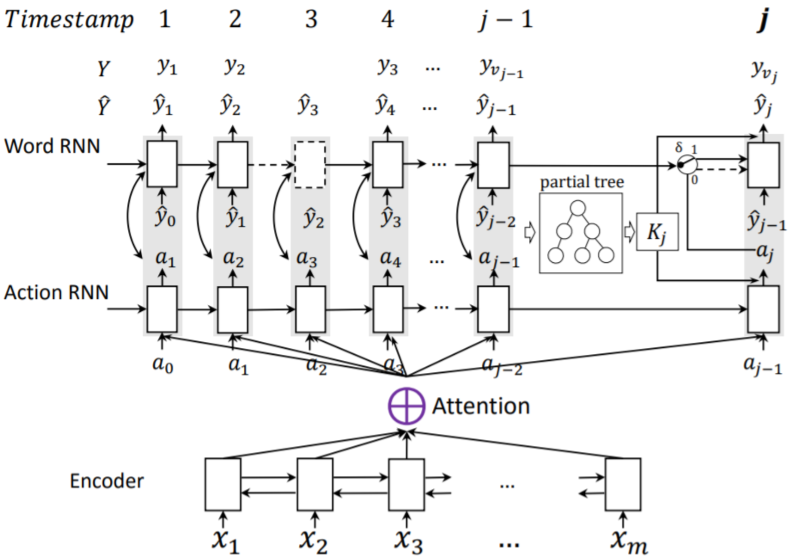
[Li et al., 2018] Neural Name Translation Improves Neural Machine Translation. CWMT.

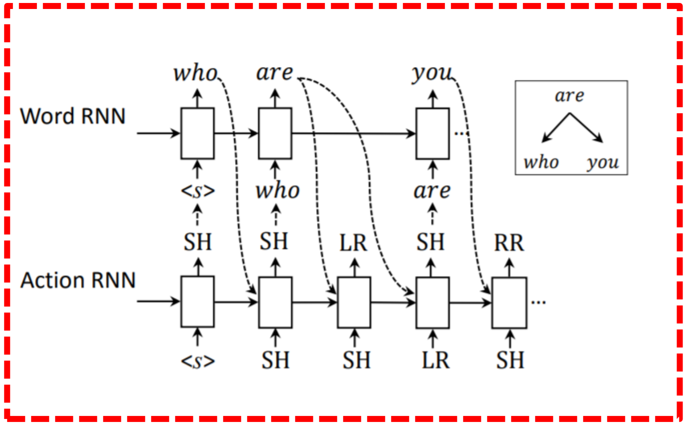
Autoregressive Neural Machine Translation

- **Structured Decoding**
  - Towards String-to-Tree Neural Machine Translation. ACL 2017.
  - Chunk-based Decoder for Neural Machine Translation. ACL 2017.
  - Chunk-Based Bi-Scale Decoder for Neural Machine Translation. ACL 2017.
  - Sequence-to-Dependency Neural Machine Translation. ACL 2017.
  - A Tree-based Decoder for Neural Machine Translation. EMNLP 2018.
  - Top-down Tree Structured Decoding with Syntactic Connections for Neural Machine Translation and Parsing. EMNLP 2018.
  - A Tree-based Decoder for Neural Machine Translation. EMNLP 2018.
  - Forest-Based Neural Machine Translation. EMNLP 2018.
  - Tree-to-tree Neural Networks for Program Translation. NeurIPS 2018.

Autoregressive Neural Machine Translation

- **Structured Decoding**

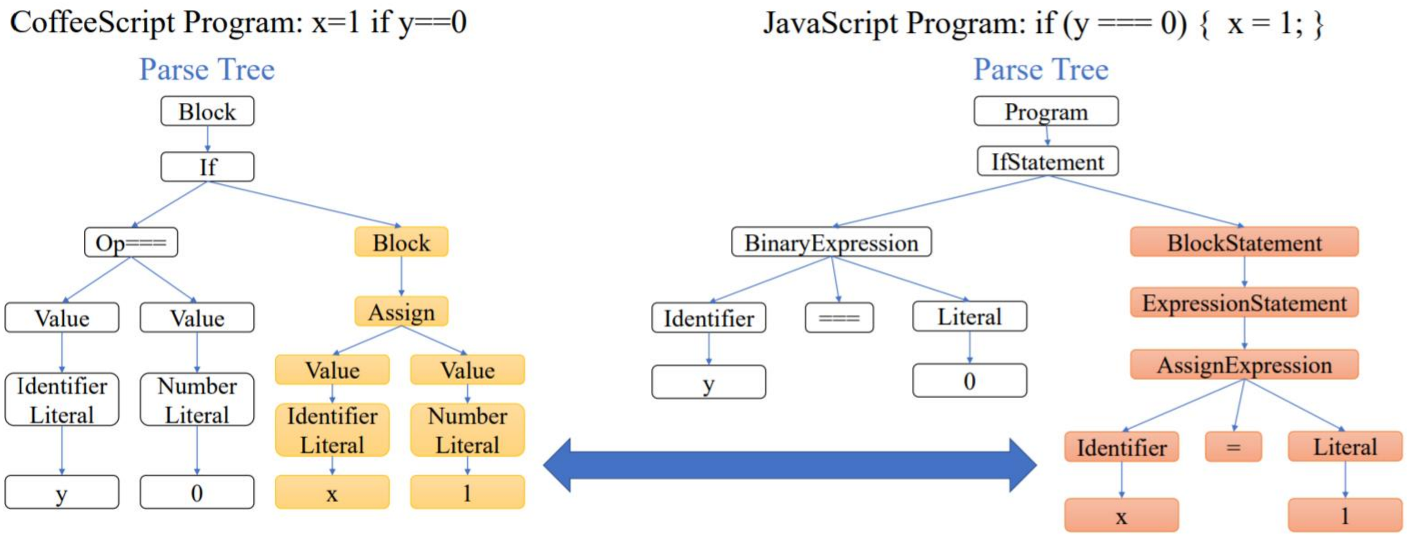




[Wu et al., 2017] Sequence-to-Dependency Neural Machine Translation. ACL.

Autoregressive Neural Machine Translation

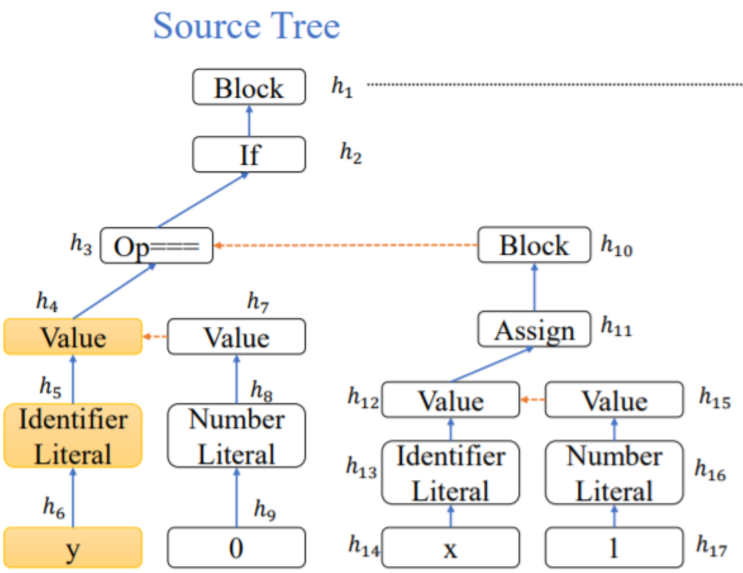
- **Structured Decoding**

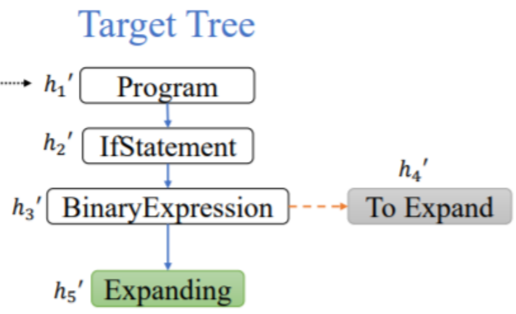


[Chen et al., 2018] Tree-to-tree Neural Networks for Program Translation. NeurIPS.

Autoregressive Neural Machine Translation

- **Structured Decoding**





Attention map: $w_i \propto \exp(h_i^T W_0 h_5')$

Source embedding: $e_s = \sum_{i=1}^{17} w_i h_i = [h_1; \ldots; h_{17}]w$

Combined embedding: $e_t = \tanh(W_1 e_s + W_2 h_5')$

Predicting the node: node $=$ argmax $\mathbf{softmax}(We_t)$

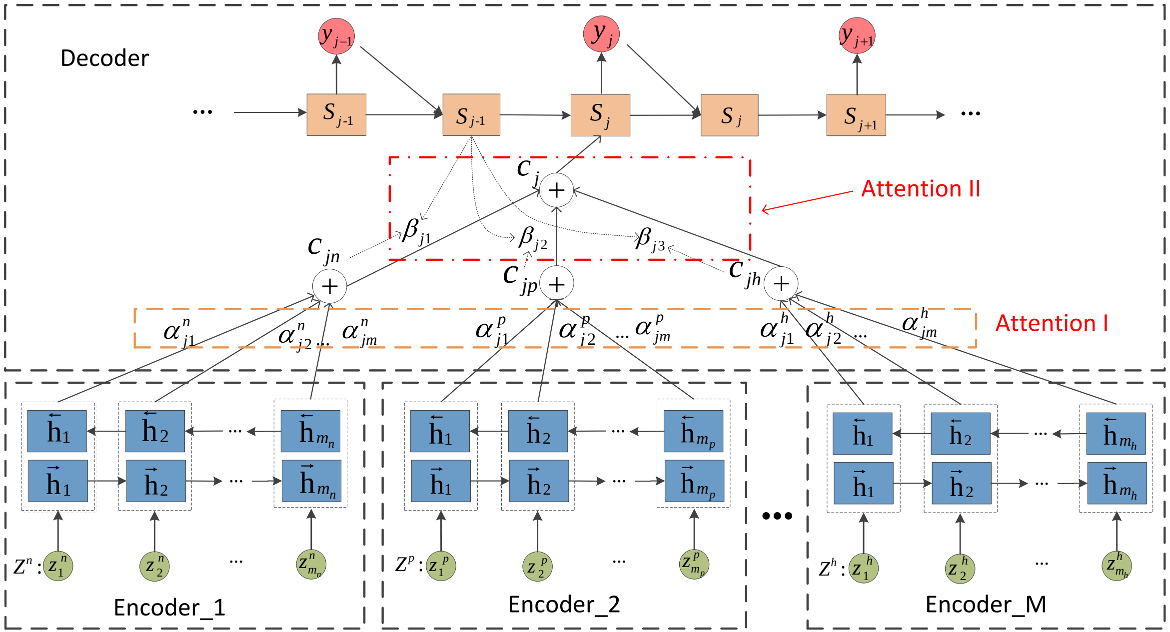
[Chen et al., 2018] Tree-to-tree Neural Networks for Program Translation. NeurIPS.

Autoregressive Neural Machine Translation

- **Multi-Pass Decoding**
  - Pre-Translation for Neural Machine Translation. CoLING 2016.
  - Neural System Combination for Machine Translation. ACL 2017.
  - Deliberation Networks: Sequence Generation Beyond One-Pass Decoding. NeurIPS 2017.
  - Asynchronous Bidirectional Decoding for Neural Machine Translation. AAAI 2018.
  - Adaptive Multi-pass Decoder for Neural Machine Translation. EMNLP 2018.

Autoregressive Neural Machine Translation

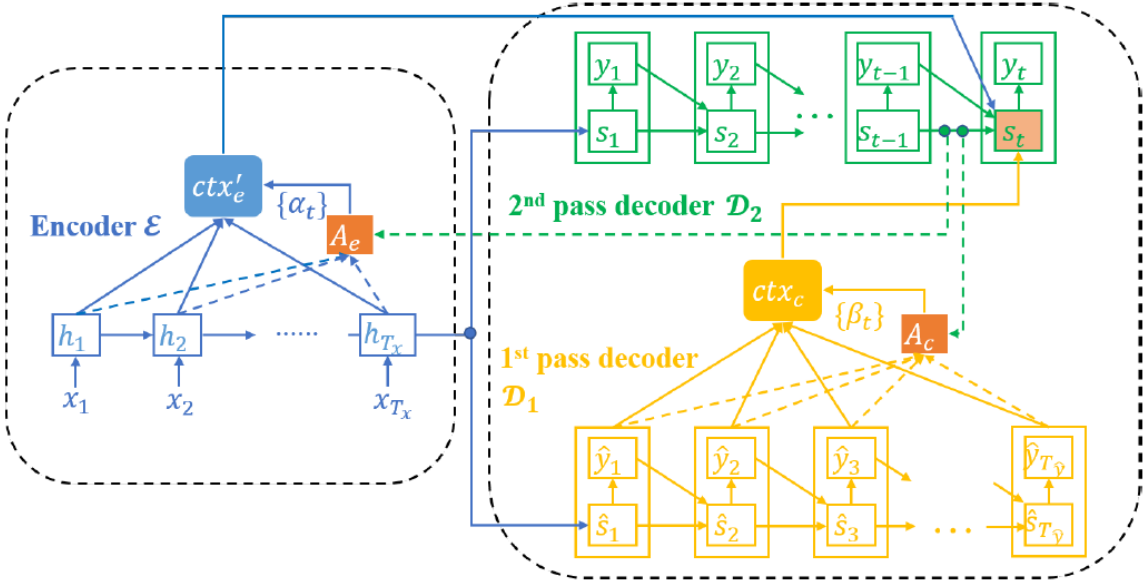
- **Multi-Pass Decoding**



[Zhou et al., 2017] Neural System Combination for Machine Translation. ACL.

Autoregressive Neural Machine Translation

- **Multi-Pass Decoding**



[Xia et al., 2018] Deliberation Networks: Sequence Generation Beyond One-Pass Decoding. NeurIPS.

Autoregressive Neural Machine Translation

- **Fast Decoding**
  - Vocabulary Manipulation for Neural Machine Translation. EMNLP 2016.
  - Speeding Up Neural Machine Translation Decoding by Shrinking Run-time Vocabulary. ACL 2017.
  - Towards Compact and Fast Neural Machine Translation Using a Combined Method. EMNLP 2017.
  - Sharp Models on Dull Hardware: Fast and Accurate Neural Machine Translation Decoding on the CPU. EMNLP 2017
  - Accelerating Neural Transformer via an Average Attention Network. ACL 2018.
  - Speeding Up Neural Machine Translation Decoding by Cube Pruning. EMNLP 2018.
  - Train Large, Then Compress: Rethinking Model Size for Efficient Training and Inference of Transformers. ICML 2020.

Autoregressive Neural Machine Translation

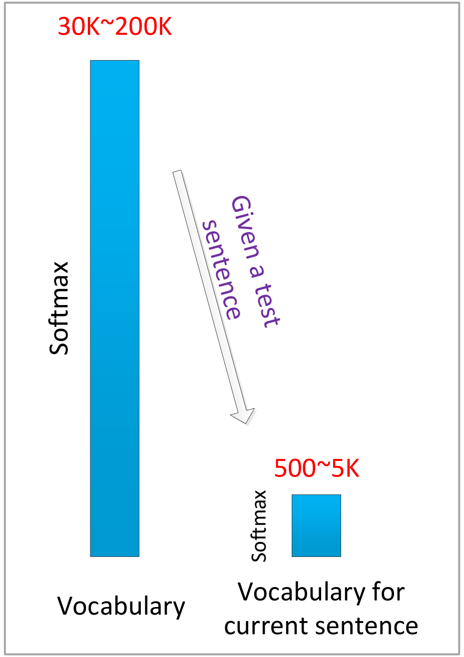
- **Fast Decoding**

$$V_{\mathbf{x}}^D = \bigcup_{i=1}^{l} D(x_i)$$

$$V_{\mathbf{x}}^P = \bigcup_{\forall x_i \ldots x_j \in subseq(\mathbf{x})} P(x_i \ldots x_j),$$

$$V_{\mathbf{x}}^T = T(n).$$

$$V_o = V_{\mathbf{x}} = V_{\mathbf{x}}^D \cup V_{\mathbf{x}}^P \cup V_{\mathbf{x}}^T,$$



[Mi et al., 2016] Vocabulary Manipulation for Neural Machine Translation. EMNLP.

Autoregressive Neural Machine Translation

- **Fast Decoding**

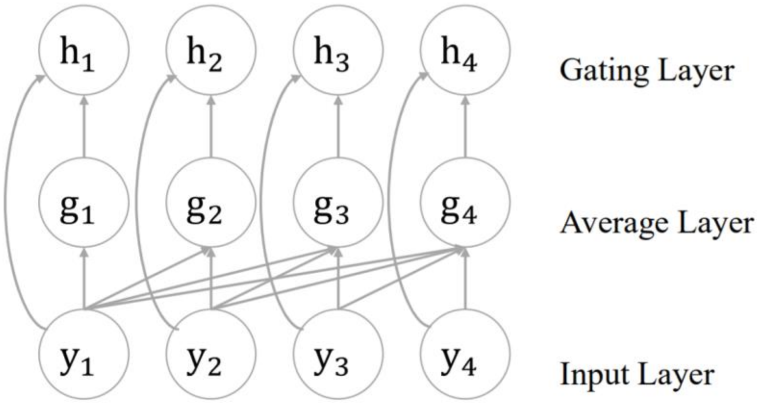


Figure: Visualization of average attention network

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Training:

$$\mathbf{g}_j = \mathrm{FFN}\left(\frac{1}{j}\sum_{k=1}^{j}\mathbf{y}_k\right)$$

Inference:

