Non-Autoregressive Models for Fast Sequence Generation

Yang Feng and Chenze Shao

EMNLP 2022

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

中国科学院大学
University of Chinese Academy of Sciences

General Information

Yang Feng
Professor at ICT/CAS
Homepage

Chenze Shao
Ph.D student at ICT/CAS

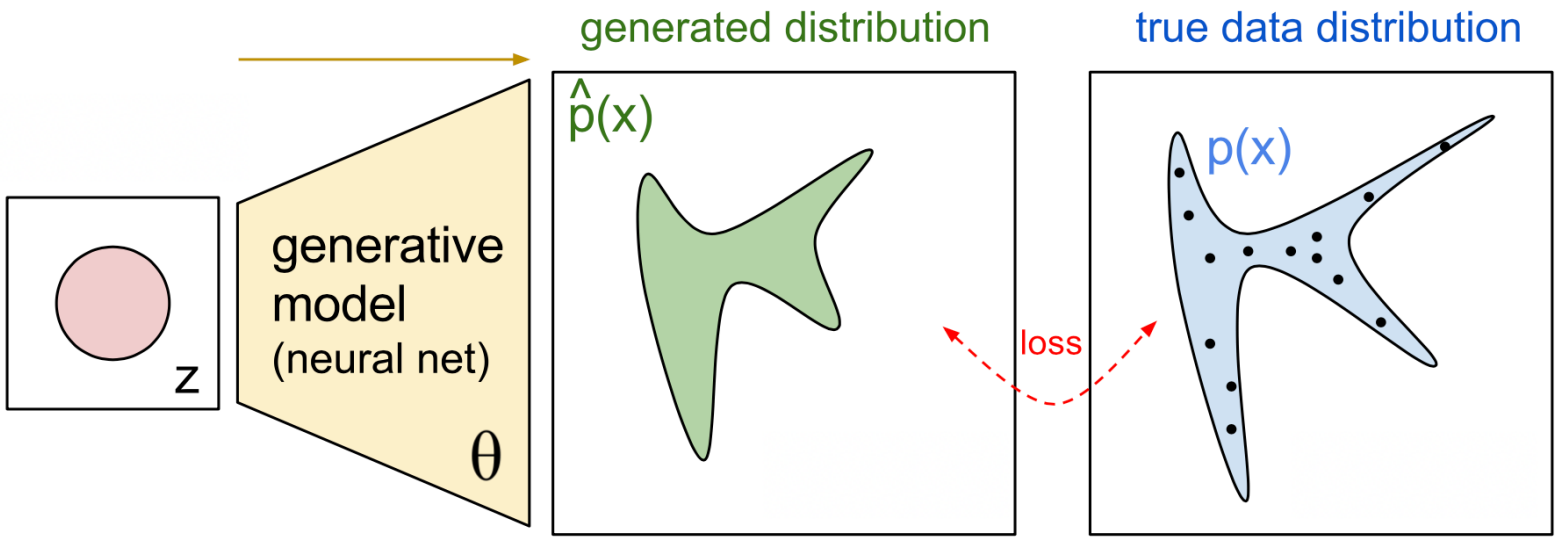
Slides are available online at: https://github.com/ictnlp/nar-tutorial

Outline

- Introduction (Yang Feng)
- Non-Autoregressive Translation (Yang Feng)
- Non-Autoregressive Generation (Chenze Shao)
- Conclusion and Future Direction (Chenze Shao)

Generative Model

- Given data x, learn $p(x)$.



Autoregressive Generation

- Decompose x into a sequence $x_{1:T}$

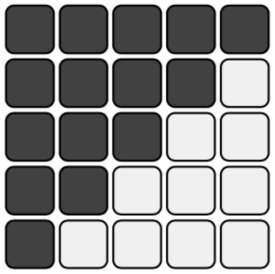
$$p(\boldsymbol{x}) = \prod_{t=1}^{T} p(x_t | x_1, x_2, ..., x_{t-1}) = \prod_{t=1}^{T} p(x_t | \boldsymbol{x}_{<t})$$

- Autoregressive (AR) generation
  - Language modeling, machine translation, speech recognition…
  - Unconditional $p(x)$ or conditional $p(y|x)$

Model Architecture

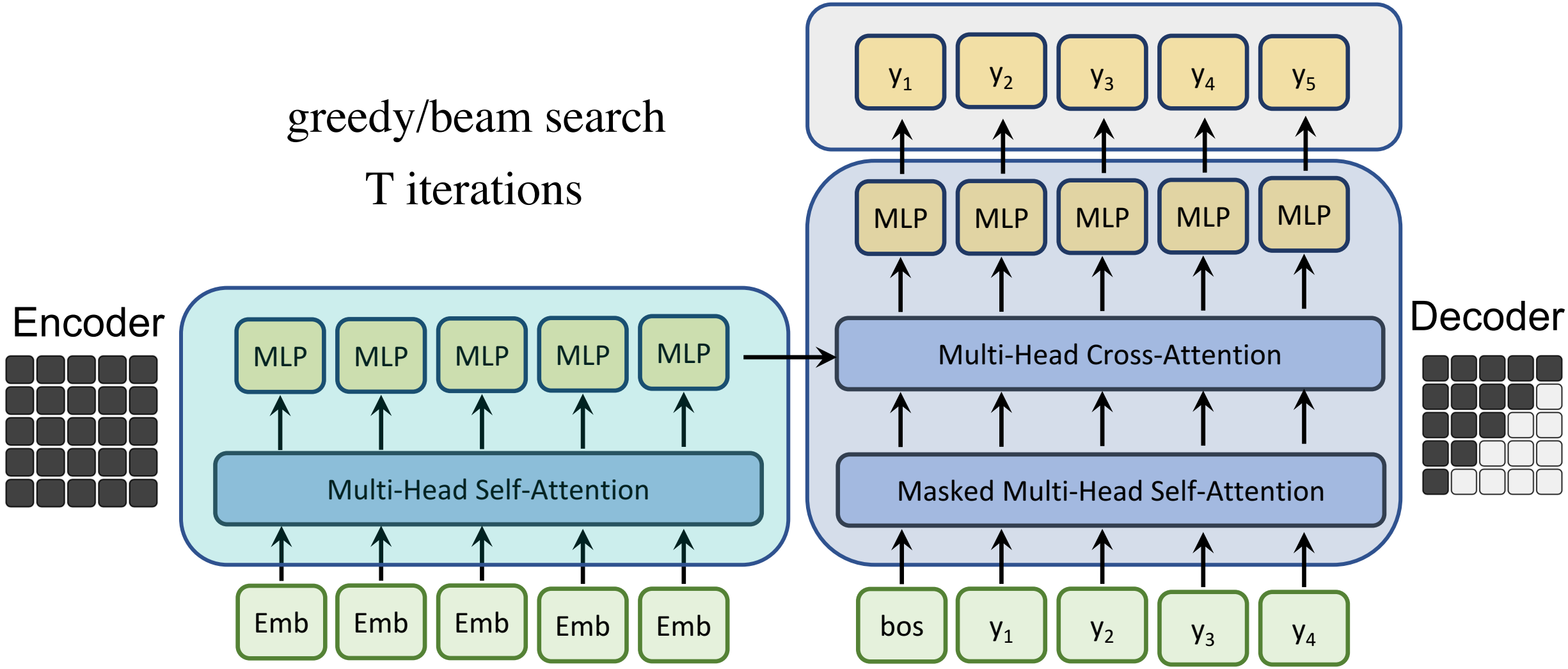
- Model architectures for sequence modeling
  - Parallelizable
    - CNN
    - Self-attention
  - Non-parallelizable
    - RNN
- Transformer based on self-attention
  - Encoder-decoder framework
  - causal decoder

self-attention mask of decoder



Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. NIPS 2017.

Autoregressive Generation (Transformer)

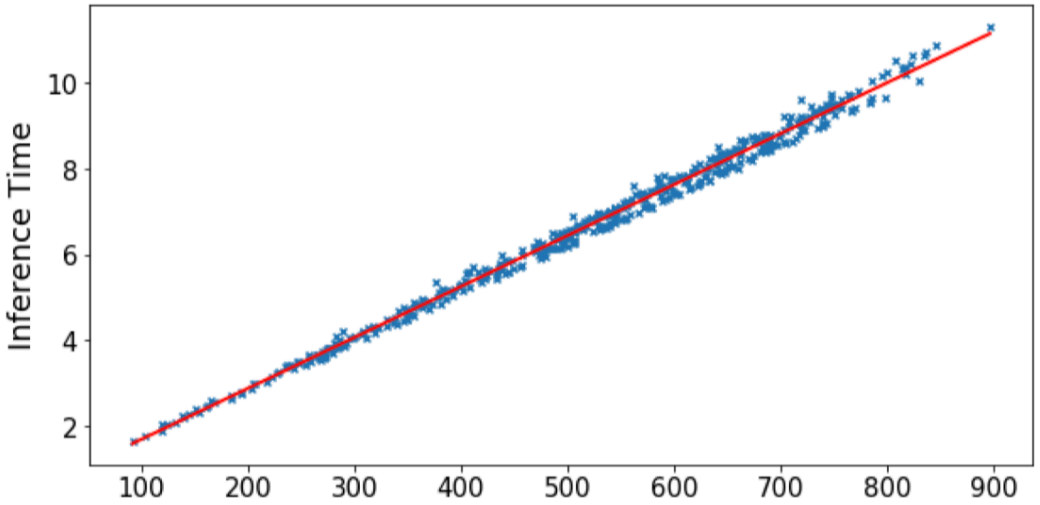


Problems of AR Generation

- Parallelizable during training but non-parallelizable at inference

Models	Training	Inference
RNNs based	×	×
CNNs based	√	×
Self-Attention based	√	×

- Inference time linearly grows with sequence length



Guo J, Tan X, He D, et al. Non-autoregressive neural machine translation with enhanced decoder input. AAAI 2019.
Ren Y, Ruan Y, Tan X, et al. Fastspeech: Fast, robust and controllable text to speech. NIPS 2019.