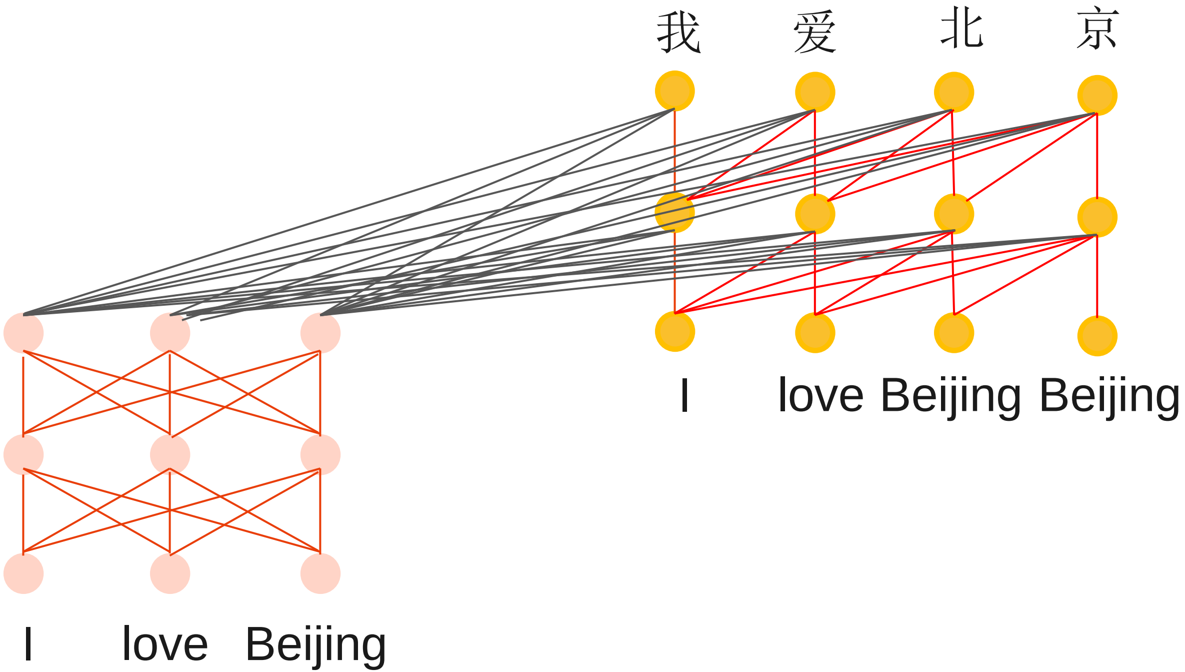
$$\tilde{\mathbf{g}}_j = \tilde{\mathbf{g}}_{j-1} + \mathbf{y}_j$$

$$\mathbf{g}_j = \mathrm{FFN}\left(\frac{\tilde{\mathbf{g}}_j}{j}\right)$$

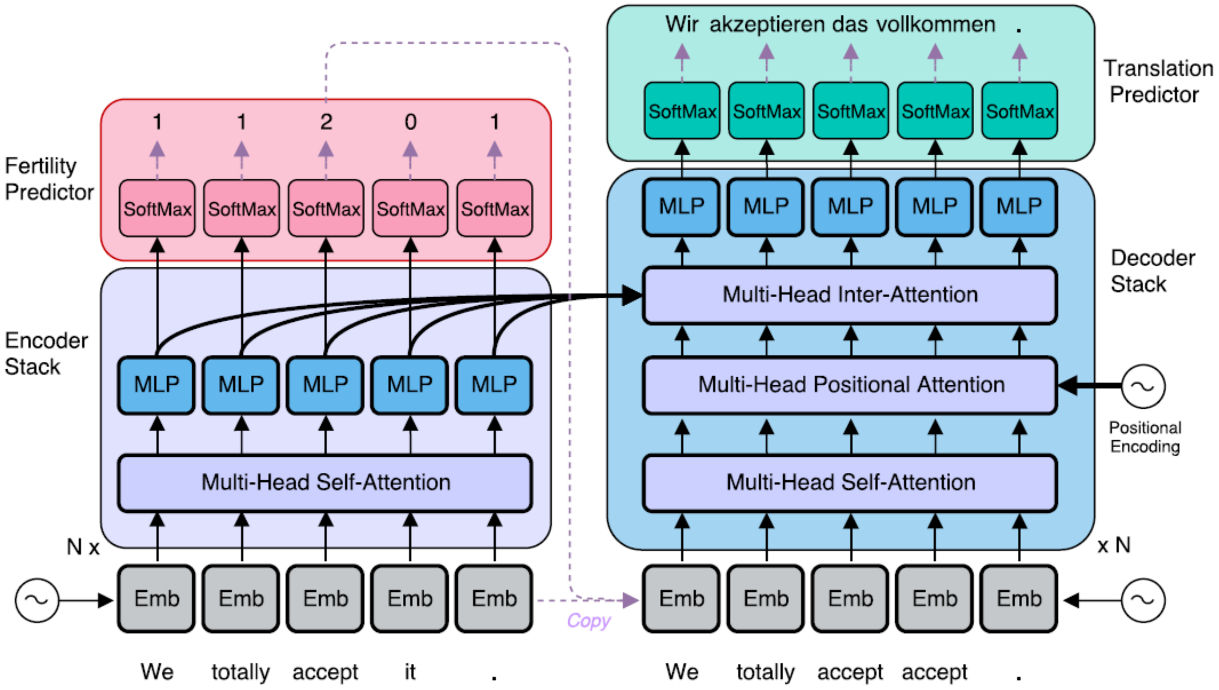
[Zhang et al., 2019] Accelerating Neural Transformer via an Average Attention Network. ACL.

Section 3: Non-Autoregressive Neural Machine Translation

Non-Autoregressive Generation



Non-Autoregressive Translation



[Gu et al., 2017] Non-autoregressive Neural Machine Translation. ICLR.

Non-Autoregressive NMT

- **Key Points**
  - Leverage knowledge distillation
- **Challenges**
  - Determine output length
  - Enhance decoder input
  - Model target dependency
- **Problems**
  - Multi-modality problems
  - Under-translation & repeat-translation

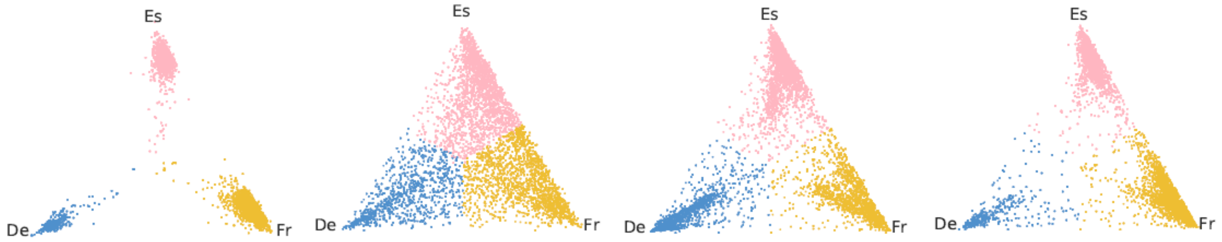
Key Points

- **Sequence-Level Knowledge Distillation**
  - Training an autoregressive NMT model (Teacher)
  - Obtaining target sentences by translating source languages with teacher model.
  - Using translated outputs as the targets and training the non-autoregressive NMT model (Student).

[Kim and Rush, 2016] Sequence-Level Knowledge Distillation. EMNLP.

Key Points

- **Understand Knowledge Distillation**



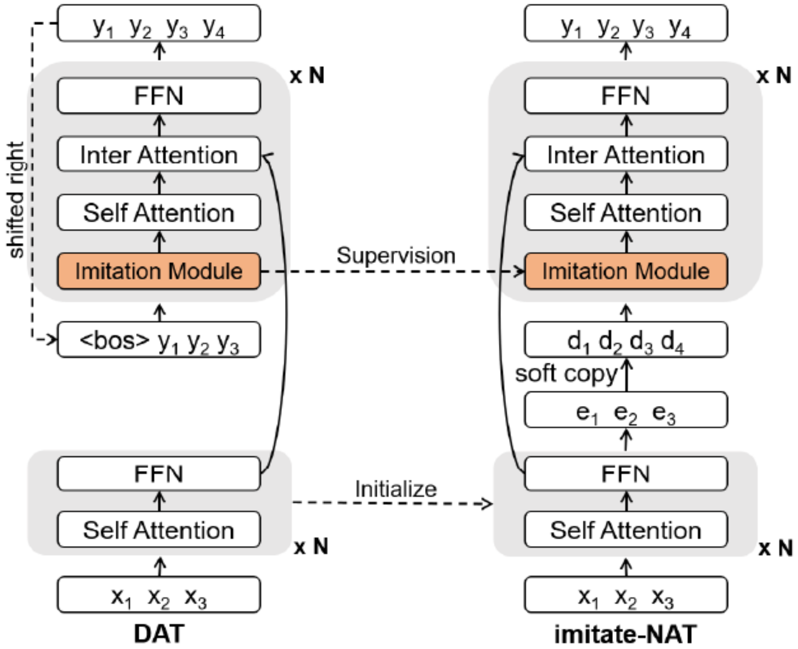
(a) AT Baseline (b) NAT Baseline (c) NAT Random Select (d) NAT Distill

- **Improvements to Knowledge Distillation**
  - Born-Again Networks
  - Mixture-of-Experts
  - Sequence-Level Interpolation

[Zhou, 2020] Understanding knowledge distillation in non-autoregressive machine translation. ICLR.

Key Points

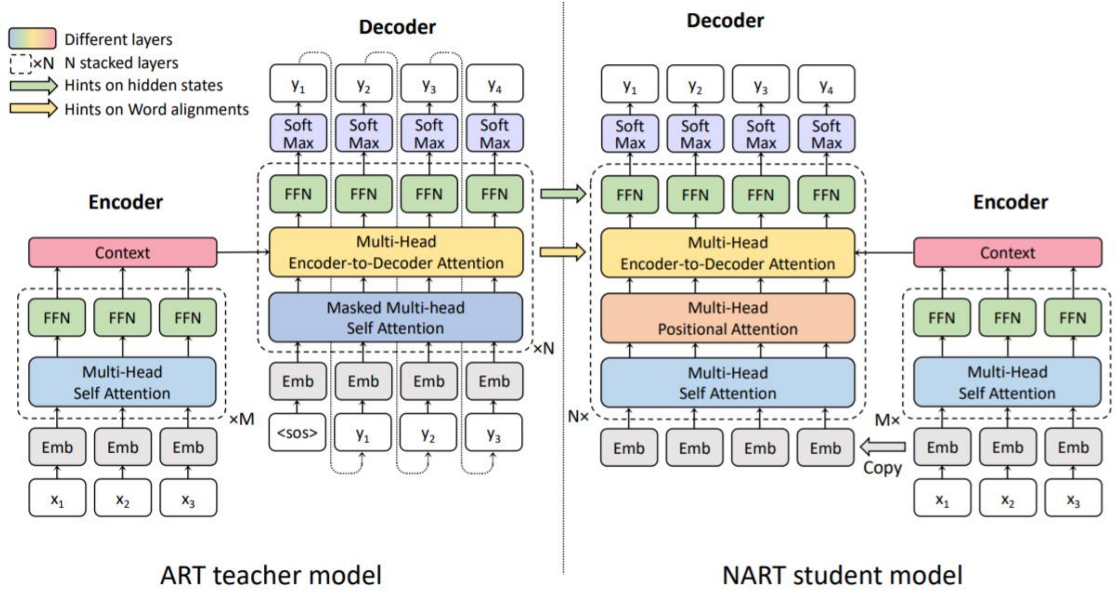
- **Knowledge Distillation**
  - Imitation Learning



[Wei et al., 2019] Imitation Learning for Non-Autoregressive Neural Machine Translation. ACL.

Key Points

- **Knowledge Distillation**
  - Hint-Based Training

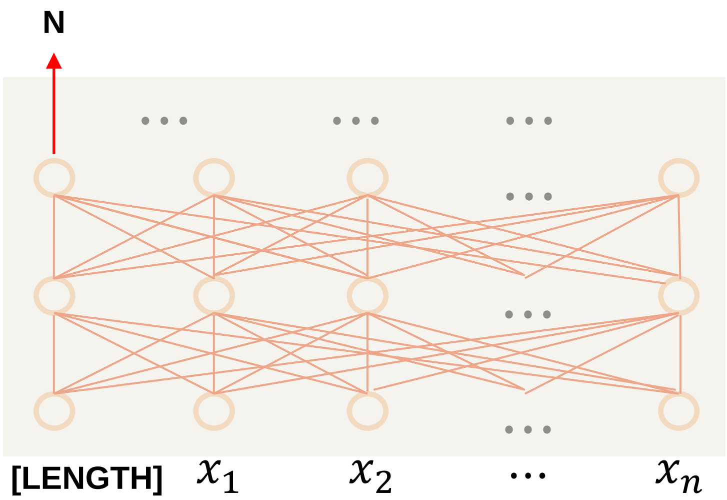


[Li et al., 2019] Hint-Based Training for Non-Autoregressive Machine Translation. EMNLP.

Challenges

- **Determine Output length**
  - Fertilities mechanism
  - Multiply/add length ratio
  - Directly predict length
  - Use multiple lengths

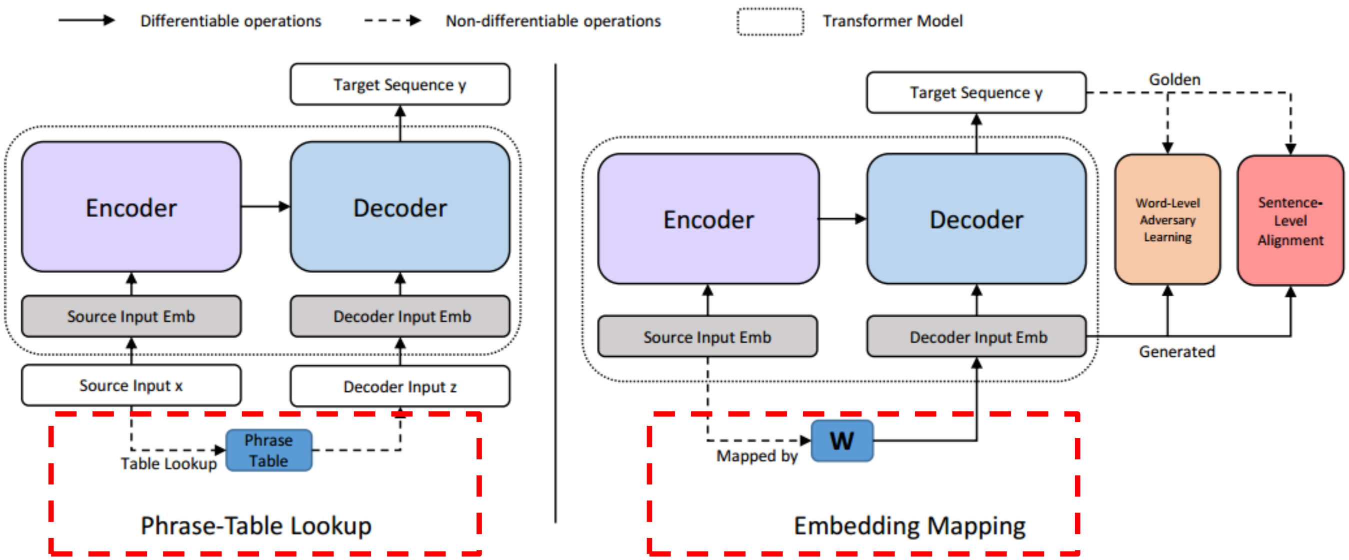
1.Draw samples from the fertility space
2. $T_y \in [T_x + \Delta T - B, T_x + \Delta T + B]$
3. $T_y \in [\alpha \cdot T_x] - B, \alpha \cdot T_x + B]$



[Ghazvininejad et al., 2019] Mask-Predict: Parallel Decoding of Conditional Masked Language Models. EMNLP.

Challenges

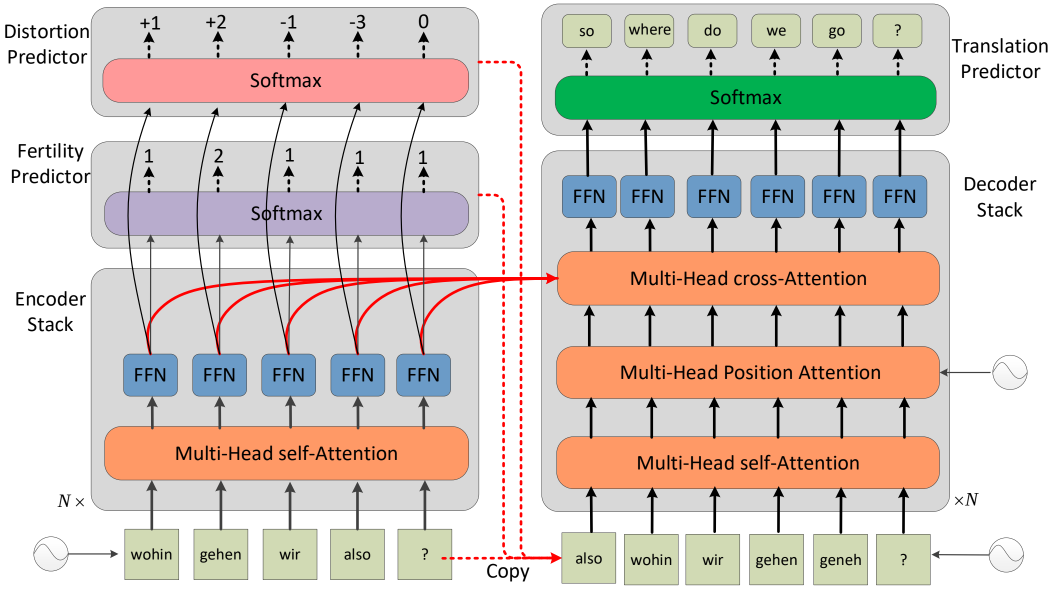
- **Enhance Decoder Input**



[Wang et al., 2019] Non-Autoregressive Machine Translation with Enhanced Decoder Input. AAAI.

Challenges

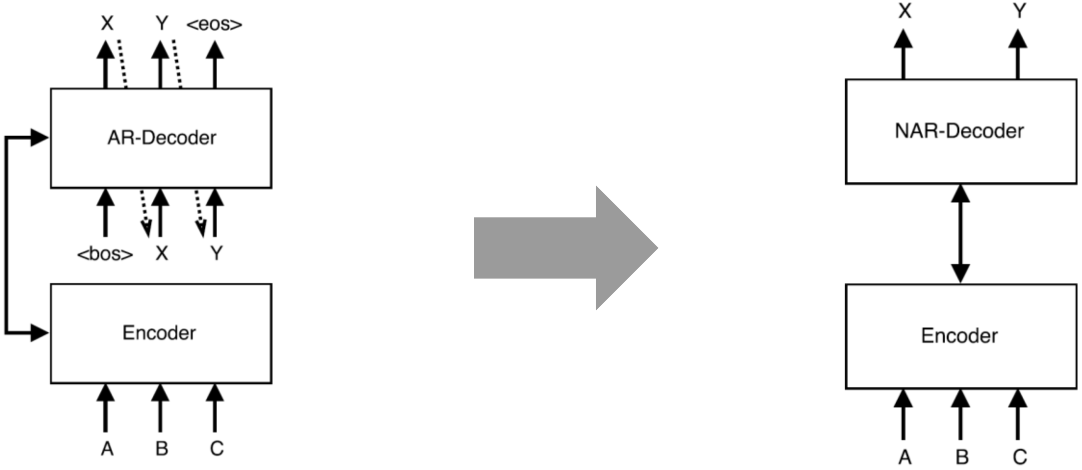
- **Enhance Decoder Input**



[Zhou et al., 2020] Non-Autoregressive Machine Translation with Distortion Model. NLPCC.

Challenges

- **Enhance Target Dependency**

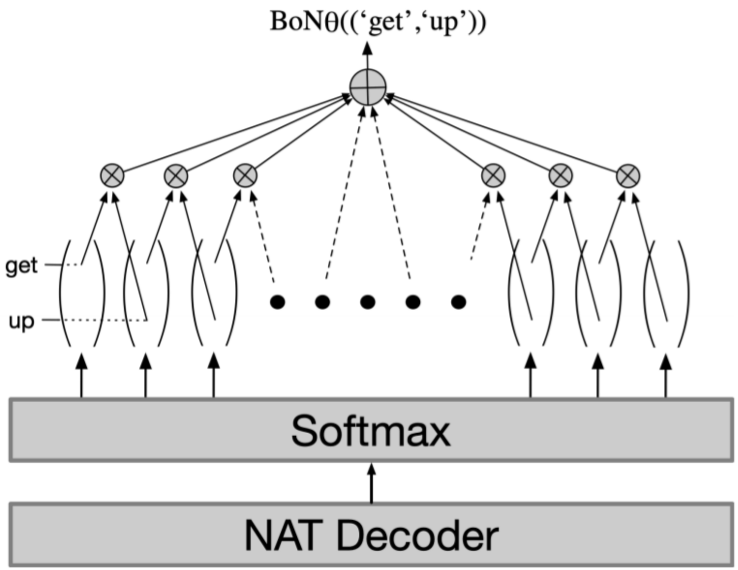


$$p_{\mathcal{AR}}(Y|X;\theta) = \prod_{t=1}^{T+1} p(y_t|y_{0:t-1}, x_{1:T'};\theta)$$

$$p_{\mathcal{NA}}(Y|X;\theta) = p_L(T|x_{1:T'};\theta) \cdot \prod_{t=1}^{T} p(y_t|x_{1:T'};\theta).$$

Challenges

- **Enhance Target Dependency**



[Shao et al., 2020] Minimizing the Bag-of-Ngrams Difference for Non-Autoregressive Neural Machine Translation. AAAI.