Problems of AR Generation

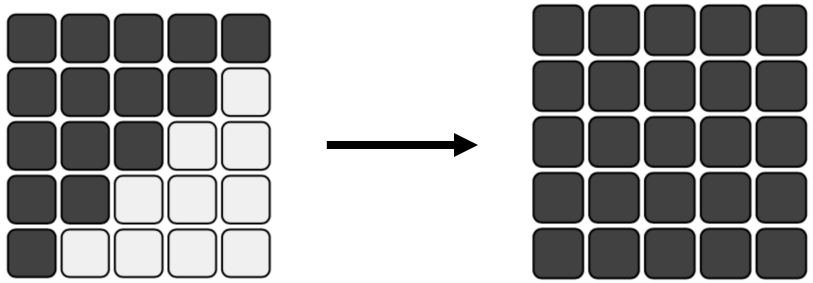
- Other concerns on AR generation
  - Exposure bias
    - Data distribution in training and predicted distribution in inference
  - Error propagation
    - Error is accumulated along the autoregressive generation

Non-Autoregressive Generation

- Generate all tokens in parallel

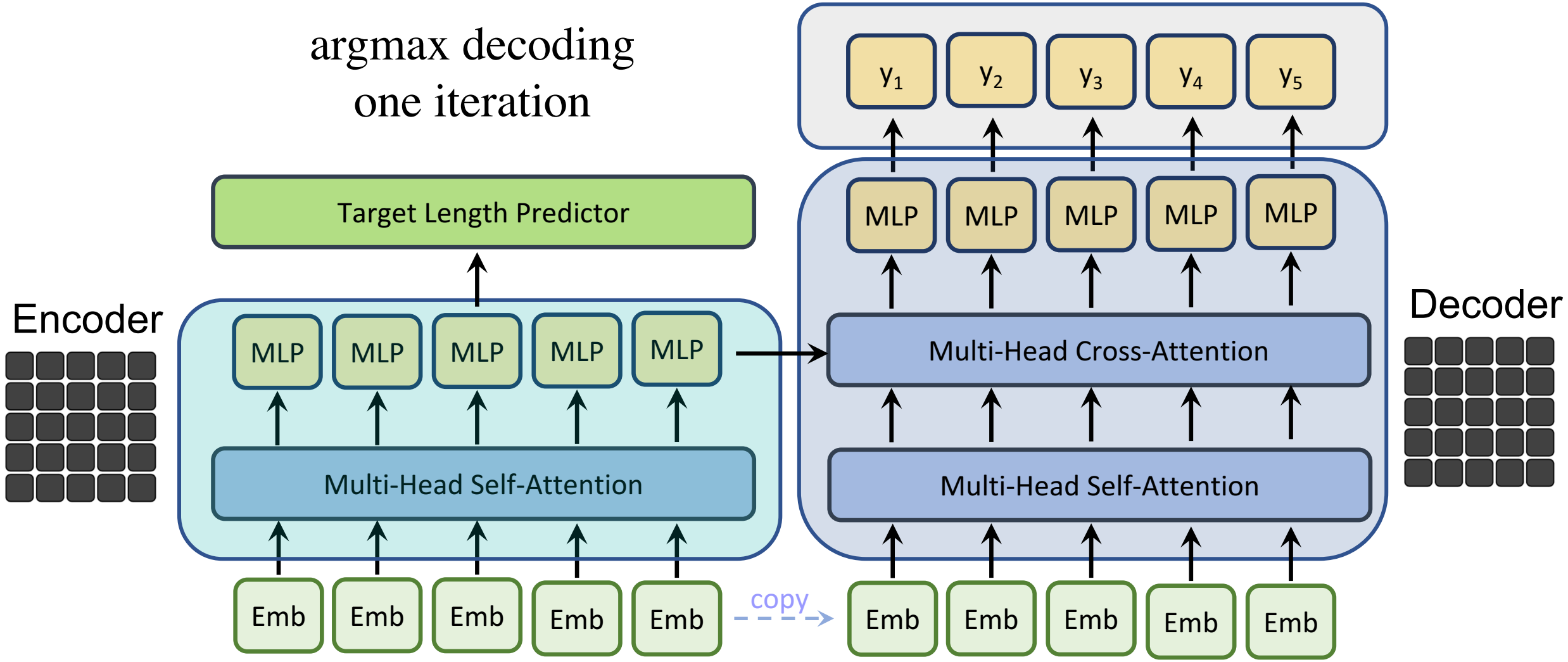
$$p(\boldsymbol{y}|\boldsymbol{x},\theta) = p_L(T|\boldsymbol{x},\theta) \cdot \prod_{t=1}^{T} p(y_t|\boldsymbol{x},\theta)$$

- Fully-visible attention for decoder



Gu J, Bradbury J, Xiong C, et al. Non-autoregressive neural machine translation. ICLR 2018.

Non-Autoregressive Generation



Non-Autoregressive Generation

- Vanilla non-autoregressive model
  - Repeated tokens
  - Missing information
- Machine translation, WMT14 English-German

Model	Transformer	NAR	NAR+KD
BLEU	27.30	11.40	19.50

Source	Es gibt Krebsarten , die aggressiv und andere , die indolent sind .
Reference	There are aggressive cancers and others that are indolent .
AT	There are cancers that are aggressive and others that are indolent .
NAT	There are cancers cancer aggressive aggressive others are indindent .

Gu J, Kong X. Fully non-autoregressive neural machine translation: Tricks of the trade. Findings of ACL 2021.
Shao C, Feng Y, Zhang J, et al. Sequence-level training for non-autoregressive neural machine translation. Computational Linguistics 2021.

Multi-Modality Problem

- Multi-modality problem:
  - Multiple correct outputs for the same source input

Input:	我今天下午吃了披萨
Output1:	I ate pizza this afternoon
Output2:	this afternoon I ate pizza

- Multi-modality degree in generation tasks
  - Low: speech recognition
  - Middle: machine translation
  - High: language modeling
- NAR models are more susceptible to multi-modality.

Multi-Modality Problem

- Maximum Likelihood Estimation (MLE), i.e. Cross-Entropy, training loss:

$$\mathcal{L}_{MLE}(\theta) = -\sum_{t=1}^{T} \log(p_t(y_t|\boldsymbol{X}, \theta))$$

- Word-by-word matching

Reference:	I	ate	pizza	this	afternoon
NAR output:	this	afternoon	I	ate	pizza

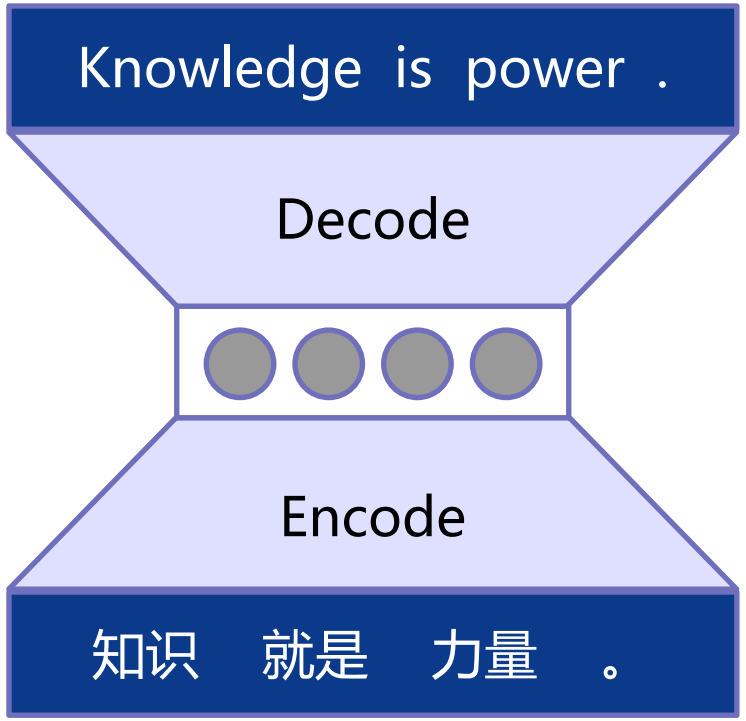
- MLE loss aggravates multi-modality problem.