Challenges

- **Enhance Target Dependency**

Target Y	*it*	*tastes*	*pretty*	*good*	*though*
Alignment $\alpha : Y \rightarrow P$	2	3	3	4	5
Model Predictions P **(Top 5)**	but	it	tastes	delicious	ε
	however	ε	makes	good	.
	ε	that	looks	tasty	,
	for	this	taste	fine	so
	and	for	feels	exquisite	though

Figure: An example illustrating how AXE aligns model predictions with targets.

[Ghazvininejad et al., 2020] Aligned Cross Entropy for Non-Autoregressive Machine Translation. ICML.

Problems

- **Multi-Modality Problems**

Source: 大幅下降

Target: decline sharply
sharp decrease

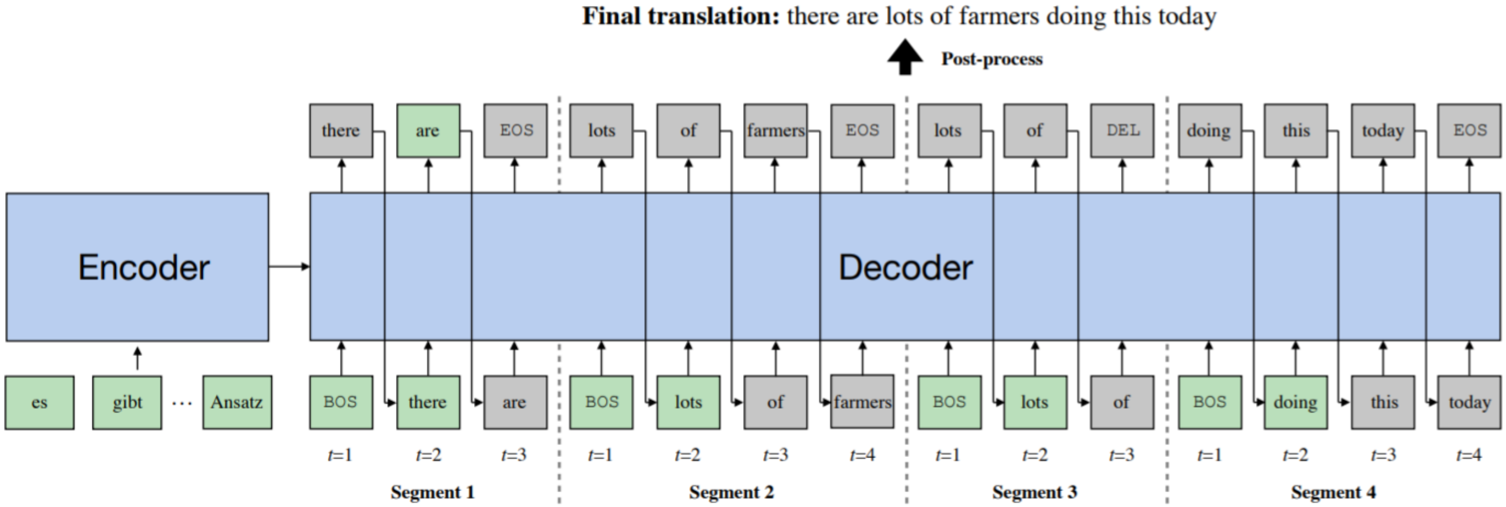
Error: decline decrease
Sharp sharply

Src.	es gibt heute viele Farmer mit diesem Ansatz
Feasible Trans.	there are lots of farmers doing this today
	there are a lot of farmers doing this today
Trans. 1	there are lots of of farmers doing this today
Trans. 2	there are a lot farmers doing this today

Table: A multi-modality problem example

Problems

- **Multi-Modality Problems**



[Ran et al., 2020] Learning to Recover from Multi-Modality Errors for Non-Autoregressive Neural Machine Translation. ACL.

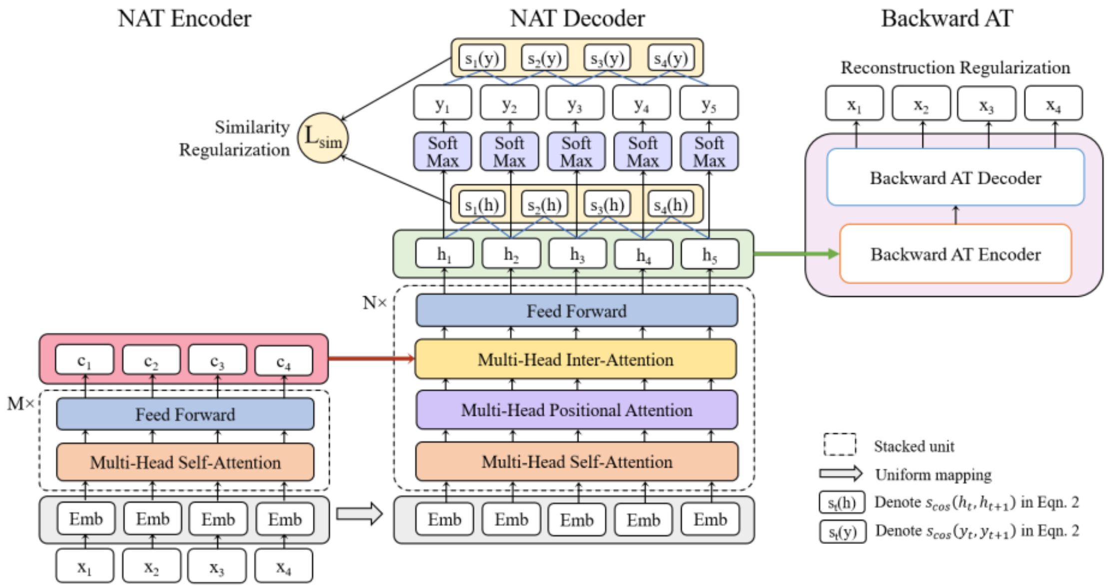
Problems

- **Under-Translation & Repeat Translation**

Source:	bei der coalergy sehen wir klimaveränderung als eine ernste gefahr für unser geschäft .
Reference:	at coalergy we view climate change as a very serious threat to our business .
AT:	in coalergy , we see climate change as a serious threat to our business .
NAT-BASE:	in the coalergy , we 'll see **climate climate change change** as a most serious danger for our business .
NAT-REG:	at coalergy , we 're seeing climate change as a serious threat to our business .
Source:	dies ist die großartigste zeit , die es je auf diesem planeten gab , egal , welchen maßstab sie anlegen :gesundheit , reichtum , mobilität , gelegenheiten , sinkende krankheitsraten .
Reference:	this is the greatest time there 's ever been on this planet by any measure that you wish to choose : health , wealth , mobility , opportunity , *declining rates of disease* .
AT:	this is the greatest time you 've ever had on this planet , no matter what scale you 're putting : health , wealth , mobility , opportunities , declining disease rates .
NAT-BASE:	this is the most greatest time that ever existed on this planet no matter what scale they 're **imsi : : , , mobility mobility , , scaniichospital rates** .
NAT-REG:	this is the greatest time that we 've ever been on this planet no matter what scale they 're **ianition** : health , wealth , mobility , opportunities , *declining disease rates* .
Source:	und manches davon hat funktioniert und manches nicht .
Reference:	and some of it worked , *and some of it didn 't* .
AT:	and some of it worked *and some of it didn 't work* .
NAT-BASE:	and some of it worked
NAT-REG:	and some of it worked *and some not* .

Problems

- **Under-Translation & Repeat Translation**



[Wang et al., 2019] Non-Autoregressive Machine Translation with Auxiliary Regularization. AAAI.

Non-Autoregressive NMT

- **Other Works**
  - End-to-end non-autoregressive neural machine translation with connectionist temporal classification. EMNLP 2018.
  - Fast decoding in sequence models using discrete latent variables. ICML 2018.
  - Deterministic Non-Autoregressive Neural Sequence Modeling by Iterative Refinement. EMNLP 2018.
  - Blockwise Parallel Decoding for Deep Autoregressive Models. NIPS 2018.
  - Semi-Autoregressive Neural Machine Translation. EMNLP 2018.
  - FlowSeq: Non-Autoregressive Conditional Sequence Generation with Generative Flow. EMNLP 2019.
  - Levenshtein Transformer. NeurIPS 2019.

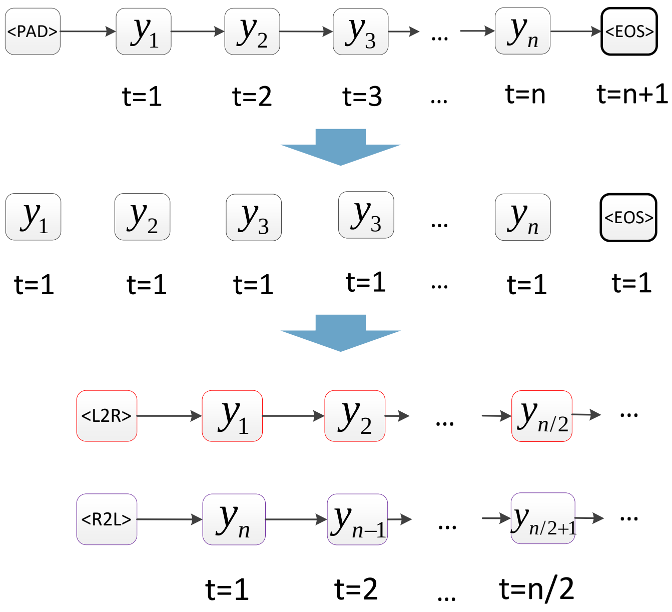
Non-Autoregressive NMT

- **Other Works**
  - Retrieving Sequential Information for Non-Autoregressive Neural Machine Translation. ACL 2019.
  - Non-autoregressive Machine Translation with Disentangled Context Transformer. ICML 2020.
  - A Study of Non-autoregressive Model for Sequence Generation. ACL 2020.
  - Latent-Variable Non-Autoregressive Neural Machine Translation with Deterministic Inference Using a Delta Posterior. AAAI 2020.
  - Parallel Machine Translation with Disentangled Context Transformer. ICML 2020.
  - Non-autoregressive Machine Translation with Latent Alignments. Arxiv 2020.

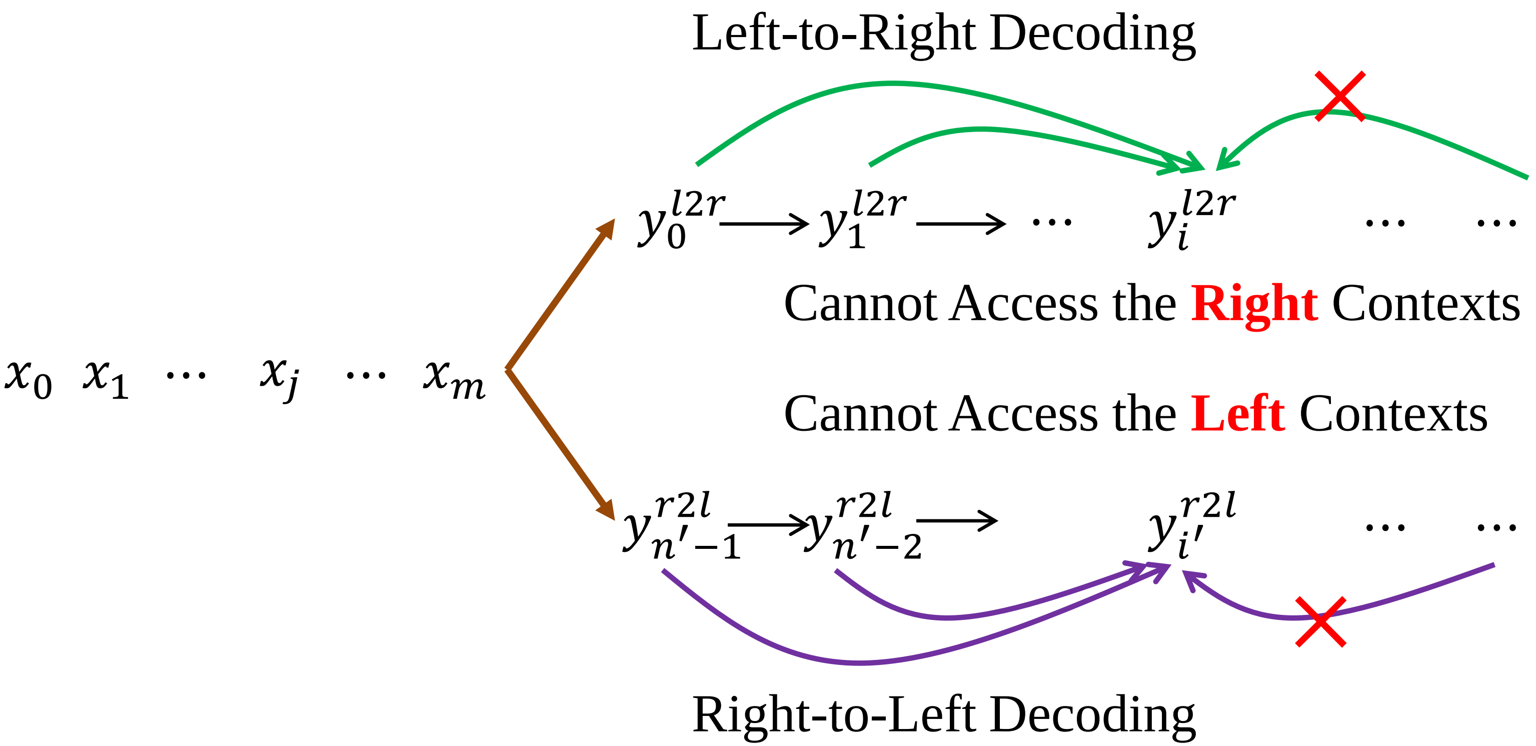
Section 4: Bidirectional Neural Machine Translation

Bidirectional Inference

- **Autoregressive Models**
  - Generate output from left to right
- **Non-Autoregressive Models**
  - Generate output in parallel
- **Bidirectional Models**
  - Generate output with L2R and R2L manner



Problems for Unidirectional Inference



Problems: Unbalanced Outputs

Source	捷克 总统 哈维 卸任 新 总统 仍 未 确定
Reference	czech president havel steps down while new president still not chosen
L2R	czech president leaves office
R2L	the outgoing president of the czech republic is still uncertain

Source	他们 正在 研制 一 种 超大型 的 叫做 炸弹 之 母 。
Reference	they are developing a kind of superhuge bomb called the mother of bombs .
L2R	they are developing a super , big , mother , called the bomb .
R2L	they are working on a much larger mother called the mother of a bomb .

Problems: Unbalanced Outputs

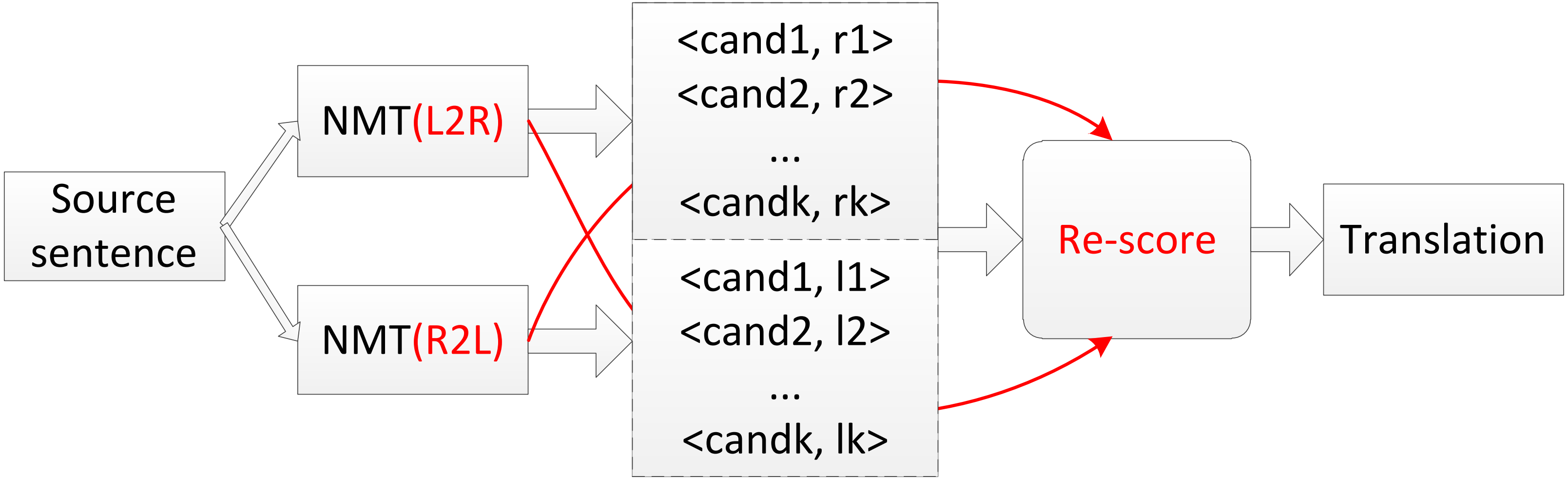
Model	The first 4 tokens	The last 4 tokens
L2R	**40.21%**	35.10%
R2L	35.67%	**39.47%**

Table: Translation accuracy of the first 4 tokens and last 4 tokens in NIST Chinese-English translation tasks.

How to effectively utilize

bidirectional decoding of NMT?

Solution 1: Bidirectional Agreement



[Liu et al., 2016] Agreement on Target-bidirectional Neural Machine Translation, NAACL.

Solution 1: Bidirectional Agreement

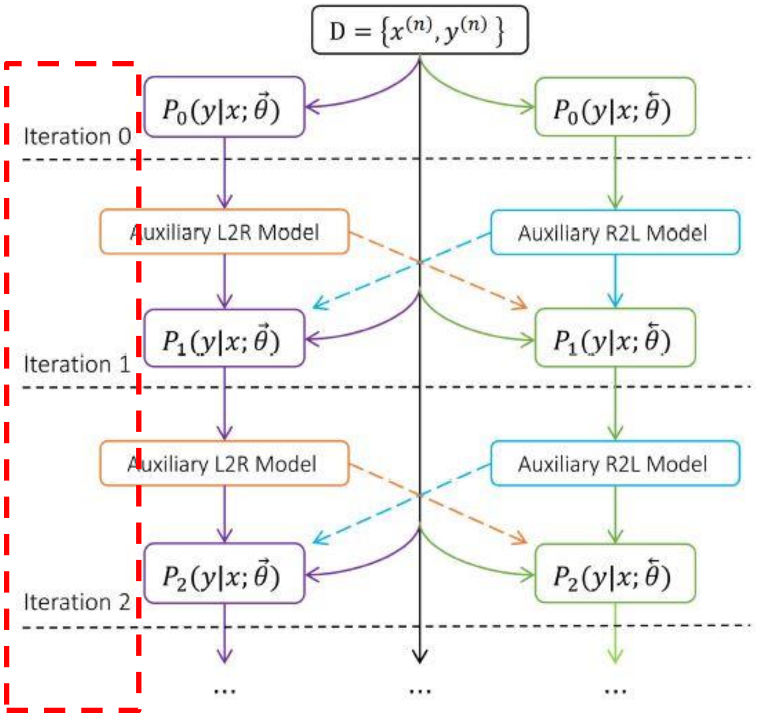
Drawbacks:

(1)Limited search space and search errors of beam search.

(2)The bidirectional decoders are often independent from each other during the translation

[Liu et al., 2016] Agreement on Target-bidirectional Neural Machine Translation, NAACL.

Solution 1: Bidirectional Agreement



$$
\begin{aligned}
L(\overrightarrow{\theta}) = & \sum_{n=1}^{N} \log P(y^{(n)}|x^{(n)};\overrightarrow{\theta}) \\
& - \lambda \sum_{n=1}^{N} \mathrm{KL}(P(y|x^{(n)};\overleftarrow{\theta})||P(y|x^{(n)};\overrightarrow{\theta})) \\
& - \lambda \sum_{n=1}^{N} \mathrm{KL}(P(y|x^{(n)};\overrightarrow{\theta})||P(y|x^{(n)};\overleftarrow{\theta}))
\end{aligned}
$$

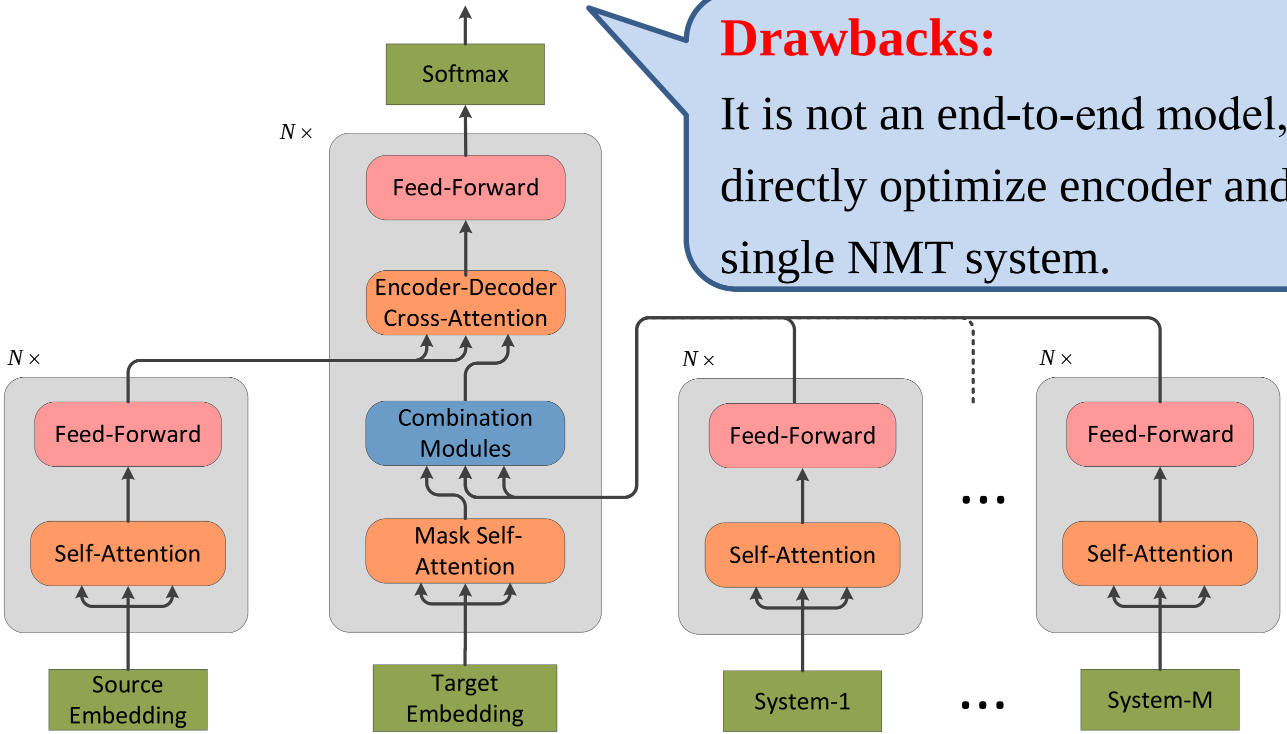
Drawbacks:

Two separate L2R and R2L models.

No interaction between bidirectional inference.

[Zhang et al., 2019] Regularizing Neural Machine Translation by Target-Bidirectional Agreement. AAAI.

Solution 2: Neural System Combination

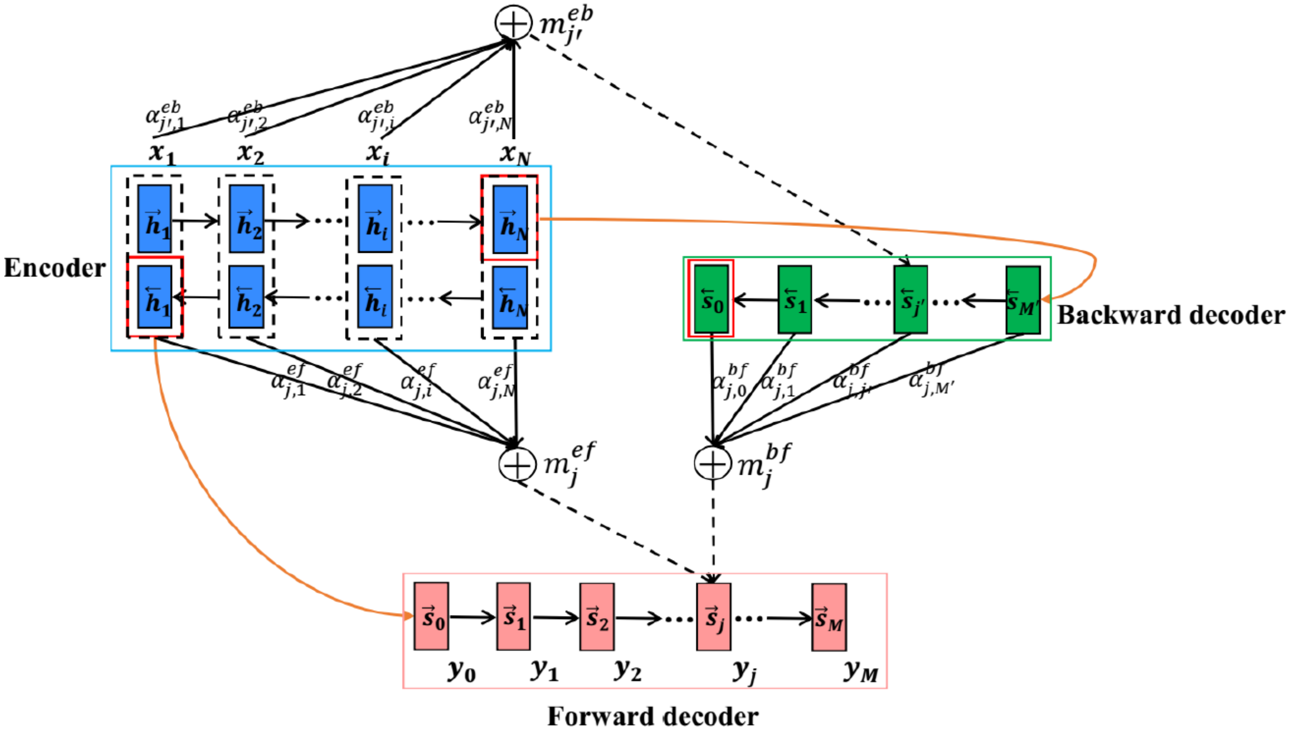


Drawbacks:

It is not an end-to-end model, and can't directly optimize encoder and decoder of single NMT system.

[Zhou et al., 2020] Deep Neural Network Based Machine Translation System Combination. TALLIP.

Solution 3: Asynchronous Bidirectional Decoding



[Zhang et al., 2018] Asynchronous Bidirectional Decoding for Neural Machine Translation. AAAI.

Solution 3: Asynchronous Bidirectional Decoding

Drawbacks:

(1) This work still requires two NMT models or decoders.

(2) Only the forward decoder can utilize information of backward decoder.

Question: How to utilize bidirectional decoding more effectively and efficiently?

[Zhang et al., 2018] Asynchronous Bidirectional Decoding for Neural Machine Translation. AAAI.

Our Solution: Synchronous Bidirectional Inference

Synchronous Bidirectional Neural Machine Translation

Long Zhou, Jiajun Zhang and Chengqing Zong.

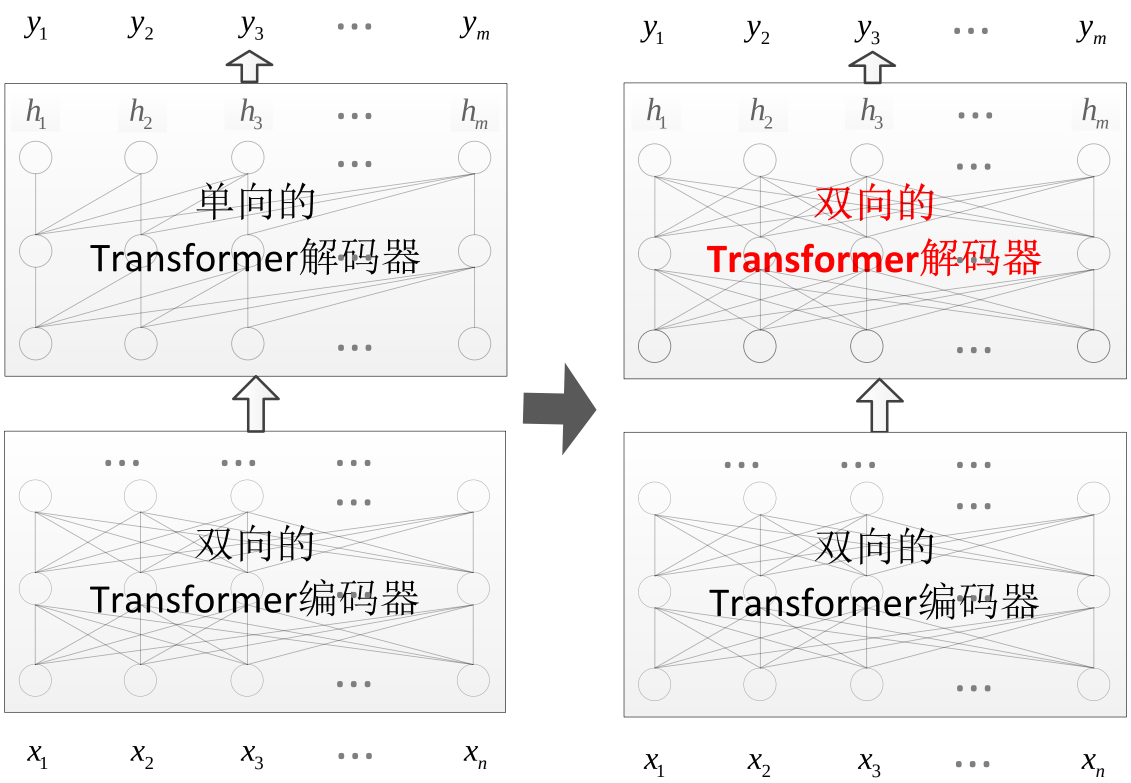
Transactions on ACL 2019.

Synchronous Bidirectional Inference for sequence generation

Jiajun Zhang, Long Zhou, Yang Zhao, and Chengqing Zong.

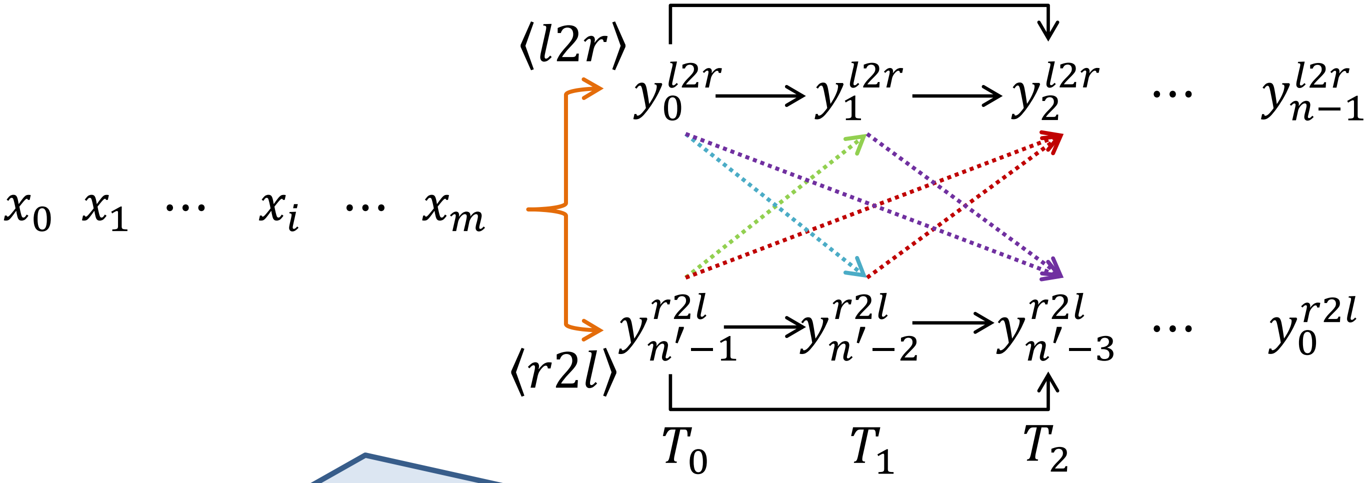
Journal of Artificial Intelligence 2020.

From Bidirectional Encoding to Bidirectional Decoding



78

Synchronous Bidirectional NMT



L2R (R2L) inference not only uses its previously generated outputs, but also uses future contexts predicted by R2L (L2R) decoding.

Synchronous Bidirectional NMT

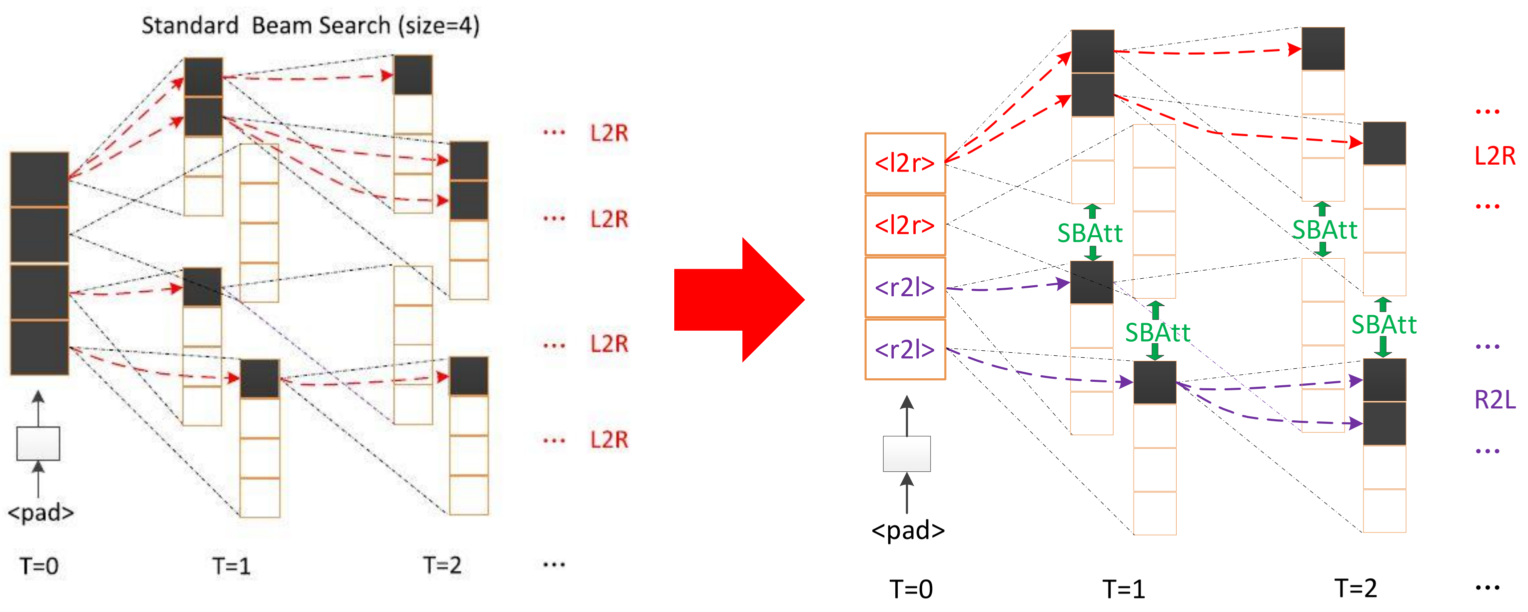
$$P(y|x) = \sum_{i=0}^{n-1} p(\vec{y}_i|\vec{y}_0 \cdots \vec{y}_{i-1}, x) \quad if\ L2R$$

$$P(y|x) = \begin{cases} \sum_{i=0}^{n-1} p(\vec{y}_i|\vec{y}_0 \cdots \vec{y}_{i-1}, x, \overleftarrow{\boldsymbol{y}}_{\boldsymbol{0}} \cdots \overleftarrow{\boldsymbol{y}}_{\boldsymbol{i-1}}) & if\ L2R \\ \sum_{i=0}^{n'-1} p(\overleftarrow{y}_i|\overleftarrow{y}_0 \cdots \overleftarrow{y}_{i-1}, x, \vec{\boldsymbol{y}}_{\boldsymbol{0}} \cdots \vec{\boldsymbol{y}}_{\boldsymbol{i-1}}) & if\ R2L \end{cases}$$

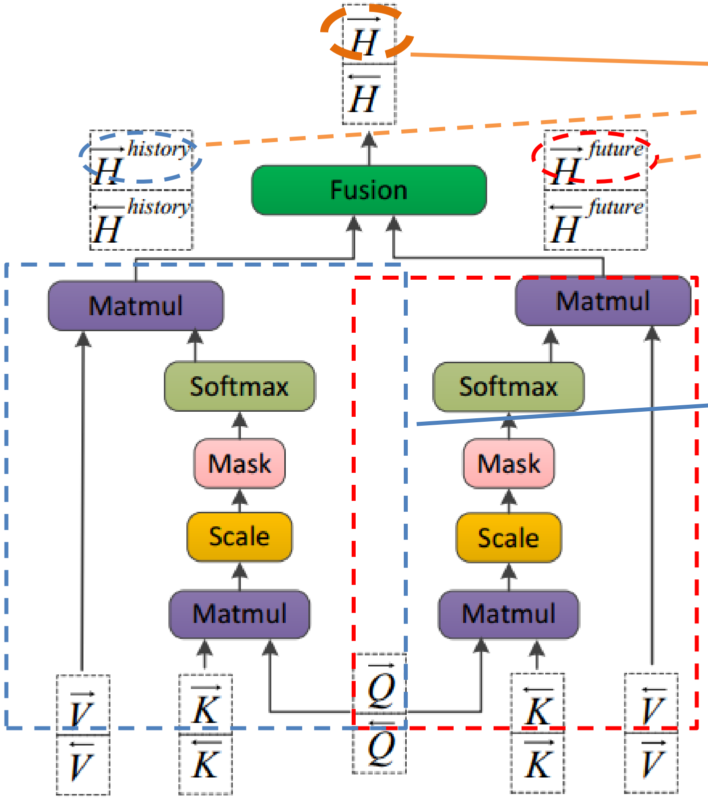
Advantages of SB-NMT

- We use a single model (one encoder and one decoder) to achieve the decoding with L2R and R2L generation, which can be processed in parallel;
- Via synchronous bidirectional attention model (SBAtt), our proposed model is an end-to-end joint framework and can optimize bidirectional decoding simultaneously;
- Instead of two-phase decoding scheme in previous work, our decoder is faster and more compact using one beam search algorithm.