Outline

- Introduction (Yang Feng)
- Non-Autoregressive Translation (Yang Feng)
- Non-Autoregressive Generation (Chenze Shao)
- Conclusion and Future Direction (Chenze Shao)

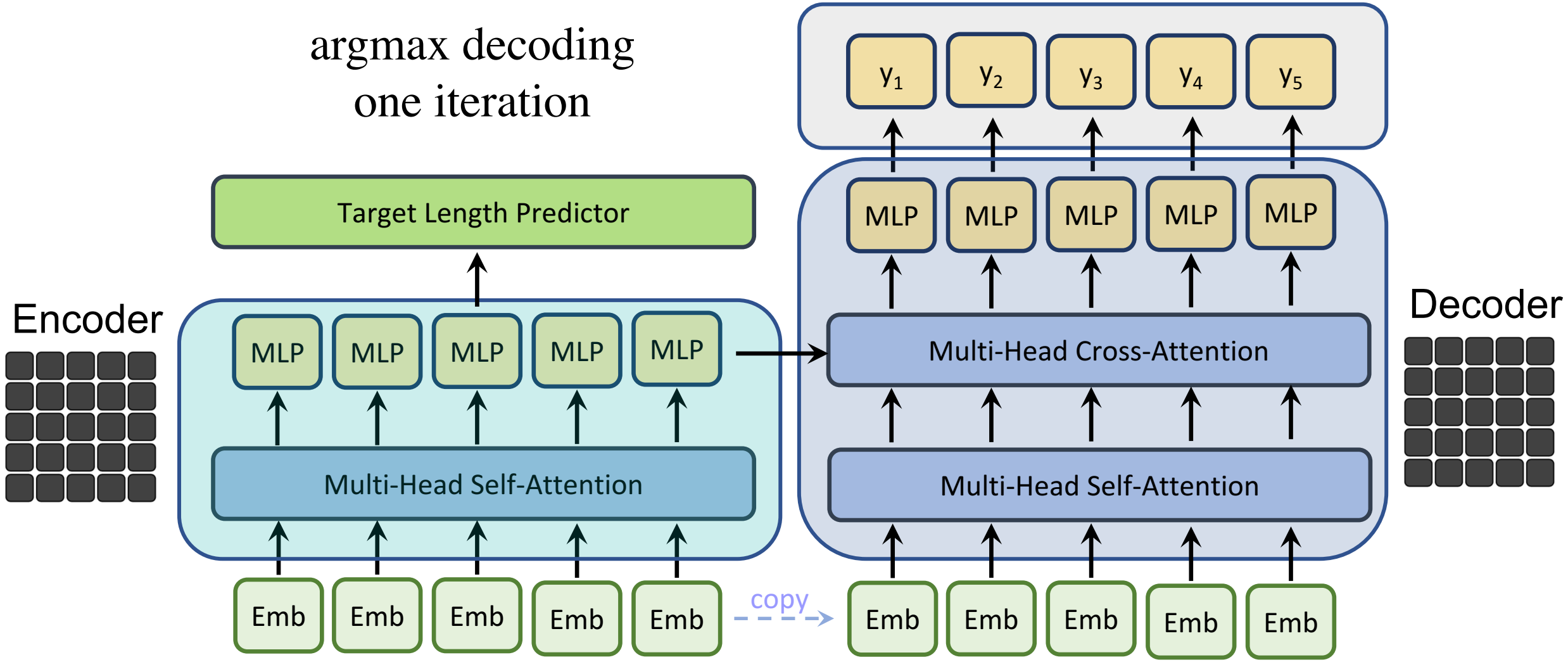
Machine Translation

- Definition:
  - Automated translation of text performed by a computer
- Task formulation:
  - seq2seq generation
- Architecture:
  - Encoder-decoder framework
  - RNN, CNN, Transformer
- Evaluation metrics: BLEU
  - Precision of matching n-grams



Encoder-decoder framework

Non-Autoregressive Generation



Overview

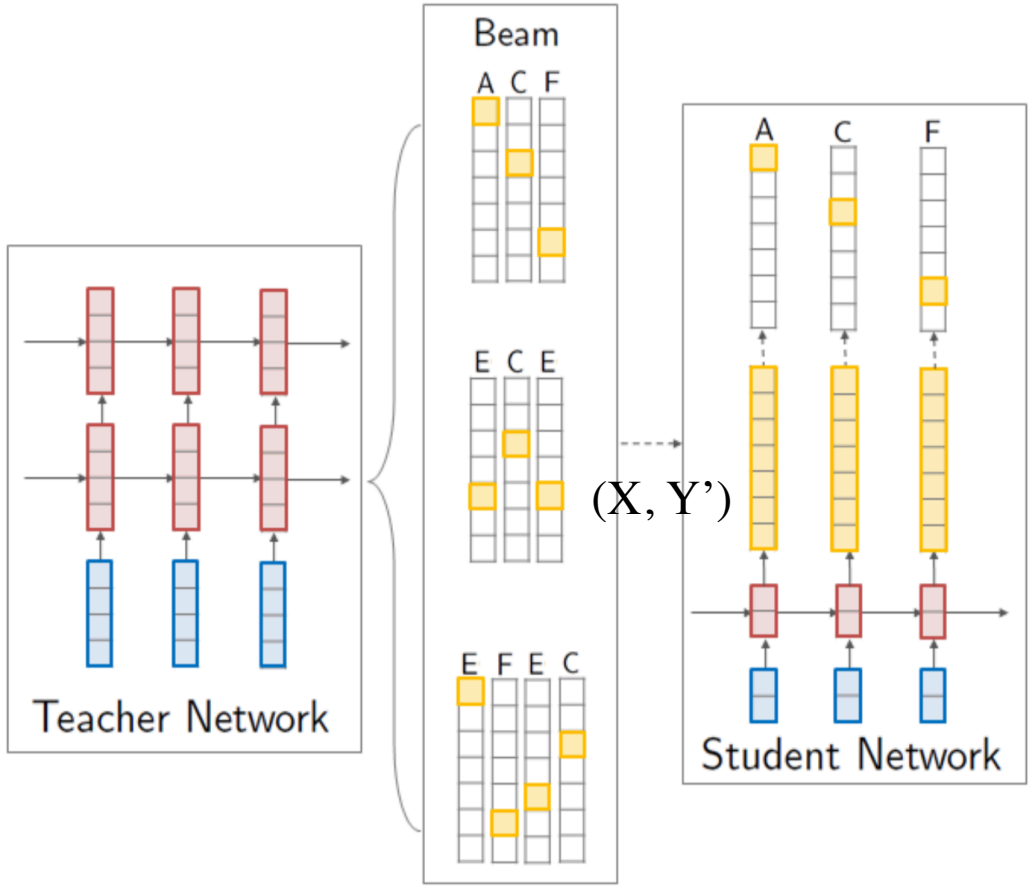
- Multi-modality problem
- Solutions:

Methods	Descriptions
Knowledge distillation	Reduce multi-modality of training data
Latent modeling	Reduce multi-modality with latent information
Expressive ability	More expressive NAR models
Training objectives	Replace MLE with cross-word objectives
Decoding approaches	Decoding approaches besides fully NAR decoding

Vanilla Knowledge Distillation

- Reduce multi-modality in the training dataset (X, Y)

- Sequence-Level Knowledge Distillation
  - 1. train an AR teacher model on (X, Y)
  - 2. use the teacher to decode X, obtain Y'
  - 3. train the NAR student on (X, Y')



Model	NAT	NAT+KD
BLEU	11.40	19.50

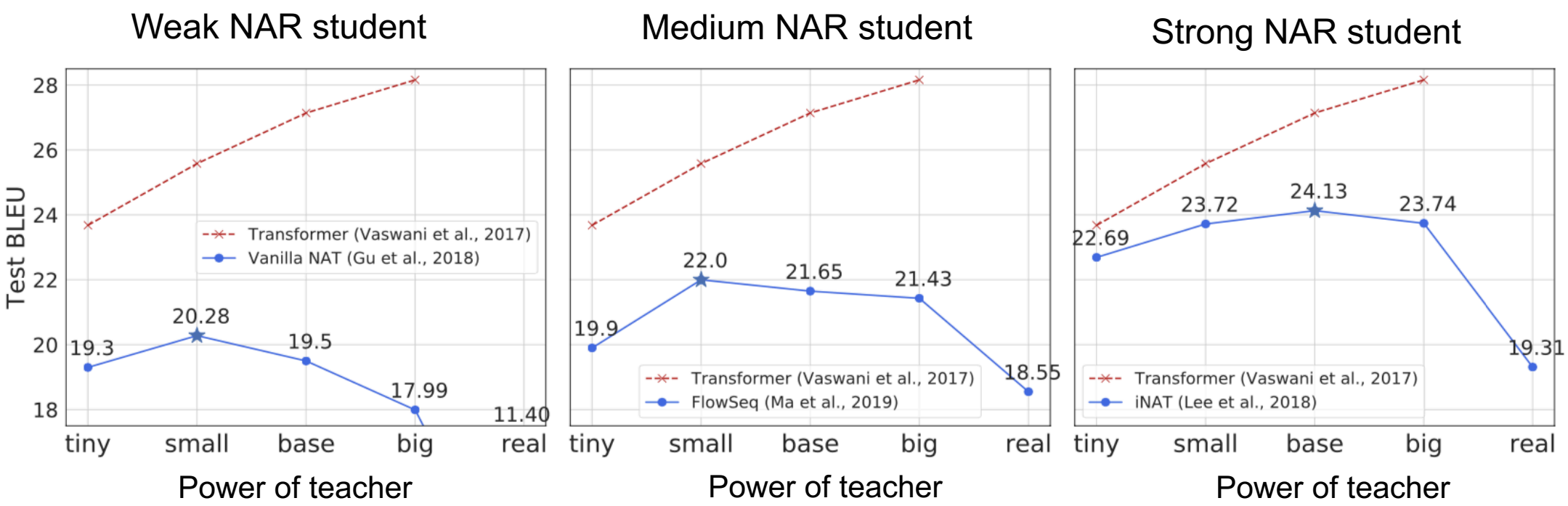
Kim Y, Rush A M. Sequence-level knowledge distillation. arXiv preprint arXiv:1606.07947, 2016.