Synchronous Bidirectional Beam Search



Synchronous Bidirectional Dot-Product Attention



$$\overrightarrow{H} = \text{Fusion}(\overrightarrow{H}^{\text{history}}, \overrightarrow{H}^{\text{future}})$$

- Linear Interpolation
- Nonlinear Interpolation
- Gate Mechanism

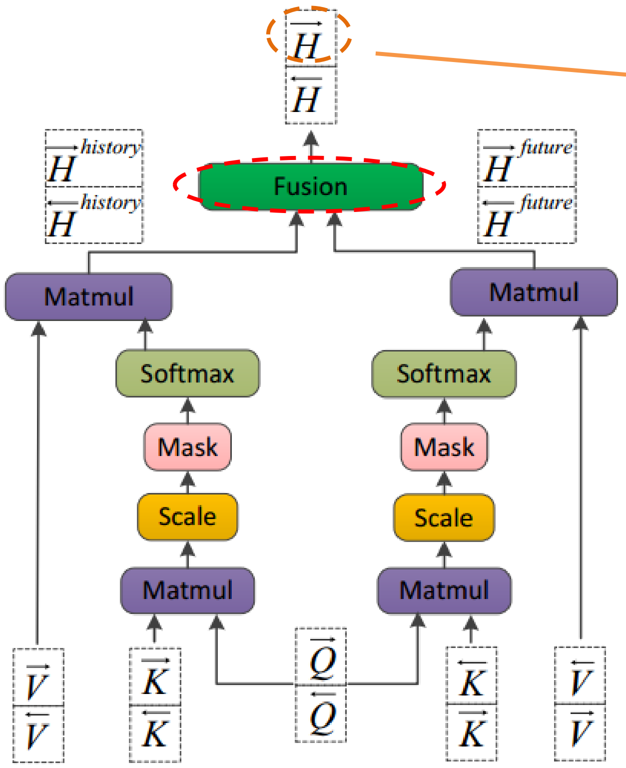
History Information:

$$\overrightarrow{H}^{\text{history}} = \text{Attention}(\overrightarrow{Q}, \overrightarrow{K}, \overrightarrow{V})$$

Future Information:

$$\overrightarrow{H}^{\text{future}} = \text{Attention}(\overrightarrow{Q}, \overleftarrow{K}, \overleftarrow{V})$$

Synchronous Bidirectional Dot-Product Attention



$$\overrightarrow{H} = \mathrm{Fusion}(\overrightarrow{H}^{\mathrm{history}}, \overrightarrow{H}^{\mathrm{future}})$$

- Linear Interpolation

$$\overrightarrow{H} = \overrightarrow{H}^{\mathrm{history}} + \lambda \times \overrightarrow{H}^{\mathrm{future}}$$

- Nonlinear Interpolation

$$\overrightarrow{H} = \overrightarrow{H}^{\mathrm{history}} + \lambda \times AF(\overrightarrow{H}^{\mathrm{future}})$$

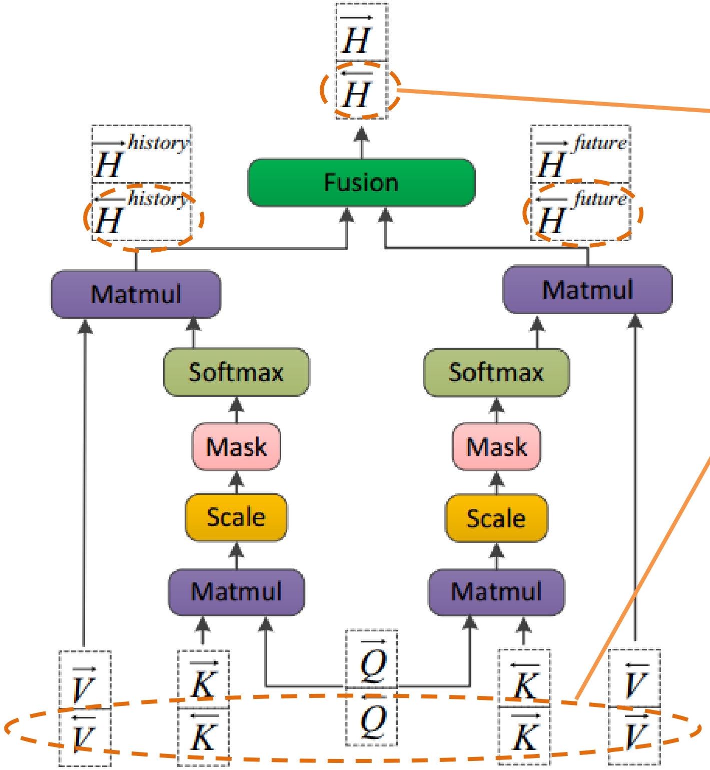
$$AF \rightarrow \begin{cases} tanh \\ relu \end{cases}$$

- Gate Mechanism

$$r_t, z_t = \sigma(W^g[\overrightarrow{H}^{\mathrm{history}}; \overrightarrow{H}^{\mathrm{future}}])$$

$$\overrightarrow{H} = r_t \odot \overrightarrow{H}^{\mathrm{history}} + z_t \odot \overrightarrow{H}^{\mathrm{future}}$$

Synchronous Bidirectional Dot-Product Attention

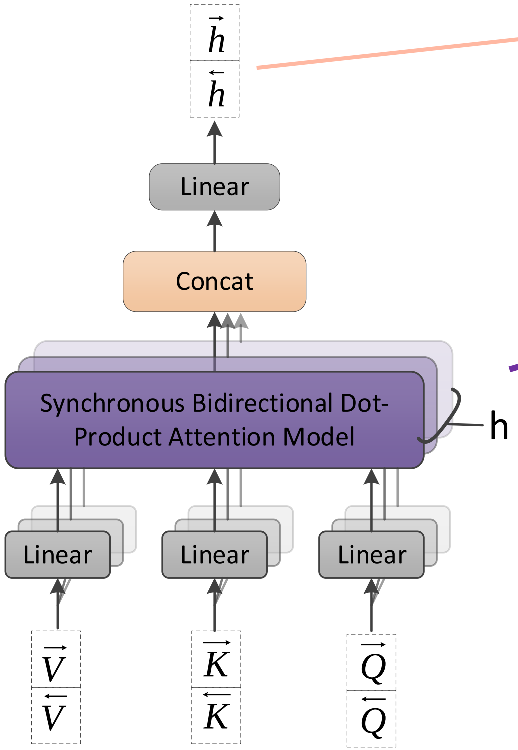


$$\overleftarrow{H}_i^f = Attention(\overleftarrow{Q}, \overleftarrow{K}, \overleftarrow{V})$$
$$\overleftarrow{H}_i^b = Attention(\overleftarrow{Q}, \overrightarrow{K}, \overrightarrow{V})$$
$$\overleftarrow{H}_i = Fusion(\overleftarrow{H}_i^f, \overleftarrow{H}_i^b)$$

We refer to the whole procedure as **SBDPT()**:

$$[\overrightarrow{H}_i, \overleftarrow{H}_i] = \mathrm{SBDPA}([\overleftarrow{Q}; \overrightarrow{Q}], [\overleftarrow{K}; \overrightarrow{K}], [\overleftarrow{V}; \overrightarrow{V}])$$

Synchronous Bidirectional Multi-Head Attention

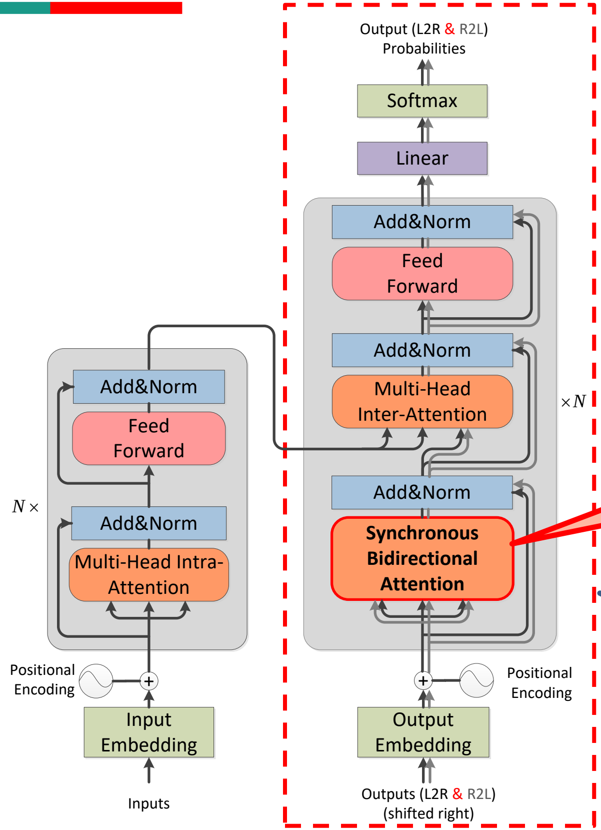


$$MultiHead([\overleftarrow{Q};\overrightarrow{Q}],[\overleftarrow{K};\overrightarrow{K}],[\overleftarrow{V};\overrightarrow{V}])$$
$$= Concat([\overrightarrow{H}_1;\overleftarrow{H}_1],\dots,[\overrightarrow{H}_h;\overleftarrow{H}_h])W^O$$

$$[\overrightarrow{H}_i;\overleftarrow{H}_i] = \mathrm{SBDPA}([\overleftarrow{Q};\overrightarrow{Q}]\mathrm{W}_{\mathrm{i}}^{\mathrm{Q}}, [\overleftarrow{K};\overrightarrow{K}]W_i^K, [\overleftarrow{V};\overrightarrow{V}]W_i^V)$$

Note that all parameters are the same as standard multi-head attention model.

Integrating Bidirectional Attention into NMT



$$p(\overrightarrow{y}_j|\overrightarrow{y}_{<j},\overleftarrow{y}_{<j},x,\theta)=softmax(\overrightarrow{s}^N W)$$
$$p(\overleftarrow{y}_j|\overleftarrow{y}_{<j},\overrightarrow{y}_{<j},x,\theta)=softmax(\overleftarrow{s}^N W)$$

Our proposed bidirectional attention model

Note that all bidirectional information flow in decoder runs in parallel.

Training

- **Strategy 1: Simply Reversing**
  - Over fitting
  - Inconsistency between training and testing

src: $x_1, x_2, \ldots, x_{m-1}, x_m$

tgt: $y_1, y_2, \ldots, y_{n-1}, y_n$

src:

$x_1, x_2, \ldots, x_{m-1}, x_m$

tgt:

<l2r> , $y_1, y_2, \ldots, y_{n-1}, y_n$

<r2l> , $y_n, y_{n-1}, \ldots, y_2, y_1$

Training

- **Strategy 2: Two-Pass Method**
  - Train L2R and R2L models, and translate source languages
  - Combine training data and train SB-NMT

src1:

$x_1, x_2, \ldots, x_{m-1}, x_m$

tgt1:

<l2r> , $y_1, y_2, \ldots, y_3, y_n$

<r2l> , $y^b_n, y^b_{n-1}, \ldots, y^b_2, y^b_1$

src2:

$x_1, x_2, \ldots, x_{m-1}, x_m$

tgt2:

<l2r> , $y^f_1, y^f_2, \ldots, y^f_{n-1}, y^f_n$

<r2l> , $y_n, y_{n-1}, \ldots, y_2, y_1$

Training

- **Strategy 3: Fine-Tune Method**

(1) Bidirectional inference without interaction

$$P(y|x) = \begin{cases} \sum_{i=0}^{n-1} p(\vec{y}_i|\vec{y}_0 \cdots \vec{y}_{i-1}, x) & if\ L2R \\ \sum_{i=0}^{n'-1} p(\overleftarrow{y}_i|\overleftarrow{y}_0 \cdots \overleftarrow{y}_{i-1}, x) & if\ R2L \end{cases}$$

(2) Fine-tuning with interaction

$$P(y|x) = \begin{cases} \sum_{i=0}^{n-1} p(\vec{y}_i|\vec{y}_0 \cdots \vec{y}_{i-1}, x, \boldsymbol{\overleftarrow{y}_0 \cdots \overleftarrow{y}_{i-1}}) & if\ L2R \\ \sum_{i=0}^{n'-1} p(\overleftarrow{y}_i|\overleftarrow{y}_0 \cdots \overleftarrow{y}_{i-1}, x, \boldsymbol{\vec{y}_0 \cdots \vec{y}_{i-1}}) & if\ R2L \end{cases}$$

Experiments: Machine Translation

- **Dataset**
  - NIST Chinese-English translation (2M, 30K tokens, MT03-06 as test set)
  - WMT14 English-German translation (4.5M, 37K shared tokens, newstest2014 as test set)

- **Train details**
  - Transformer_big setting
  - Chinese-English: 1 GPUs, single model, case-insensitive BLEU.
  - English-German: 3 GPUs, model averaging, case-insensitive BLEU.

Experiments: Machine Translation

- **Baselines**
  - **Moses:** an Open source phrase-based SMT system.
  - **RNMT:** RNN-based NMT with default setting.
  - **Transformer:** Predict target sentence from left to right.
  - **Transformer(R2L):** Predict sentence from right to left.
  - **Rerank-NMT:** (1) first run beam search to obtain two k-best lists; (2) then re-score and get the best candidate.
  - **ABD-NMT:** (1) use backward decoder to generate reverse sequence states; (2) perform beam search on the forward decoder to find the best translation.

Experiments: Machine Translation

- **Results on Chinese-English Translation**
  - Effect of Fusion Mechanism

Best performance (less parameters)

Be sensitive to lambda

Combination		λ =0.1	λ =0.5	λ =1.0
Linear		51.05	50.71	46.98
Nonlinear	*tanh*	50.99	50.72	50.96
	relu	50.79	50.57	50.71
Gate		50.51		

Experiments: Machine Translation

- **Results on Chinese-English Translation**
  - Translation Quality

Model	DEV	MT03	MT04	M05	MT06	AVE	Δ
Moses	37.85	37.47	41.20	36.41	36.03	37.78	-9.41
RNMT	42.43	42.43	44.56	41.94	40.95	42.47	-4.72
Transformer	48.12	47.63	48.32	47.51	45.31	47.19	-
Transformer(R2L)	47.81	46.79	47.01	46.50	44.13	46.11	-1.08
Rerank-NMT	49.18	48.23	48.91	48.73	46.51	48.10	+0.91
ABD-NMT	48.28	49.47	48.01	48.19	47.09	48.19	+1.00
Our Model	**50.99**	**51.61**	**51.41**	**51.19**	**49.84**	**51.01**	**+3.82**

!!!!!!

Experiments: Machine Translation

- **Results on English-German Translation**

Model	Test
GNMT (Wu et al., 2016)	24.61
Conv (Gehring et al., 2017)	25.16
AttIsAll (Vaswani et al., 2017)	28.40
Transformer	27.72
Transformer(R2L)	27.13
Rerank-NMT	27.81
ABD-NMT	28.22
Our Model	**29.21**

State-of-the-art NMT models

(+1.49)

MT Analysis

- **Parameters and Speeds**

Not increase any parameters except for lambda.

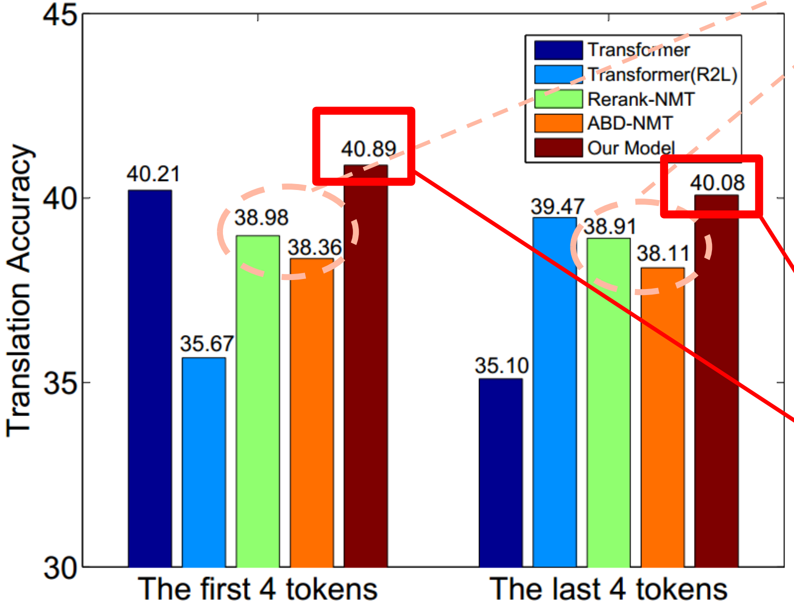
Model	Param	Speed	
		Train	*Test*
Transformer	207.8M	2.07	19.97
Transformer(R2L)	207.8M	2.07	19.81
Rerank-NMT	415.6M	1.03	6.51
ABD-NMT	333.8M	1.18	7.20
Our Model	207.8M	1.26	17.87

Two or three times faster than Rerank-NMT and ABD-NMT

MT Analysis

- **Effect of Unbalance Outputs**

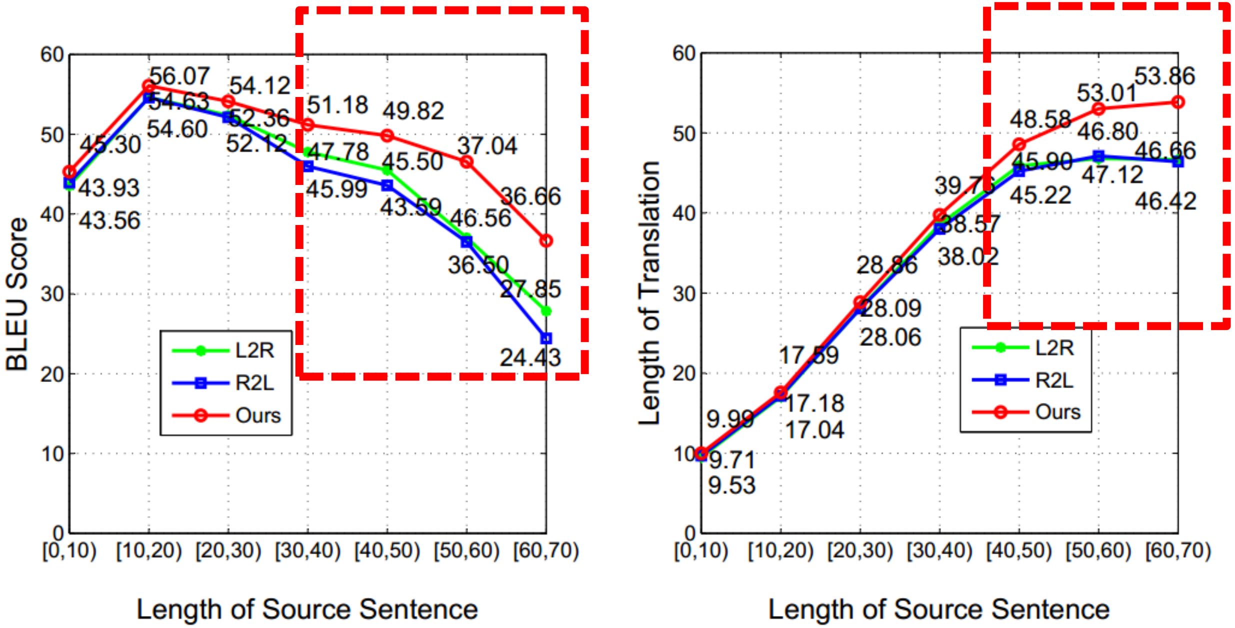
Balance between L2R and R2L models



Get the best translation accuracy no matter for the first 4 words and the last 4 words.