Vanilla Knowledge Distillation

- Why does knowledge distillation (KD) work?
  - Lower diversity of generated training data (X, Y')
    - same input, same output
  - Easier training with the generated training data
    - More proper evaluation with training loss

Understanding Knowledge Distillation

- Do we always need a strong teacher model?



Teacher model should match the ability of NAR student

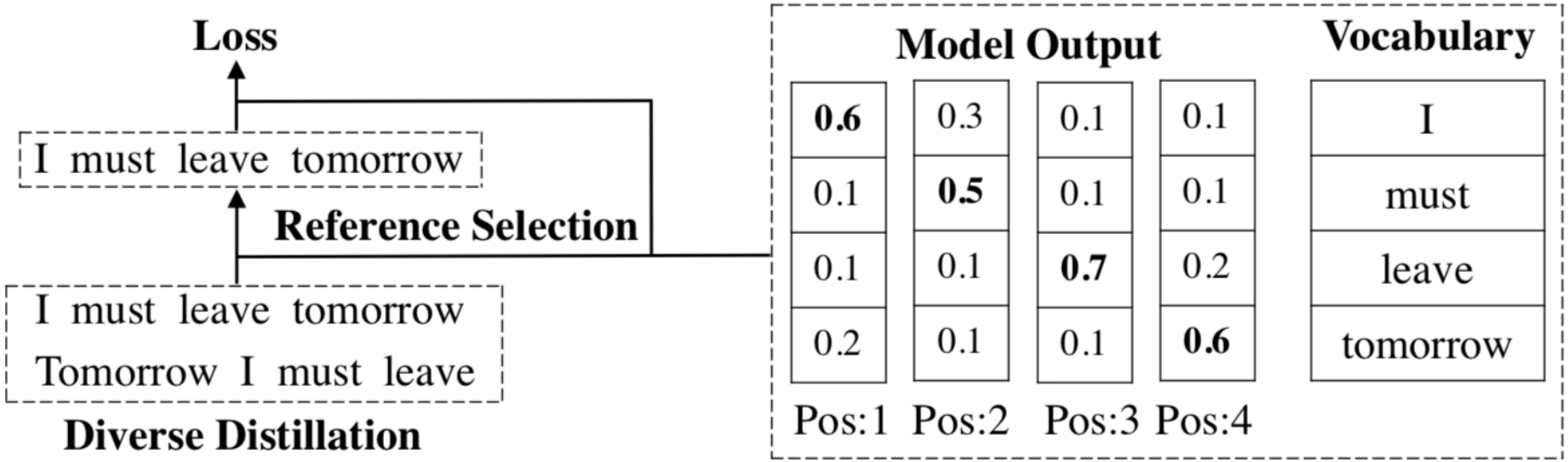
Problems of Knowledge Distillation

- Problems:
  - Diversity still exists in the training data.
  - Hard to get the teacher matching with the student.

- Refined knowledge distillation (KD)
  - Diverse Distillation
  - KD with Rephraser
  - EM-NAR

Diverse Distillation

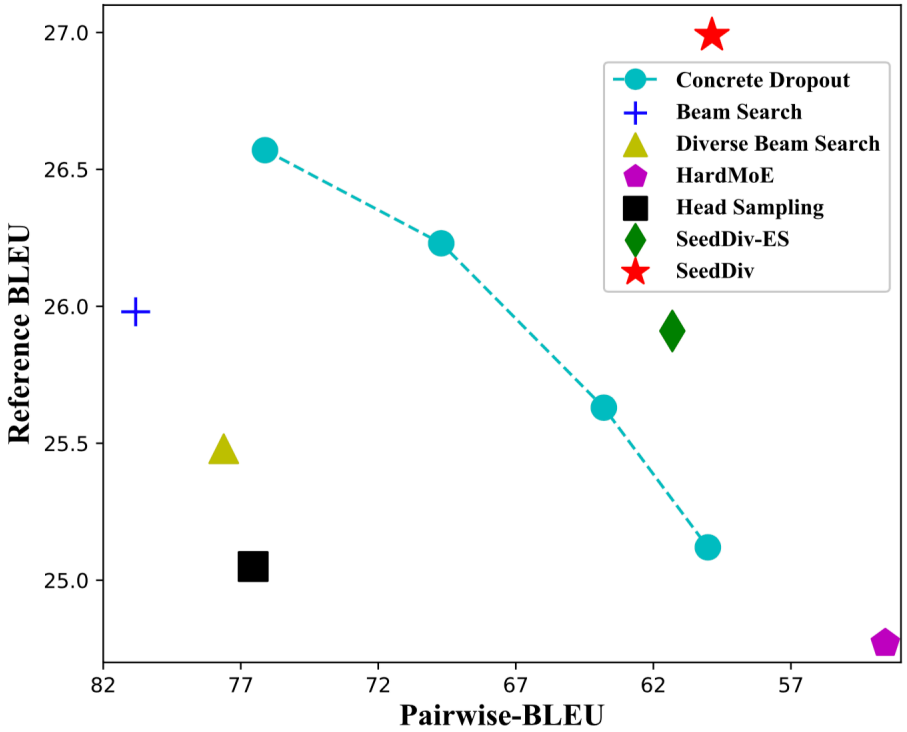
- Problem: diversity in the training data
- Solution 1: Diverse Distillation
  - Dynamically reference selection
  - Diverse references provided by different teachers



Shao C, Wu X, Feng Y. One Reference Is Not Enough: Diverse Distillation with Reference Selection for Non-Autoregressive Translation. NAACL 2022.

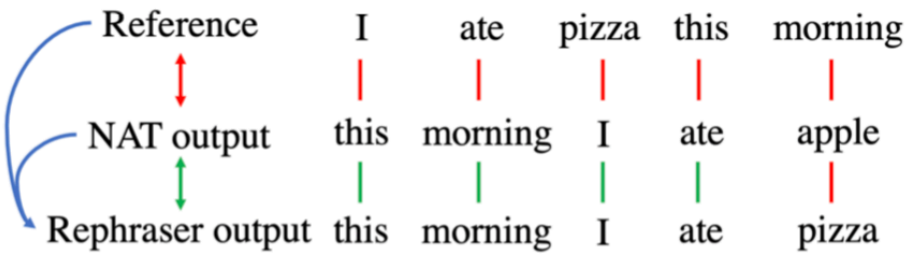
Diverse Distillation

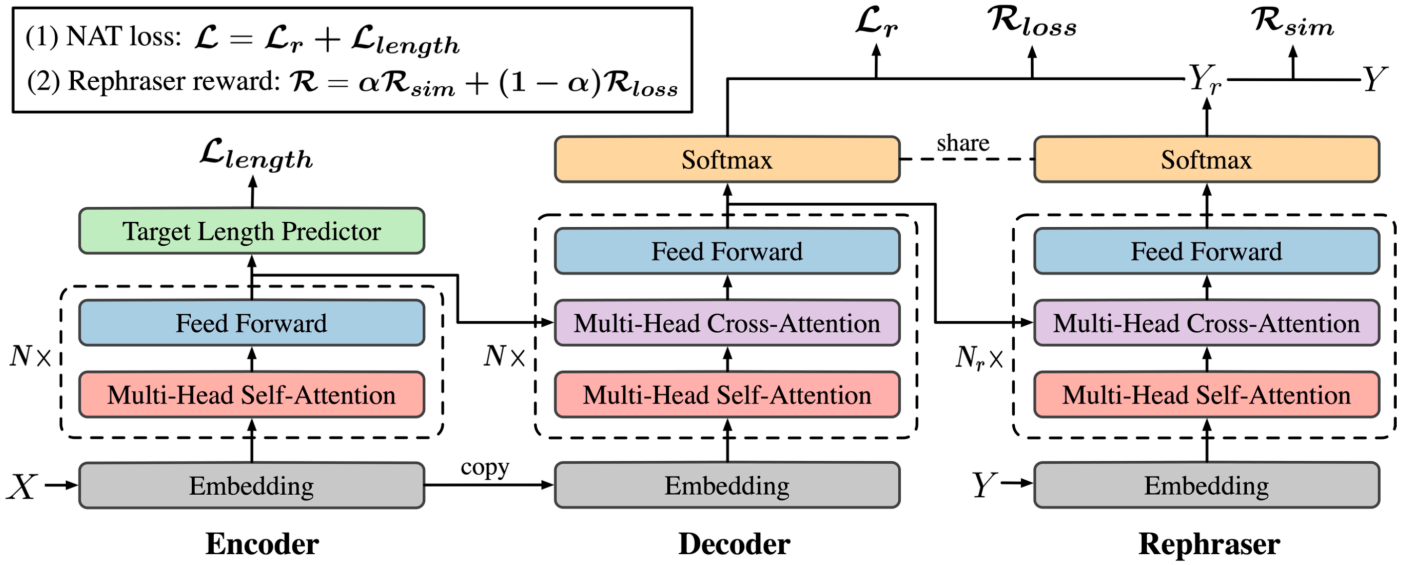
- The generated references are
  - lower pairwise-BLEU → diverse
  - Greater BLEU → high-quality



KD with Rephraser

- Problem: diversity in training data
- Solution 2: introduce training-friendly references generated by rephraser
  - Keep original meaning of reference
  - Easier to train



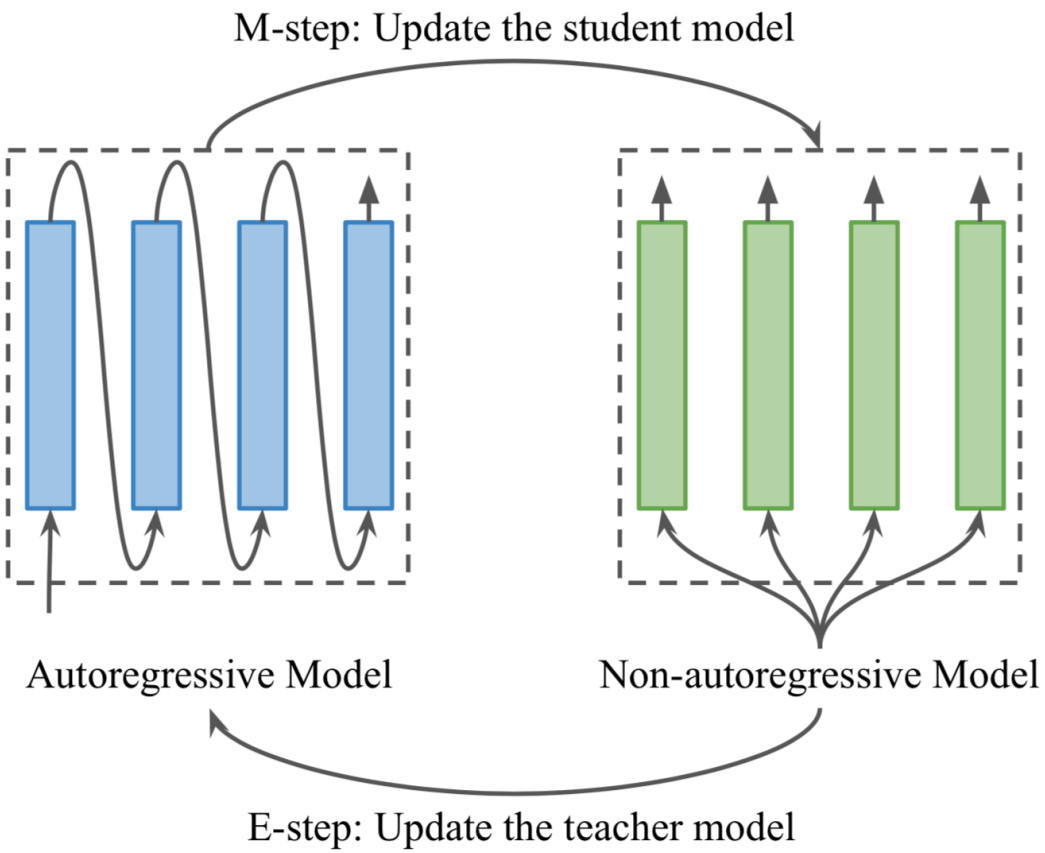


Shao C, Zhang J, Zhou J, Feng Y. Rephrasing the Reference for Non-Autoregressive Machine Translation. AAAI 2023.

EM-NAR

- Problem: keep matching between the teacher and student
- Solution: iteratively update the AR teacher and NAR student by EM

	NAR Model	AR Model
$t = 1$	19.55	**27.84**
$t = 2$	22.27	27.50
$t = 3$	22.85	27.13
$t = 4$	**23.27**	26.78
$t = 5^*$	22.86	26.11
$t = 5$	23.18	26.72
$t = 6$	23.16	26.75



Sun Z, Yang Y. An EM approach to non-autoregressive conditional sequence generation. ICML 2020.

Overview

- Multi-modality problem
- Solutions:

Methods	Descriptions
Knowledge distillation	Reduce multi-modality of training data
Latent modeling	Reduce multi-modality with latent information
Expressive ability	More expressive NAR models
Training objectives	Replace MLE with cross-word objectives
Decoding approaches	Decoding approaches besides fully NAR decoding

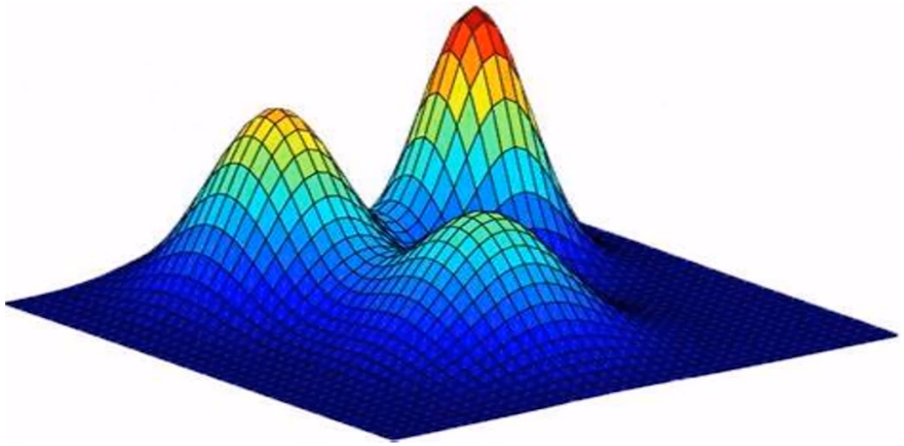
Latent Modeling

- Latent modeling:

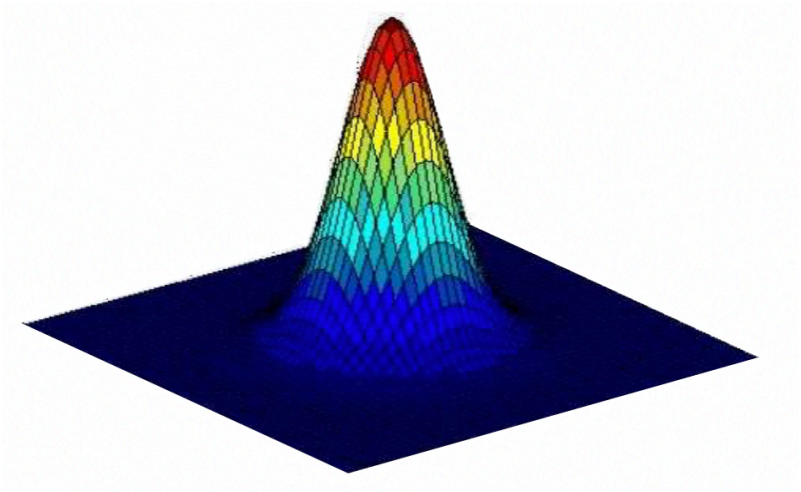
$$p(y|x) = \sum_z p(z|x)p(y|x,z)$$

- Introduce latent variable z to reduce multi-modality

$p(y|x)$

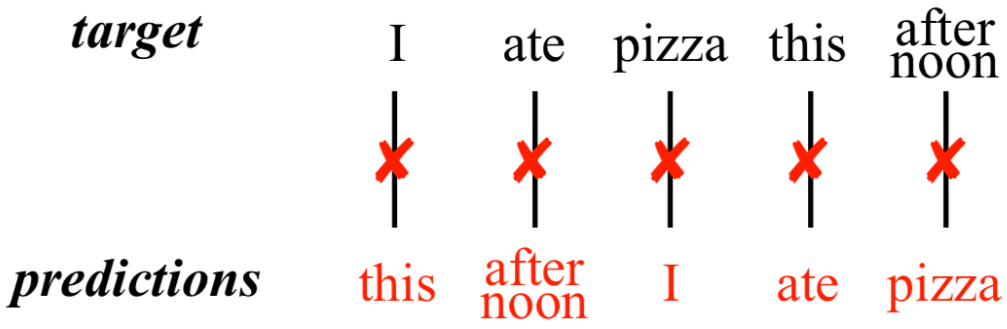


$p(y|x,z)$



Latent Modeling

- MLE loss: word-by-word matching



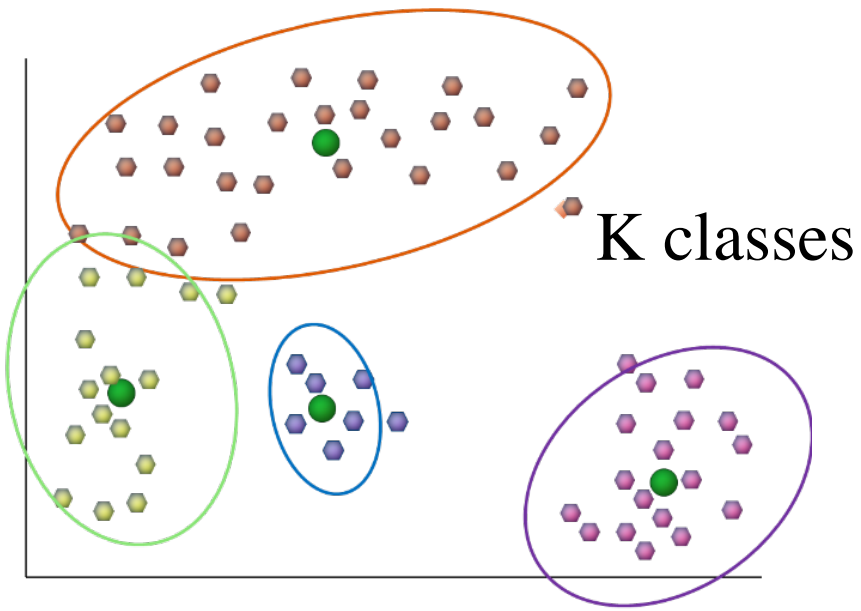
- Latent variables: leak more reference information
  - Training, sample z from posterior $q(z|x,y)$
  - Decoding, sample z from prior $p(z|x)$

Latent Modeling

- Techniques for latent modeling
  - Discrete latent variables:
    - Vector quantization
  - Continuous latent variables:
    - Variational autoencoder (VAE)
  - Manually designed latent information:
    - fertility
    - word alignment
    - partial reference