MT Analysis

- **Effect of Long Sentence**



MT Analysis

- **Subjective Evaluation**

Over-translation < Under-translation

Model	Over-Trans		Under-Trans	
	Ratio	Δ	Ratio	Δ
L2R	0.07%	-	7.85%	-
R2L	0.14%	-	7.81%	-
Our Model	0.07%	-0.00%	5.42%	-30.6%

Especially effective for alleviating under-translation

MT Analysis

- Case Study

Source	捷克 总统 哈维 卸任 新 总统 仍 未 确定
Reference	czech president havel steps down while new president still not chosen
L2R	czech president leaves office
R2L	the outgoing president of the czech republic is still uncertain
Ours	czech president havel leaves office , new president yet to be determined

Source	他们 正在 研制 一 种 超大型 的 叫做 炸弹 之 母 。
Reference	they are developing a kind of superhuge bomb called the mother of bombs .
L2R	they are developing a super , big , mother , called the bomb .
R2L	they are working on a much larger mother called the mother of a bomb .
Ours	they are developing a super-large scale , called the mother of the bomb .

MT Analysis

- **Case Study**

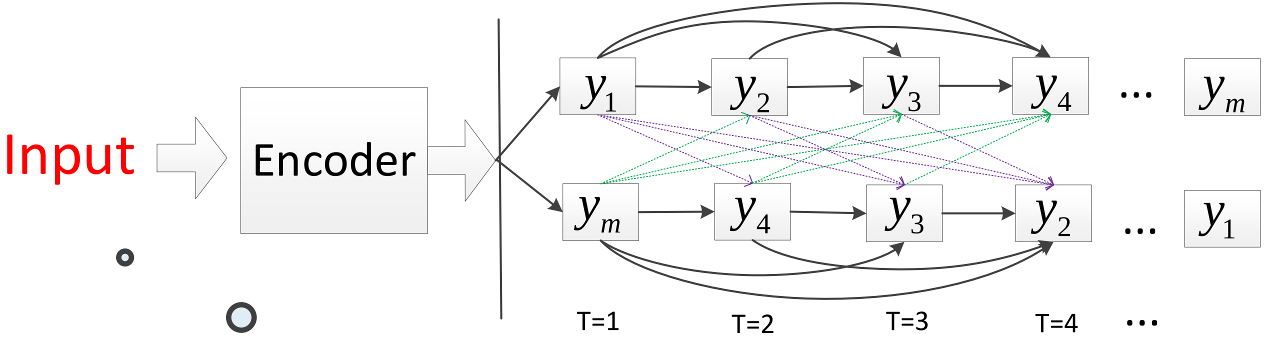
Source	捷克 总统 哈维 卸任 新 总统 仍 未 确定
Reference	czech president havel steps down while new president still not chosen
L2R	czech president leaves office
R2L	the outgoing president of the czech republic is still uncertain
Ours	czech president havel leaves office , new president yet to be determined

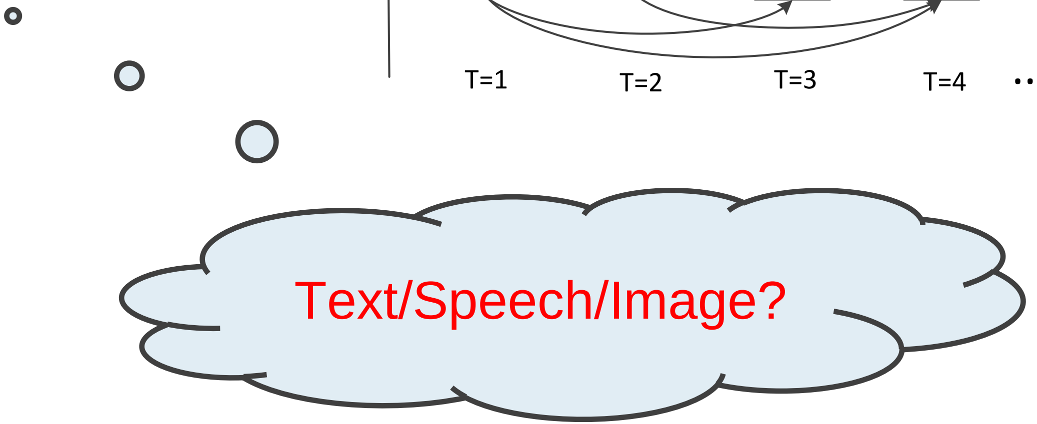
L2R produces good prefix, whereas R2L generates better suffixes.

Our approach can make full use of bidirectional decoding and produce balanced outputs in these cases.

Bidirectional Inference: Extending Tasks

Synchronous Bidirectional Decoder:





Bidirectional Inference for Text Summarization

- **Definition**
  - ➢ Generate a shorter version of a given sentence
  - ➢ Preserve its original meaning
- **Example**

Input	resident nelson mandela acknowledged saturday the african national congress violated human rights during apartheid , setting him at odds with his deputy president over a report that has divided much of south africa .
Output	mandela acknowledges human rights violations by african national congress

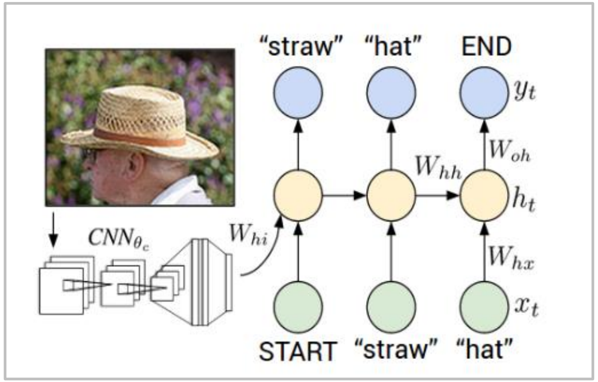
Bidirectional Inference for Text Summarization

- **Results on DUC2004 and English Gigaword**

Model	DUC-2004			English Gigaword		
	R1	R2	R-L	R1	R2	R-L
ABS	26.55	7.06	22.05	29.55	11.32	26.42
Feats2s	28.35	9.46	24.59	32.67	15.59	30.64
Selective-Env	**29.21**	9.56	25.51	**36.15**	**17.54**	**33.63**
Transformer	28.09	9.52	24.91	34.12	16.04	31.46
Our Model	29.17	10.30	**26.05**	35.68	17.39	32.89
	+1.08	+0.78	+1.14	+1.56	+1.25	+1.43

Bidirectional Inference for Image Caption

- **Setup**
  - Dataset
    (1) Flickr30k (Young et al., 2014)
    (2) 29,000 image-caption for training
    (3) 1014 for validation and 2000 for test



  - Baselines
    (1) VGGNet encoder + LSMT decoder (Xu et al., 2015)
    (2) Transformer

Bidirectional Inference for Image Caption

- **Results on English Image Caption**
  - BLEU score

Method	**Validation**	**Test**
Xu et al., (2015)	~	19.90
Transformer	22.11	21.25
Ours	**23.27**	**22.41**

Bidirectional Inference: Improving Efficiency

Sequence Generation: From Both Sides to the Middle

Long Zhou, Jiajun Zhang, Chengqing Zong, and Heng Yu

In Proceedings of IJCAI 2019.

Sequence Generation: From Both Sides to the Middle

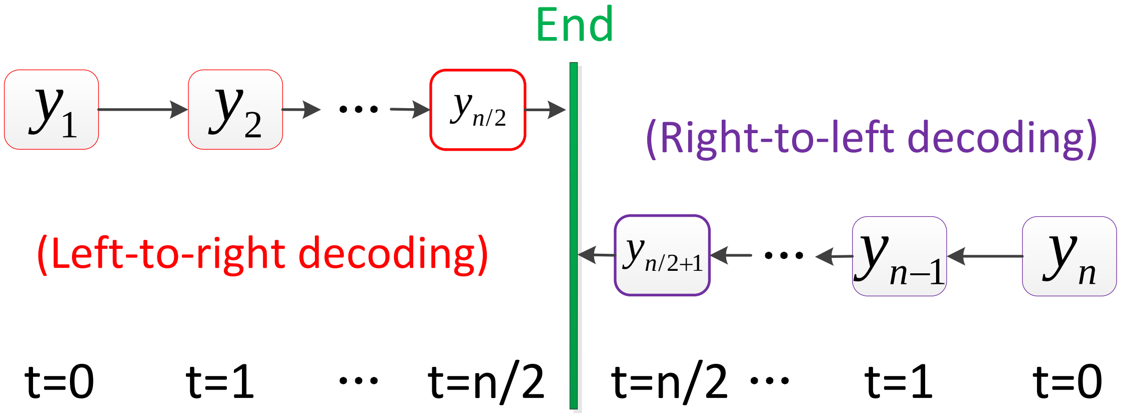
- **Autoregressive Translation (AT)**
  - ➢ Advantages: high quality
  - ➢ Disadvantages:

    (1) time-consuming when sentences become longer

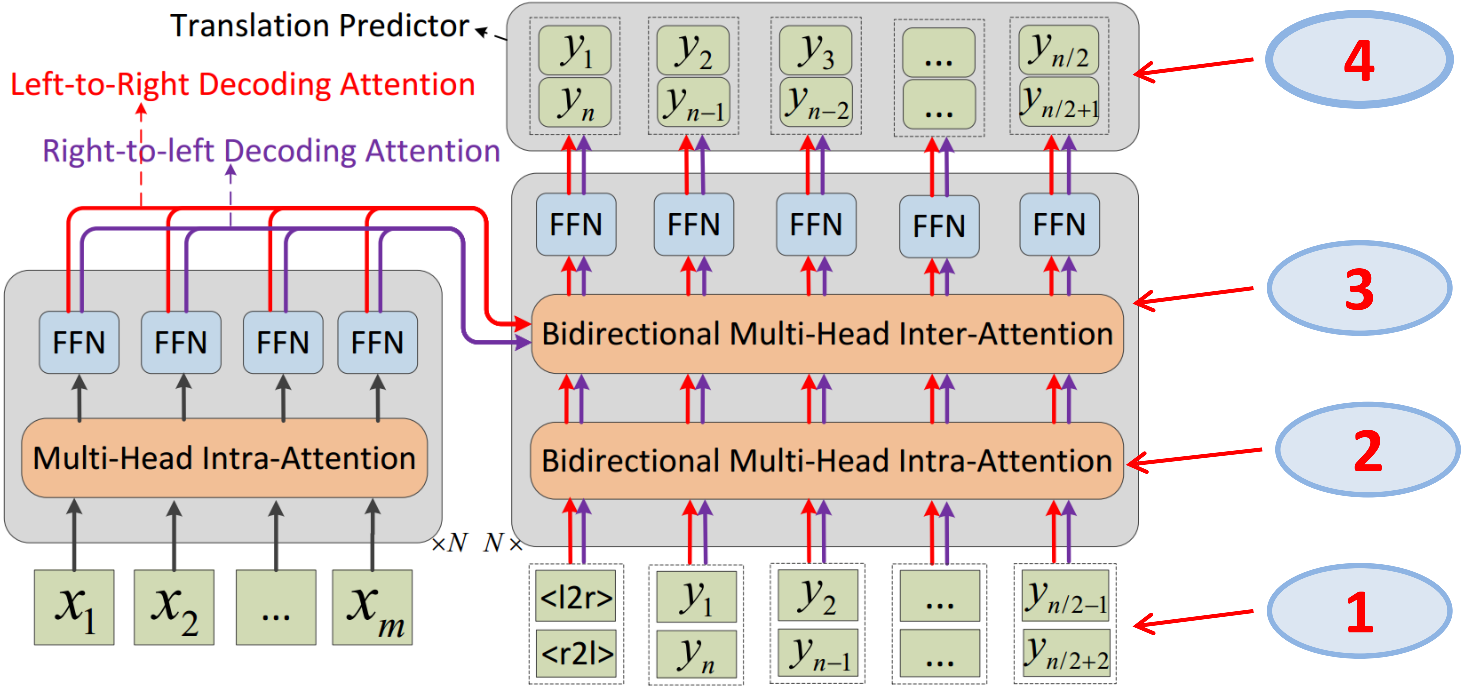
    (2) lack the guidance of future information
- **Non-Autoregressive Translation (NAT)**
  - ➢ Advantages: speed up the decoding procedure
  - ➢ Disadvantages: substantial drop in generation quality

Sequence Generation: From Both Sides to the Middle

- **SBSG: Synchronous Bidirectional Sequence Generation**
  - Speedup decoding: Generates two tokens at a time
  - Improve quality: Rely on history and future context



The Framework (SBSG)



Training

- **Training objective:**

$$J(\theta) = \frac{1}{Z}\sum_{z=1}^{Z}\sum_{j=1}^{n/2}\{log\, p(\overrightarrow{y}_j^{(z)} \mid \overrightarrow{y}_{<j}^{(z)}, \overleftarrow{y}_{<j}^{(z)}, x^{(z)}, \theta)$$
$$+ log\, p(\overleftarrow{y}_j^{(z)} \mid \overleftarrow{y}_{<j}^{(z)}, \overrightarrow{y}_{<j}^{(z)}, x^{(z)}, \theta)\}$$

- **The Smoothing model:**

$y:$	<l2r>	y_1	y_2	$\cdots$	$y_{n/2}$	$y_{n/2+1}$	<eos>
$y:$	<r2l>	y_n	y_{n-1}	$\cdots$	$y_{n/2+2}$	<null>	<eos>

Application to NMT

- **Train details**

  (1) WMT14 EN-DE; NIST ZH-EN; WMT16 EN-RO

  (2) *Transformer_base* setting

- **Baselines**

  (1) Transformer: autoregressive neural machine translation

  (2) NAT: non-autoregressive neural machine translation

  (3) D-NAT: NAT model based on iterative refinement

  (4) LT: NAT model based on discrete latent variables

  (5) SAT: semi-autoregressive neural machine translation

Application to NMT

System	Architecture	English-German Quality	English-German Speed	Chinese-English Quality	Chinese-English Speed	English-Romanian Quality	English-Romanian Speed
Existing NMT systems							
[Gu *et al.*, 2017]	NAT	17.35	N/A	-	-	26.22	15.6×
	NAT (s=100)	19.17	N/A	-	-	29.79	2.36×
[Lee *et al.*, 2018]	D-NAT	12.65	11.71×	-	-	24.45	16.03×
	D-NAT (adaptive)	18.91	1.98×	-	-	29.66	5.23×
[Kaiser *et al.*, 2018]	LT	19.80	3.89×	-	-	-	-
	LT (s=100)	22.50	N/A	-	-	-	-
[Wang *et al.*, 2018] (beam search)	SAT (K=2)	26.90	1.51×	39.57	1.69×	-	-
	SAT (K=6)	24.83	2.98×	35.32	3.18×	-	-
[Wang *et al.*, 2018] (greedy search)	SAT (K=2)	26.09	1.70×	38.37	1.71×	-	-
	SAT (K=6)	23.93	4.57×	33.75	4.70×	-	-
Our NMT systems							
This work (beam search)	Transformer	27.06	1.00×	46.56	1.00×	32.28	1.00×
	Transformer (R2L)	26.71	1.02×	44.63	0.94×	32.29	0.98×
	Our Model	**27.45**	1.38×	**47.82**	1.41×	**33.02**	1.43×
This work (greedy search)	Transformer	26.23	1.00×	44.63	1.00×	31.71	1.00×
	Transformer (R2L)	25.38	0.97×	43.68	0.98×	31.19	1.04×
	Our Model	**27.22**	1.61×	**47.50**	1.51×	**32.82**	1.46×

Application to Text Summarization

- **Example**

the **sri lankan** government on wednesday announced the **closure** of **government schools** with **immediate effect** as a **military campaign** against **tamil separatists escalated** in the north of the country .

sri lanka closes schools as war escalates

- **Setup**

(1) English Gigaword dataset (3.8M training set, 189K dev set, DUC2004 as our test set)

(2) shared vocabulary of about 90K word types

(3) *Transformer_base* setting, ROUGE-1, ROUGE-2, ROUGE-L

Application to Text Summarization

DUC2004	RG-1	RG-2	RG-L	Speed
ABS‡	26.55	7.06	22.05	-
Feats2s‡	28.35	9.46	24.59	-
Selective-Enc‡	29.21	9.56	25.51	-
Transformer	28.09	9.52	24.91	1.00×
SBSG (beam)	28.77	10.11	26.11	1.48×
SBSG (greedy)	28.70	9.88	25.93	2.09×

Our proposed SBSG model significant outperforms the conventional Transformer model in terms of both decoding speed and generation quality.

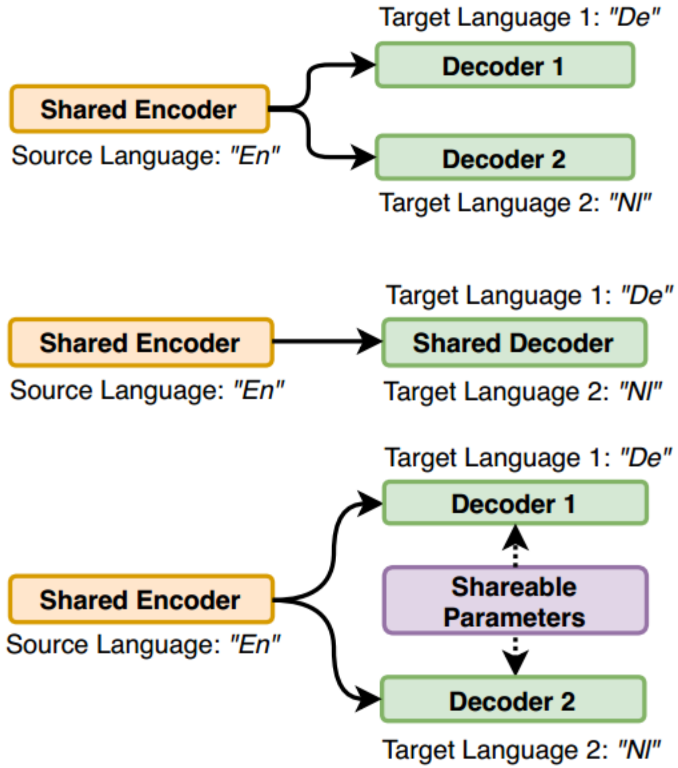
From Two Directions to Two Tasks

Synchronously Generating Two Languages with Interactive Decoding

Yining Wang, Jiajun Zhang, Long Zhou, Yuchen Liu and Chengqing Zong.