Vector Quantization

- Introduce:
  - A latent embedding matrix E with K classes
  - A latent variable z
- Working process:
  - Cluster x into a class of E with K-means
  - assign the center of corresponding class to z



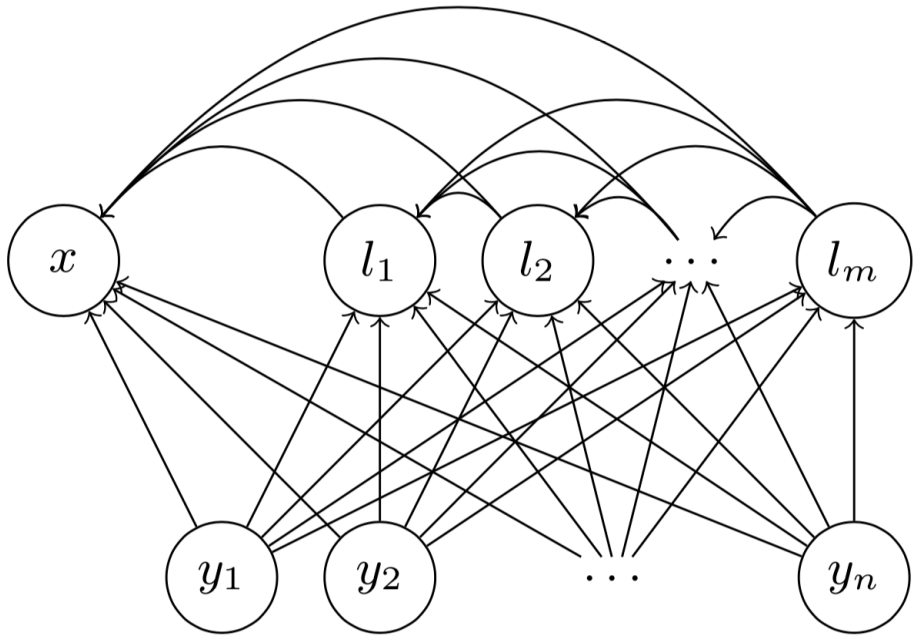
- Training: maximize $p(z|x)$ and $p(y|x,z)$
- Decoding: fitst $p(z|x) \rightarrow z$, then $p(y|x,z) \rightarrow y$

Van Den Oord A, Vinyals O. Neural discrete representation learning. NIPS 2017.

Vector Quantization

- A single latent variable z → latent variable sequence

- Latent Transformer
  - $ae(y)$: autoencode y to a into a shorter sequence of latent variables
  - latent prediction $p(z|x)$: autoregressively predict the latent sequence
  - decoder $p(y|x,z)$: non-autoregressive, predict y in parallel



Kaiser L, Bengio S, Roy A, et al. Fast decoding in sequence models using discrete latent variables. ICML 2018.
Bao Y, Huang S, Xiao T, et al. Non-autoregressive translation by learning target categorical codes. NAACL 2021.

Variational Autoencoder

- Introduce continuous latent variables z
  - Posterior distribution $q(z|x, y)$
  - prior distribution $p(z|x)$

- Training: maximize the evidence lower-bound (ELBO)

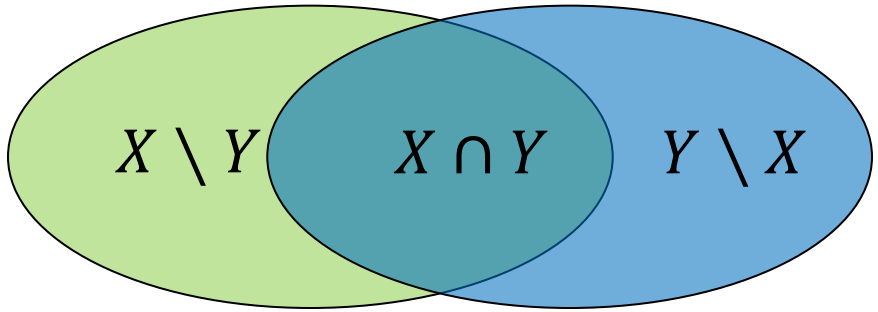
$$\log P_\theta(\mathbf{y}|\mathbf{x}) \geq \mathrm{E}_{q_\phi(\mathbf{z}|\mathbf{y},\mathbf{x})}[\log P_\theta(\mathbf{y}|\mathbf{z},\mathbf{x})] - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{y},\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})).$$

- Decoding: sample z from the prior distribution or use the mean of prior.

Gu J, Kong X. Fully non-autoregressive neural machine translation: Tricks of the trade. Findings of ACL 2021.

Manually Designed Latent Information

- What kind of information should latent variables carry?



Latent variables should carry the information of $Y \setminus X$

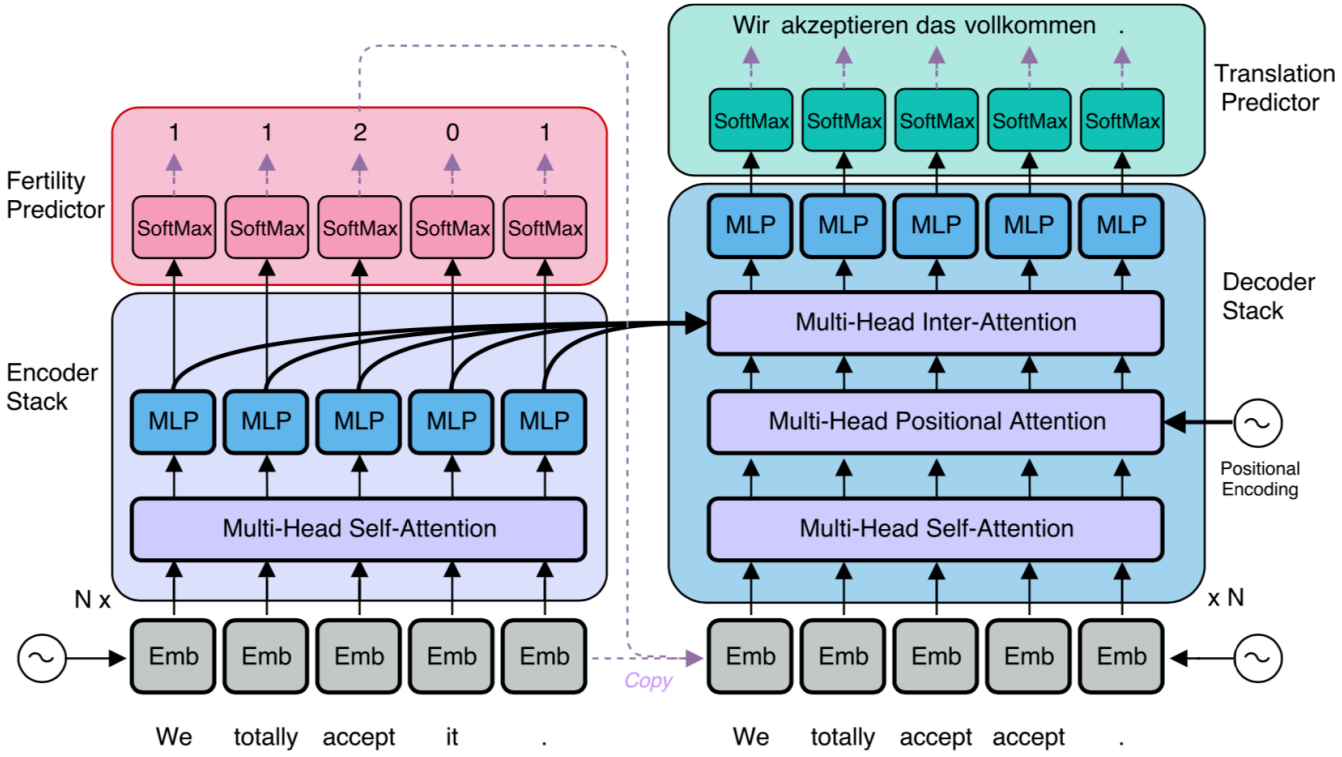
Manually Designed Latent Information

- Target length
  - Use golden length in training, predict the length in inference

- Other latent information to reduce multi-modality
  - Fertility
  - Word alignment
  - Partially masked target sentence