In Proceedings of EMNLP 2019.

Conventional Multilingual Translation



Separate Encoder or Decoder network

Shared Encoder or Decoder network

Shared with partial parameter

[Sachan and Neubig, 2018] Parameter Sharing Methods for Multilingual Self-Attentional Translation Models. WMT.

From Generating Two Directions to Generating Two languages

$y_0^{l2r} \longrightarrow y_1^{l2r} \longrightarrow \cdots \quad y_i^{l2r} \quad \cdots \quad \cdots$

Left-to-Right Decoding

$x_0 \ x_1 \ \cdots \ x_j \ \cdots \ x_m$

Right-to-Left Decoding

$y_{n'-1}^{r2l} \longrightarrow y_{n'-2}^{r2l} \longrightarrow \cdots \quad y_{i'}^{r2l} \quad \cdots \quad \cdots$

$\Downarrow$

$y_0 \longrightarrow y_1 \longrightarrow \cdots \quad y_i \quad \cdots \quad \cdots$

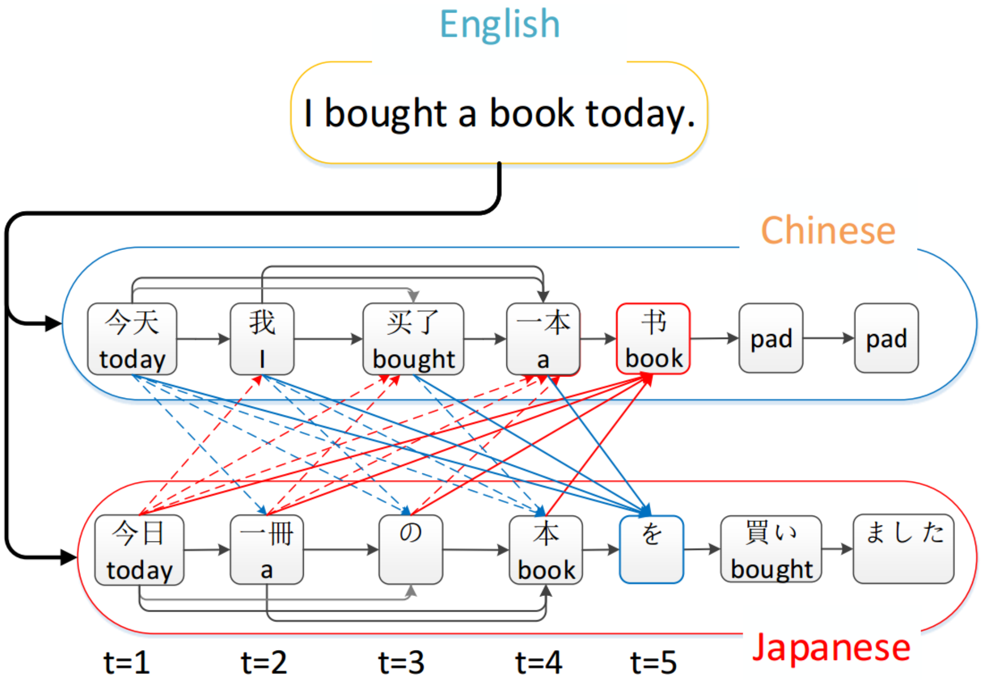
English-to-Chinese Decoding

$x_0 \ x_1 \ \cdots \ x_j \ \cdots \ x_m$

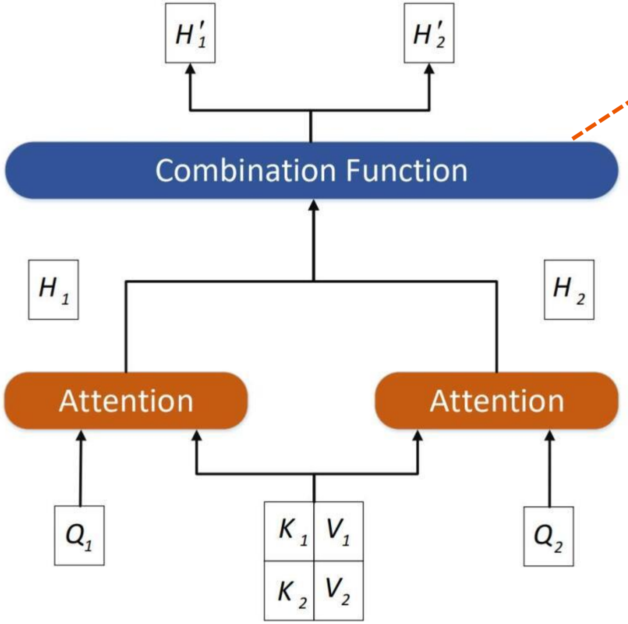
English-to-Japanese Decoding

$z_0 \longrightarrow z_1 \longrightarrow \cdots \quad z_j \quad \cdots \quad \cdots$

Synchronously Generating Two Languages with Interactive Decoding



Interactive Attention



Combination of two languages:

$$H'_1 = f(H_1;\widetilde{H}_1) = H_1 + \lambda \times \tanh(\widetilde{H}_1)$$
$$H'_2 = f(H_2;\widetilde{H}_2) = H_2 + \lambda \times \tanh(\widetilde{H}_2)$$

Lang-2 information:

$$H_2 = \text{Attention}\ (Q_2, K_2, V_2)$$
$$\widetilde{H}_2 = \text{Attention}\ (Q_2, K_1, V_1)$$

Lang-1 information:

$$H_1 = \text{Attention}\ (Q_1, K_1, V_1)$$
$$\widetilde{H}_1 = \text{Attention}\ (Q_1, K_2, V_2)$$

Training

- **Training objective**

$$\boldsymbol{L(\theta)} = \sum_{(x,y^1,y^2)\in D} \left(\sum_{i=1}^{|y_i^1|} \log P\,(y_i^1|x) + \sum_{i=1}^{|y_i^2|} \log P\,(y_i^2|x) \right)$$

for each decoder:

$$\log P(y^1|x) = \log \prod_{i=0}^{n-1} p(y_i^1|x, y_0^1, \ldots, y_{i-1}^1, y_0^2, \ldots, y_{i-1}^2)$$

$$\log P(y^2|x) = \log \prod_{i=0}^{n-1} p(y_i^2|x, y_0^2, \ldots, y_{i-1}^2, y_0^1, \ldots, y_{i-1}^1)$$

Training

- **Constructing/Using Trilingual Data**

Step1(train):

(x1, y1) $\xrightarrow{\text{Training}}$ Model: M1

(x2, y2) $\xrightarrow{\text{Training}}$ Model: M2

Step2 (decode):

M1 $\xrightarrow[\text{Decoding}]{x_2}$ (x2,y2*)

M2 $\xrightarrow[x_1]{\text{Decoding}}$ (x1,y1*)

Step3 (combination):

(x1,y1,y2*) ∪ (x2,y1*,y2)

Experiments

- ## Training Data
  - Small scale (IWSLT)

	IWSLT			
	En-Ja	En-Zh	En-De	En-Fr
Train	223K	231K	206K	233K
Test	3003	3003	1305	1306

  - Large scale (WMT)

	WMT14 subset		WMT14
	En-De	En-Fr	En-De
Train	2.43M	2.43M	4.50M
Test	3003	3003	3003

Experiments

- **Results on IWSLT Dataset**

Method	En-Zh/Ja		En-De/Fr	
	En-Zh	En-Ja	En-De	En-Fr
Indiv	15.68	16.56	27.11	40.62
Indiv + pseudo	16.72	18.02	28.47	40.39
Multi	17.06	18.31	27.79	40.97
Multi + pseudo	17.10	18.40	28.56	40.62
SyncTrans	**17.97**	**19.31**	**29.16**	**41.53**

- *Indiv*: the NMT models trained on individual language pair
- *Multi*: typical one-to-many translation adopting Johnson et al. (2017) method on Transformer
- *SyncTrans* significantly outperforms both *Indiv* and *Multi*.

Experiments

- **Results on WMT Dataset**

Method	WMT14 (2.43M)		WMT14 (4.50M)
	En-De	En-Fr	En-De
Indiv	24.33	37.12	26.53
Multi	23.46	36.33	25.81
SyncTrans	**24.84**†*	**37.66**†*	**27.01**†*

➢ Our *SyncTrans* also performs better than *Indiv* and *Multi* on large scale data.

From Two Directions to Two Tasks

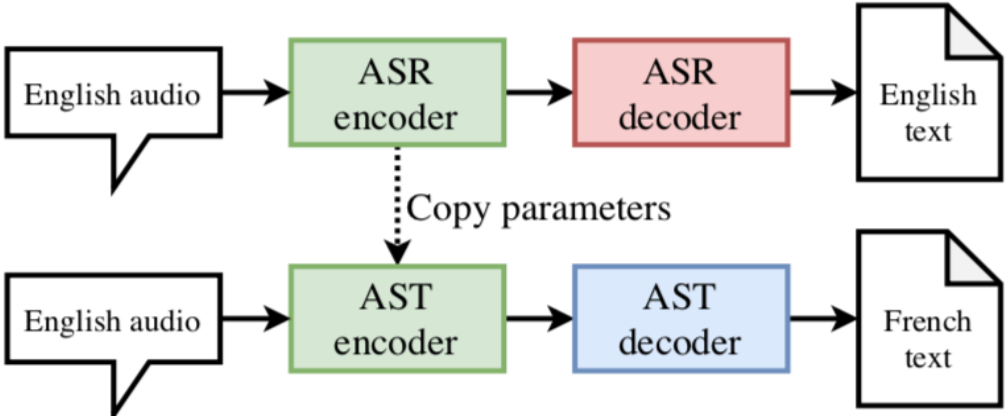
Synchronous Speech Recognition and Speech-to-Text Translation with Interactive Decoding

Yuchen Liu, Jiajun Zhang, Hao Xiong, Long Zhou, Zhongjun He, Hua Wu, Haifeng Wang, Chengqing Zong

In Proceedings of AAAI 2020.

Combine Speech Recognition and Translation

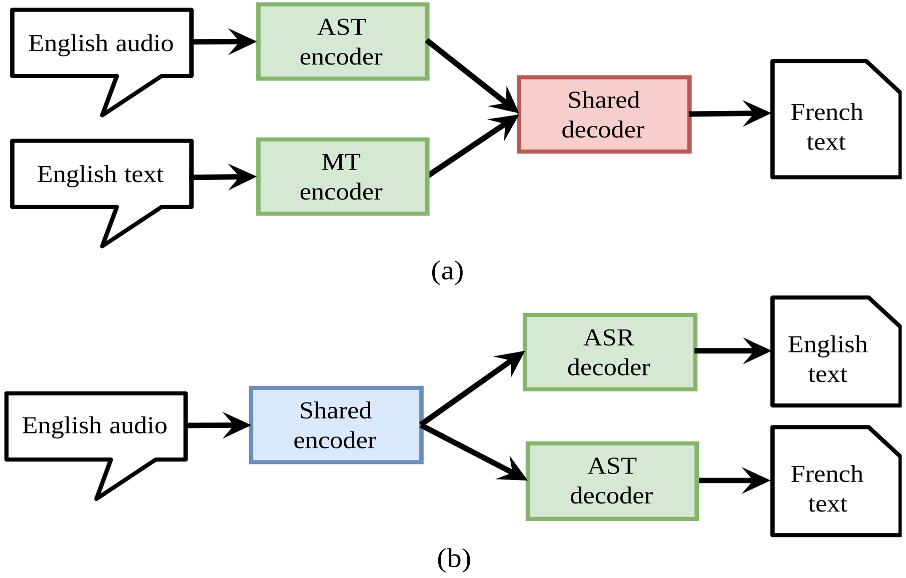
- **Pre-training**



[Sameer et al., 2018] Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. NAACL.

Combine Speech Recognition and Translation

- **Multi-task Learning**



[J. Weiss et al., 2017] Sequence-to-sequence models can directly translate foreign speech. Interspeech.

Combine Speech Recognition and Translation

- **Multi-task Learning**

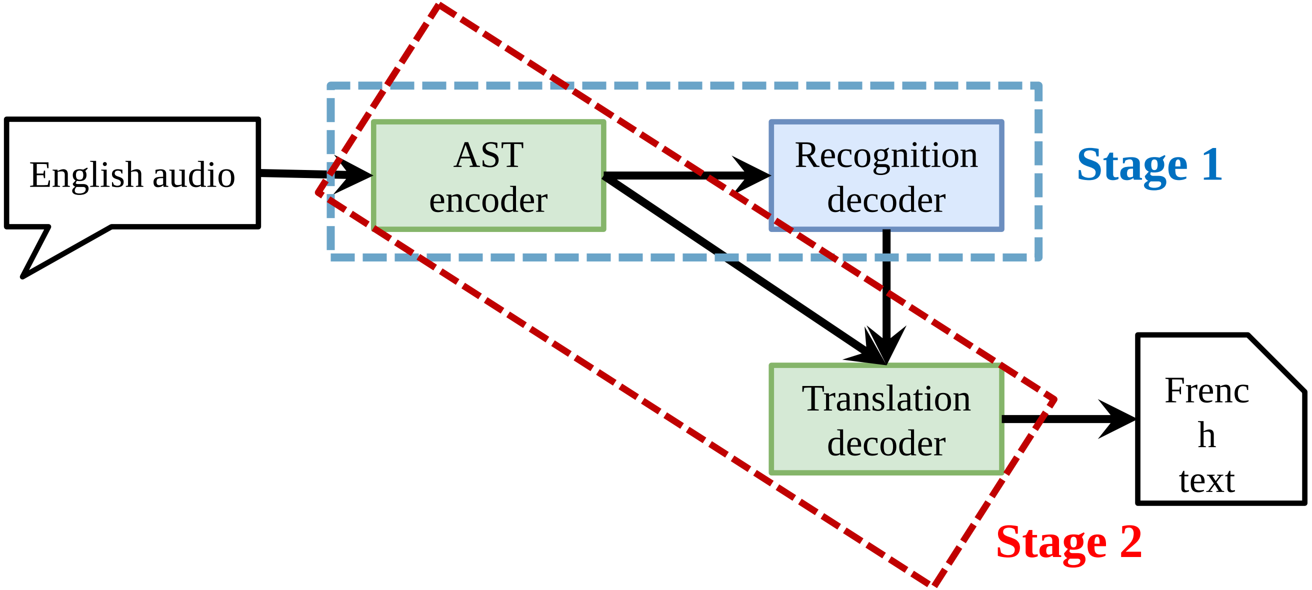
Drawbacks:

(1) Different tasks are treated independently which cannot use the information of each other.
(2) During decoding, only one task can be generated at one time.

[J. Weiss et al., 2017] Sequence-to-sequence models can directly translate foreign speech. Interspeech.

Combine Speech Recognition and Translation

- **Two-Stage Model**



[Sperber et al., 2019] Attention-Passing Models for Robust and Data-Efficient End-to-End Speech Translation. ACL 2019.

Combine Speech Recognition and Translation

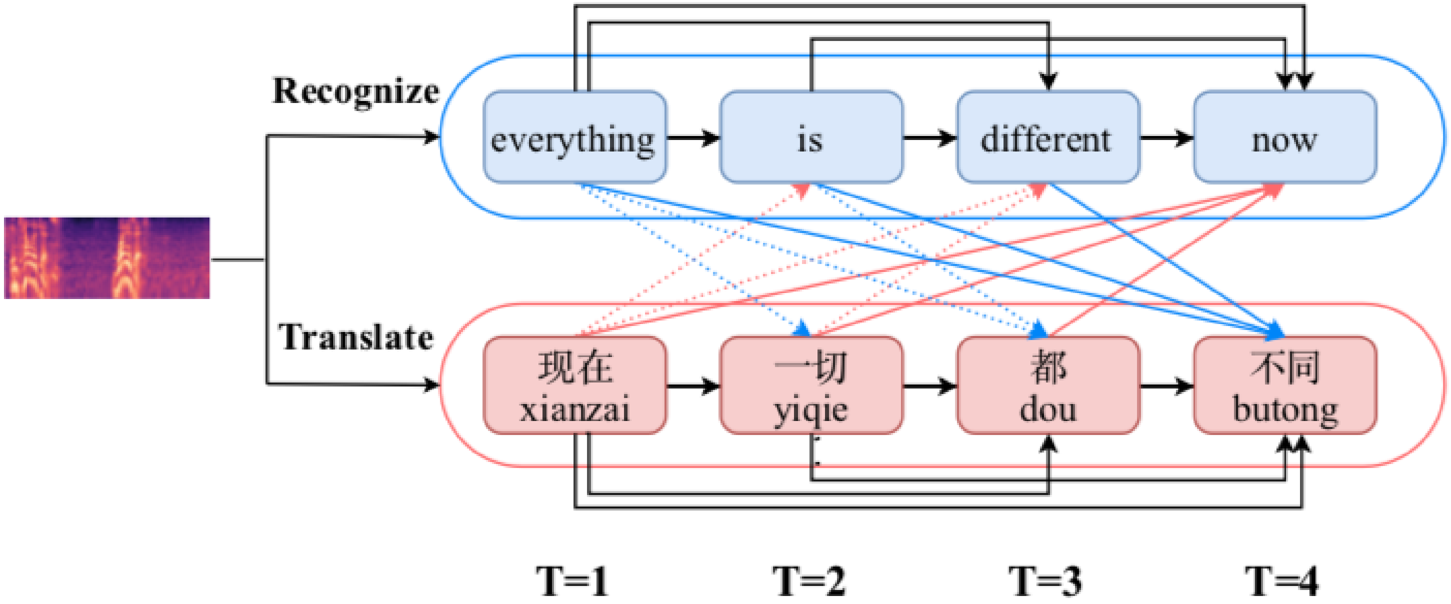
- **Two-Stage Model**

Drawbacks:

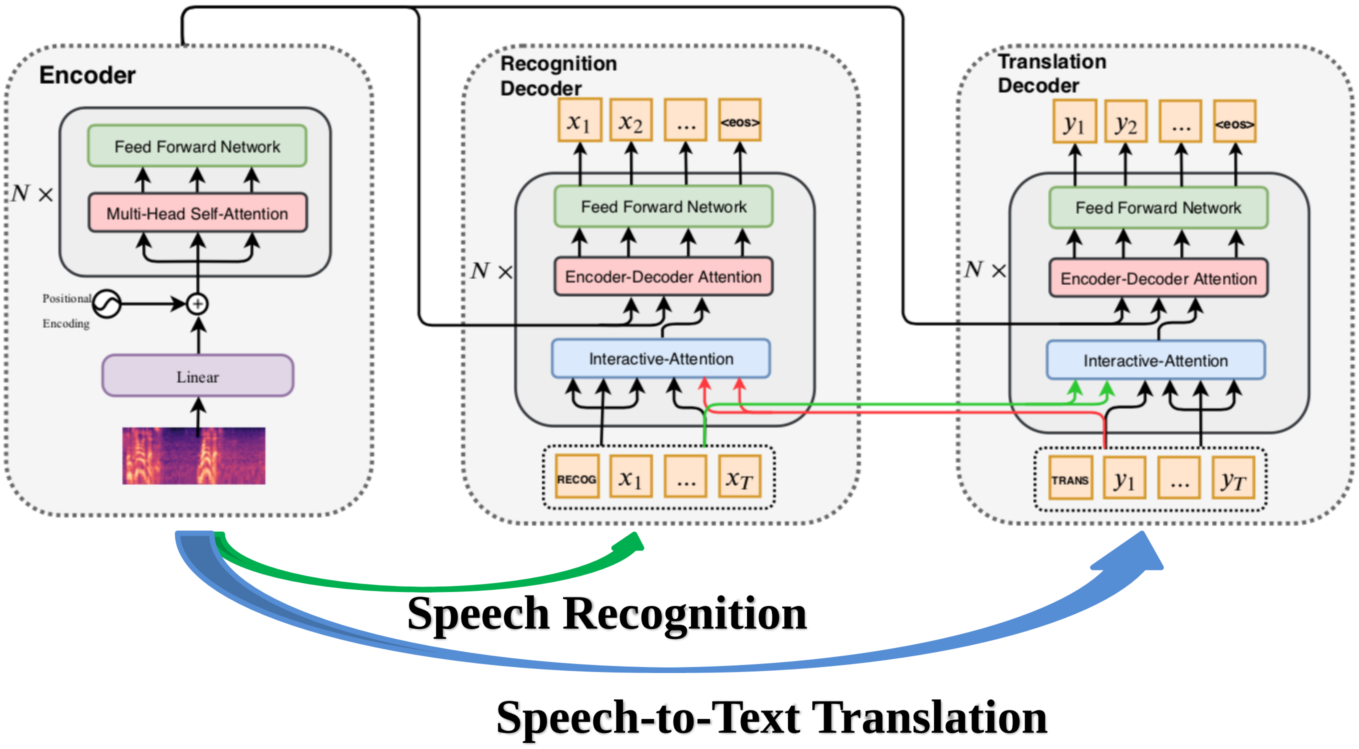
(1) Only the translation decoder can utilize information of recognition decoder.
(2) Translation can only be generated after transcription, leading to a high time delay.