Fertility

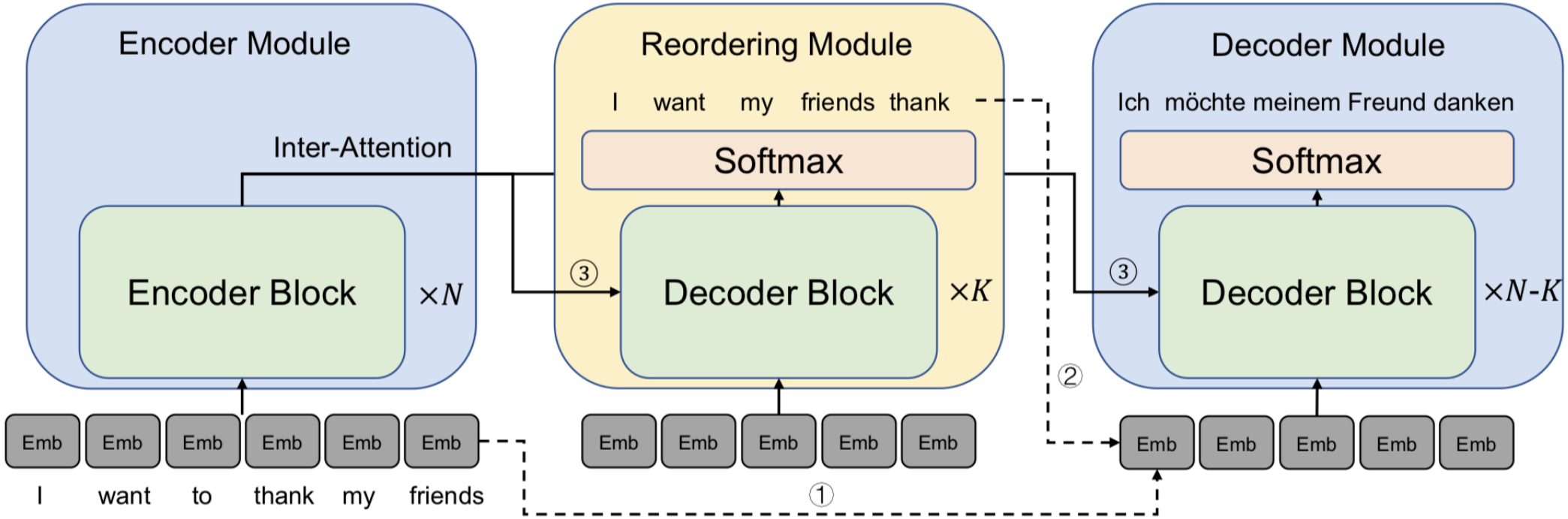
- Fertility: the number of target words that a source token will be translated to
- Supervised training: fast-align as golden; Finetune by RL



Gu J, Bradbury J, Xiong C, et al. Non-autoregressive neural machine translation. ICLR 2018.

Reordering

- Model the reordering information
  - Reordering module: light AR or fully NAR
- Golden reordering: fast-align



Ran Q, Lin Y, Li P, et al. Guiding non-autoregressive neural machine translation decoding with reordering information. AAAI 2021.

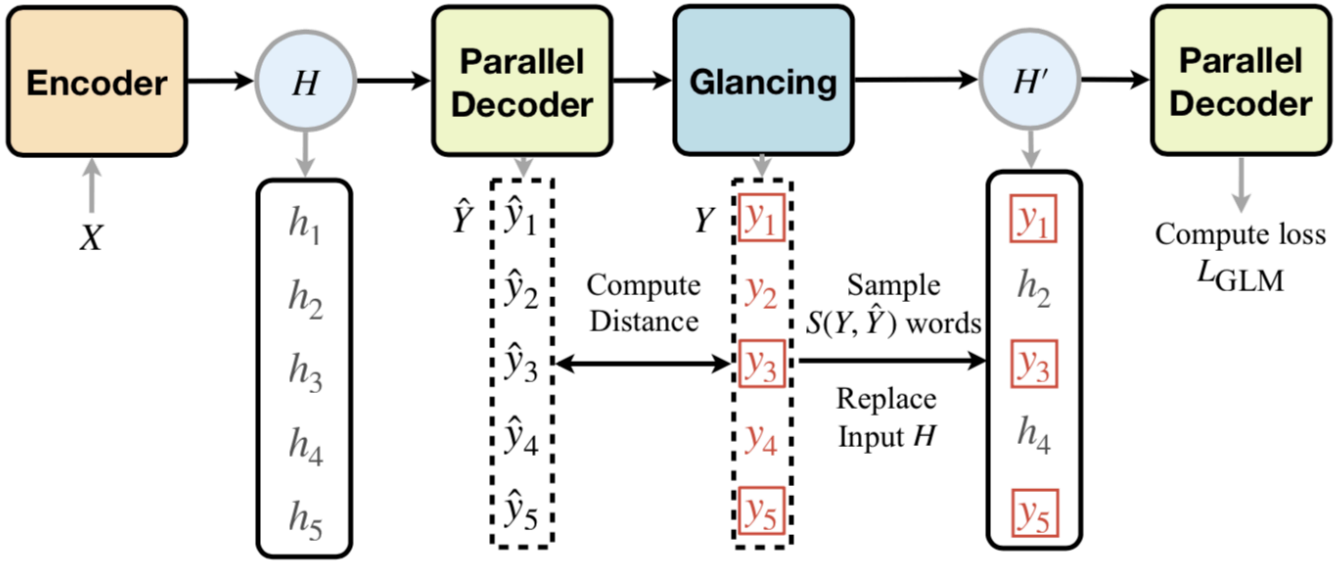
Problem of Latent Modeling

- Problem: latent gap between training and inference
  - Latent from autoencoder:
    - posterior during training, prior at inference
  - Manual latent information
    - golden during training, model prediction at inference

- Solution: glancing training (GLAT)

GLAT

- Mechanism：Mask-predict
- Latent information: unmasked tokens
- Glancing strategy
  - First, full mask
  - Then, determine the mask ratio according to hamming distance



Qian L, Zhou H, Bao Y, et al. Glancing transformer for non-autoregressive neural machine translation. ACL 2021.

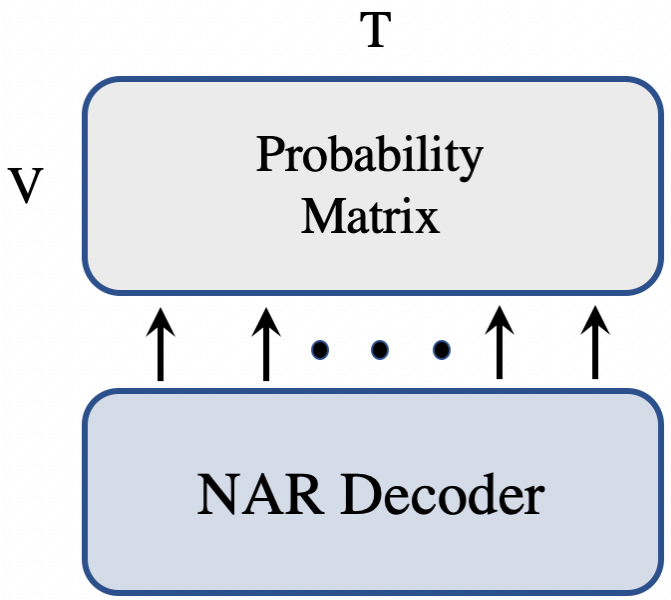
Overview

- Multi-modality problem
- Solutions:

Methods	Descriptions
Knowledge distillation	Reduce multi-modality of training data
Latent modeling	Reduce multi-modality with latent information
Expressive ability	More expressive NAR models
Training objectives	Replace MLE with cross-word objectives
Decoding approaches	Decoding approaches besides fully NAR decoding

Output Space of NAR

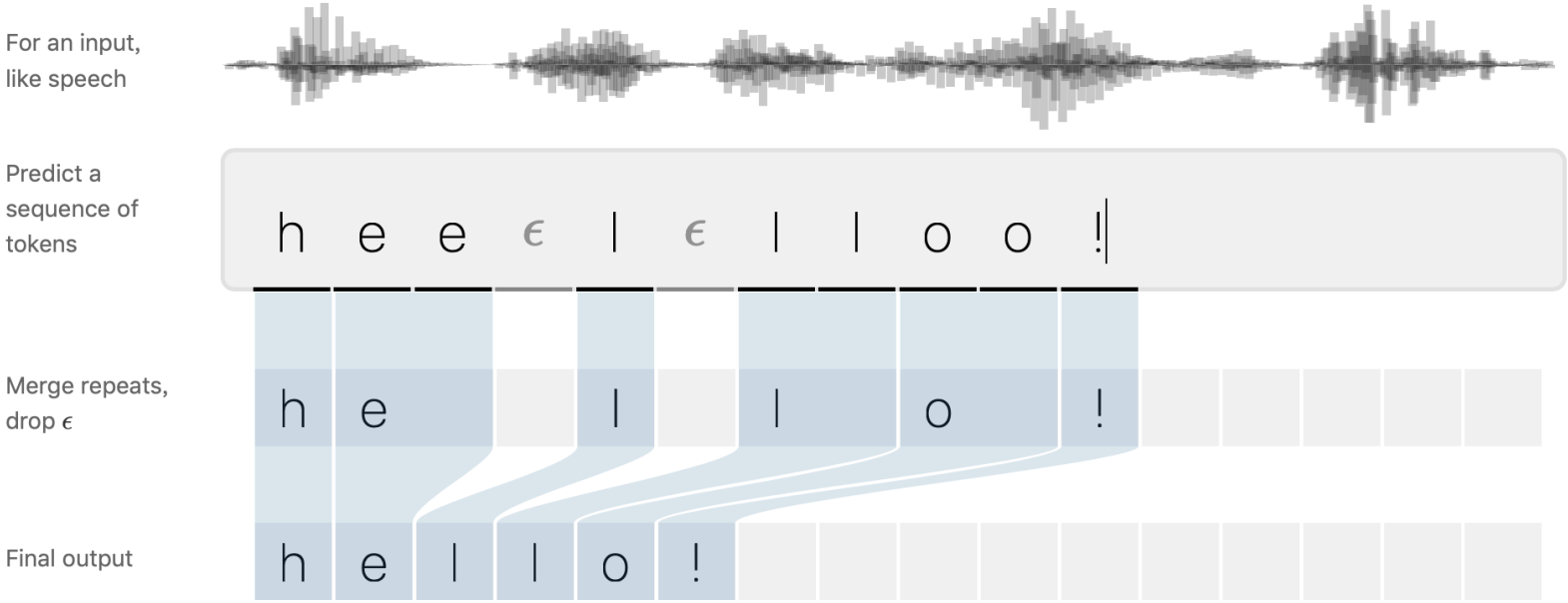
- NAR model is unable to capture the multi-modal distribution
  - Independence assumption
- The output space of NAR model is small
  - Vocubulary size V, target length T, the output space of Vanilla NAR model has demension VT
  - The number of possible outputs is exponential, p_{data} has demension $V^T \gg VT$