[Sperber et al., 2019] Attention-Passing Models for Robust and Data-Efficient End-to-End Speech Translation. ACL 2019.

Our Solution: Synchronous Speech Recognition and Speech-to-Text Translation

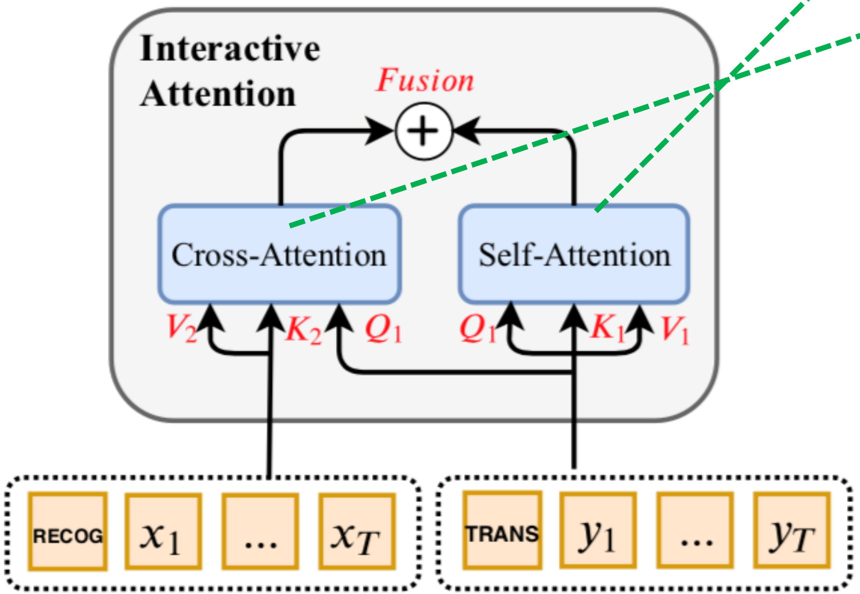


Overall Architecture



133

Interactive Attention



$$\mathrm{H}_1 = Attention(\mathrm{Q}_1, \mathrm{K}_1, \mathrm{V}_1)$$

$$\mathrm{H}_2 = Attention(\mathrm{Q}_1, \mathrm{K}_2, \mathrm{V}_2)$$

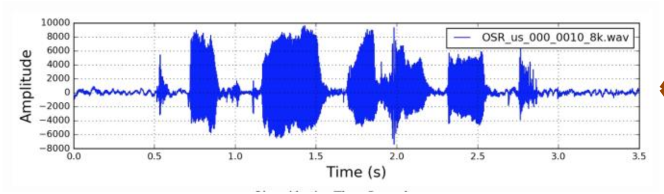
$$\mathrm{H}_{\mathrm{final}} = Fusion(\mathrm{H}_1, \mathrm{H}_2)$$

- **Fusion function**

Linear Interpolation:

$$\mathrm{H}_{\mathrm{final}} = \mathrm{H}_1 + \lambda * \mathrm{H}_2$$

Training



$x_0 \longrightarrow x_1 \longrightarrow \cdots \longrightarrow x_i \quad \cdots \quad \cdots$

Speech Recognition

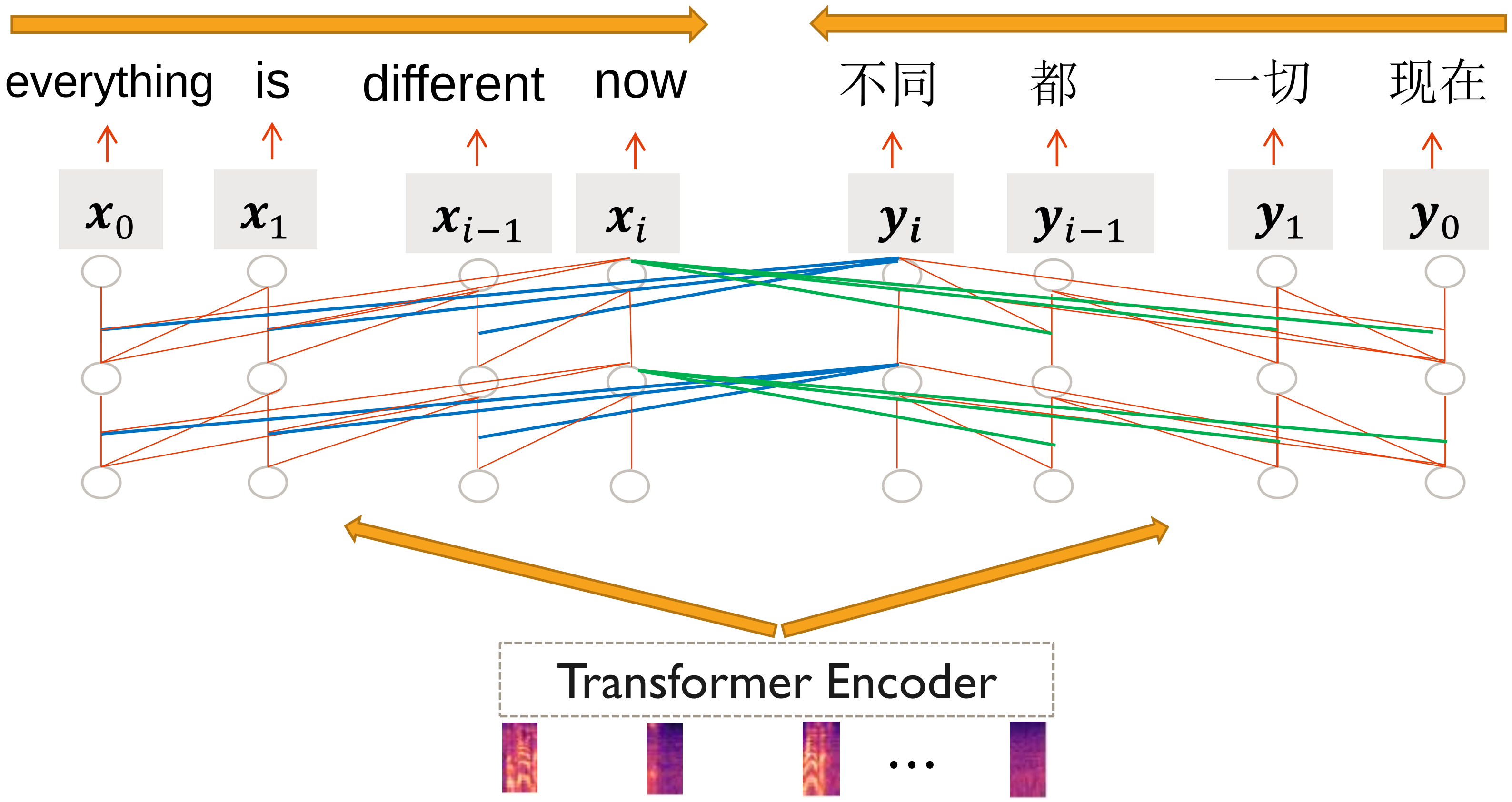
Speech-to-Text Translation

$y_0 \longrightarrow y_1 \longrightarrow \cdots \longrightarrow y_j \quad \cdots \quad \cdots$

- **Object Function:**

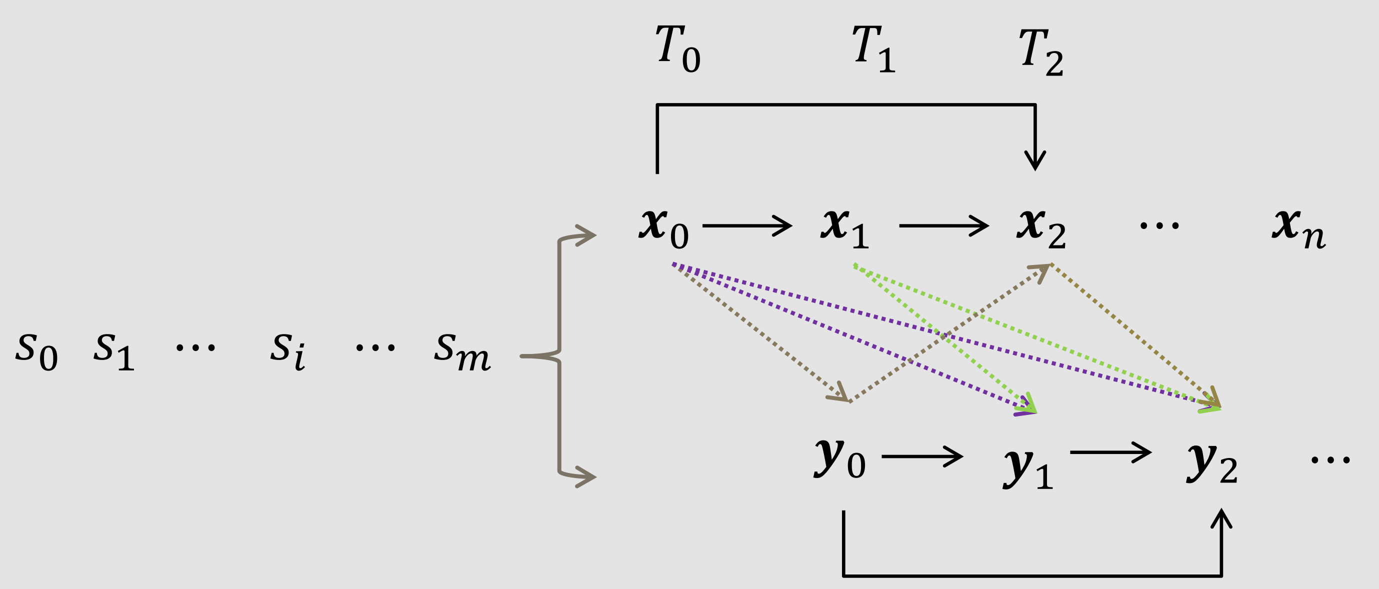
$$L(\theta) = \sum_{j=1}^{|D|} \sum_{i=1}^{n} \{\log p\left(x_i^j \middle| x_{<i}^j, y_{<i}^j, \mathbf{s}^j, \boldsymbol{\theta}_{\mathrm{Rec}}\right) + \log p\left(y_i^j \middle| y_{<i}^j, x_{<i}^j, \mathbf{s}^j, \boldsymbol{\theta}_{\mathrm{Tran}}\right)\}$$

Inference



everything
is
different
now
不同
都
一切
现在
x_0
x_1
x_{i-1}
x_i
y_i
y_{i-1}
y_1
y_0
Transformer Encoder
...

Wait-k Model



Experiments

- **Dataset**

(1)TED multilingual speech translation corpus, with English transcriptions and translations in other languages

(2)Size: En-De/Fr/Zh/Ja (235K/299K/299K/273K)

- **Train details**

(1) *Transformer_base* setting

(2) Transcription: remove punctuations, lowercase and tokenize

Translation: lowercase and tokenize/segment

(2) WER: lowercased, tokenized transcriptions without punctuations

(3) BLEU: case-insensitive tokenized/character BLEU

Experiments

- **Baselines**
  - **Pipeline System:** Transformer ASR + Transformer MT.
  - **Pre-trained ST Model:** Pretrain on ASR, finetune on ST.
  - **Multi-task Model:** ASR + ST with a shared encoder.
  - **Two-stage Model:** The first decoder is used to generate transcription with which the second decoder generates translation.

Experiments

- **Main Results**

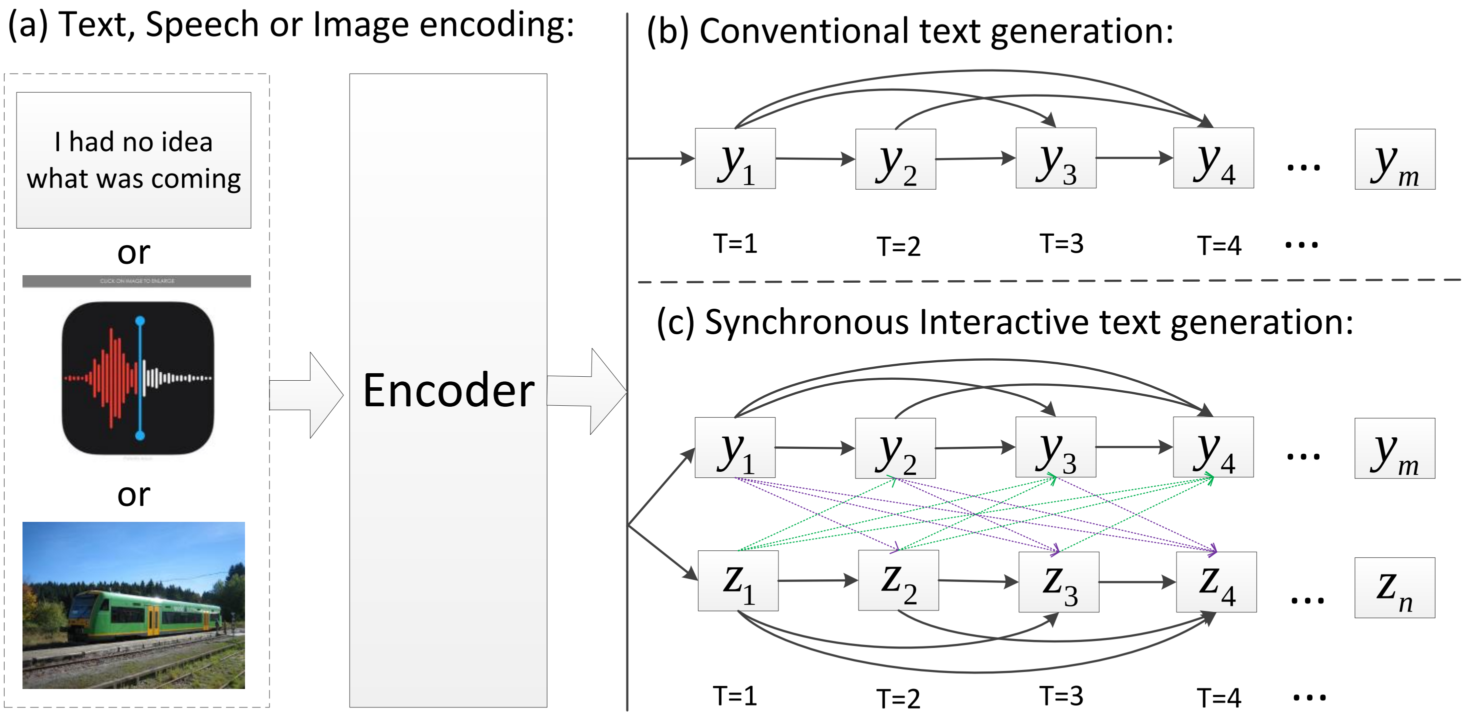
Model	En-De		En-Fr		En-Zh		En-Ja	
	WER	BLEU	WER	BLEU	WER	BLEU	WER	BLEU
Text MT	/	22.19	/	30.68	/	25.01	/	22.93
Pipeline	16.19	19.50	14.20	26.62	14.20	21.52	14.21	**20.87**
E2E	16.19	16.07	14.20	27.63	14.20	19.15	14.21	16.59
Multi-task	15.20	18.08	13.04	28.71	13.43	20.60	14.01	18.73
Two-stage	15.18	19.08	13.34	**30.08**	13.55	20.99	14.12	19.32
Interactive	**14.76**	**19.82**	**12.58**	29.79	**13.38**	**21.68**	**13.91**	20.06

Experiments

- **Effect of k in wait-k model on En-Zh**

Delay	**Dev**		**Test**	
	WER	BLEU	WER	BLEU
Delay-0	14.51	16.28	13.24	21.01
Delay-1	14.29	16.09	**13.17**	21.30
Delay-3	**14.24**	**16.74**	13.38	**21.68**
Delay-5	14.36	16.55	13.51	21.45

Beyond Bidirectional Inference



Section 5: Summary & Future Work

Summary & Future Work

- **Autoregressive Generation Methods**

- History Enhanced Decoding
- Future Enhanced Decoding
- Constrained Decoding
- Memory Enhanced Decoding
- Structured Decoding
- Multi-Pass decoding
- Fast Decoding
- …

- How to realize controllable text output?
- How to compress model and accelerate decoding?

Summary & Future Work

- **Non-Autoregressive Generation Methods**

- **Key Points**
  (1) Leverage Knowledge Distillation
- **Challenges**
  (1) Determine Output length
  (2) Enhance Decoder Input
  (3) Model Target Dependency
- **Problems**
  (1) Multi-Modality Problems
  (2)Under-Translation
  & Repeat-Translation

- How to model target dependency?
- How to leverage monolingual data?
- How to help NAT outperform AT?

Summary & Future Work

- **Bidirectional Generation Methods**

  - Bidirectional Inference: Improve Quality
  - Bidirectional Inference: Improving Efficiency
  - Interactive Inference: Multilanguage Translation
  - Interactive Inference: Recognition & Translation

  - How to perform efficient training without generating pseudo parallel instances?
  - How to generalize the interactive inference idea into multi-task problems in which three or more tasks are concerned?

Thanks

Jiajun Zhang

Shujiu Liu

Yining Wang

Yuchen Liu

Thanks for your attentions!

Any questions?