- Design more expressive NAR models with larger output spaces
  - CTC
  - Directed Acyclic Transformer (DA-Transformer)

CTC

- Working mechanism
  - Assume longer output than reference
  - Allow to predict blank and repetitive tokens
  - Remove blank and repetitive tokens for final translation



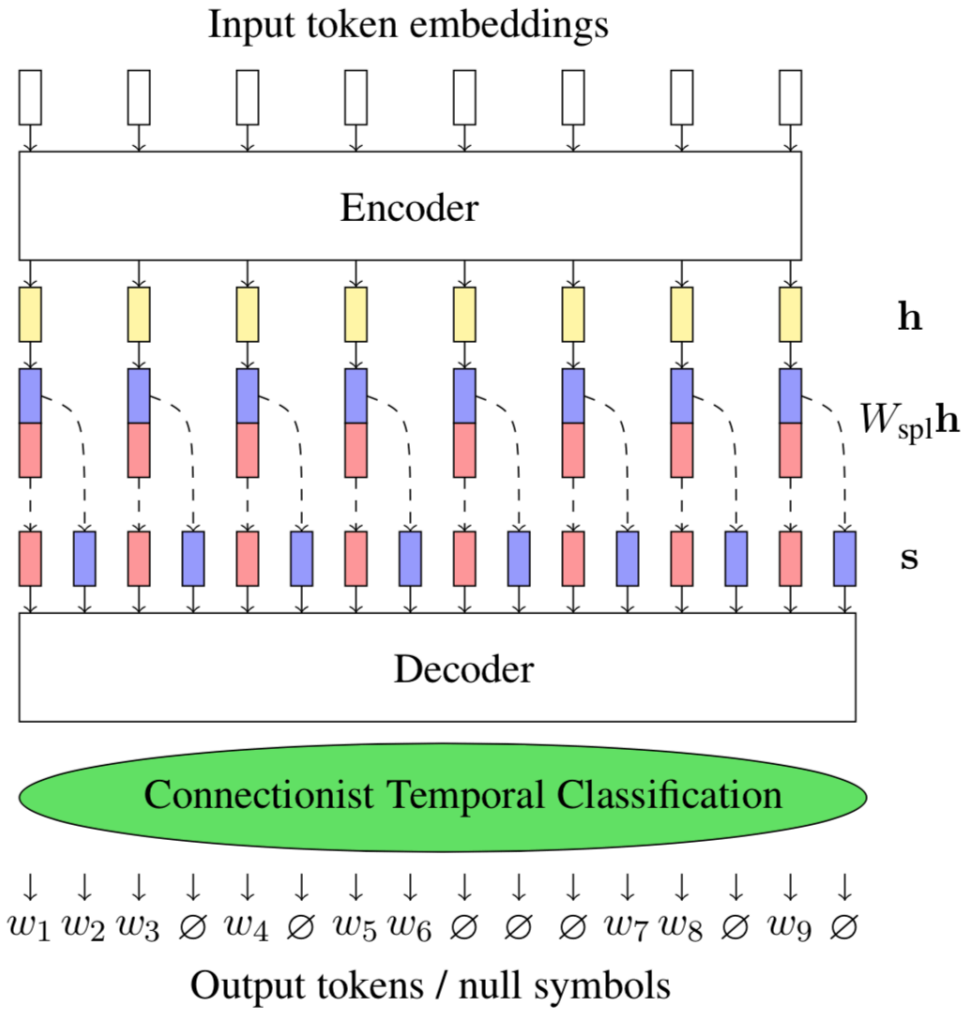
Graves A, Fernández S, Gomez F, et al. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. ICML 2006.

CTC

- Modelling:
  - upsample the encoder outputs
  - Allow to predict blank or repeated tokens
- Training: dynamic programming over all alignments to reference

$$\log p(y|x,\theta) = \log \sum_{a \in \beta(y)} p(a|x,\theta)$$

- Decoding: Argmax or beam search

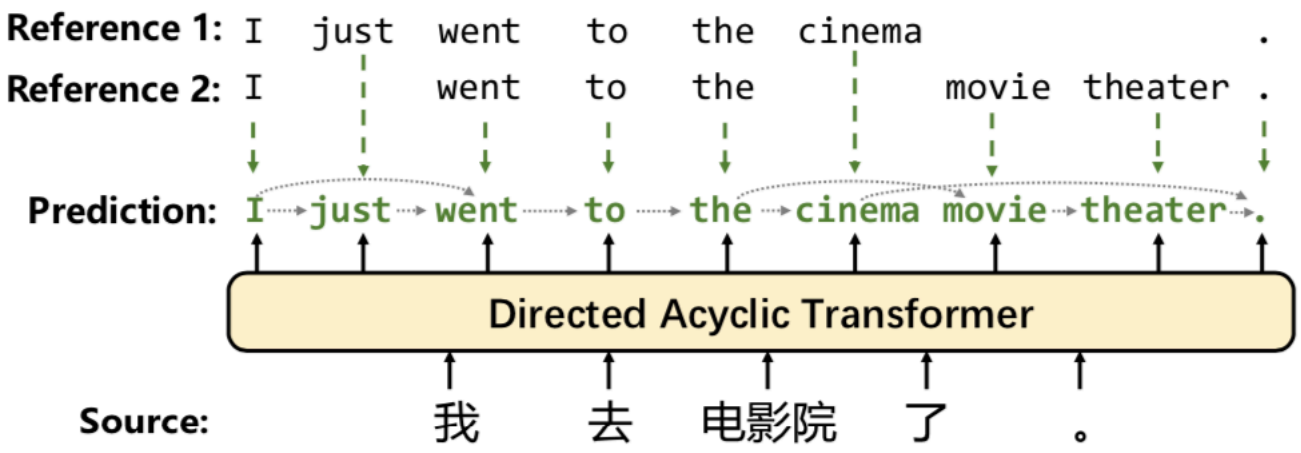


Gu J, Kong X. Fully non-autoregressive neural machine translation: Tricks of the trade. Findings of ACL 2021.

Libovický J, Helcl J. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. EMNLP 2018.

DA-Transformer

- Working mechanism:
  - Upsample the encoder by λ times as decoder inputs
  - Introduce translation path to capture different translation modalities simultaneously



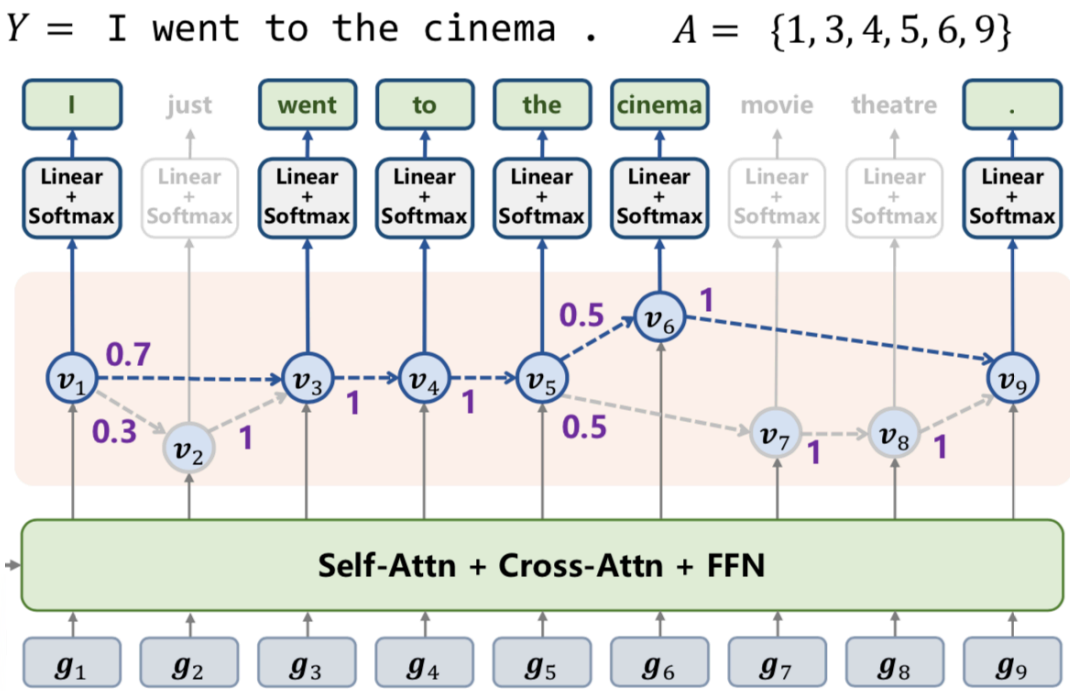
Huang F, Zhou H, Liu Y, et al. Directed Acyclic Transformer for Non-Autoregressive Machine Translation. ICML 2022.

DA-Transformer

- Introduce translation path A with Markov assumption

$$P_\theta(A|X) = \prod_{i=1}^{M-1} P_\theta(a_{i+1}|a_i, X)$$

$$P_\theta(Y|X) = \sum_{A\in\Gamma} P_\theta(A|X) P_\theta(Y|X, A)$$



- Training:
  - dynamic programming over all translation paths
- Decoding:
  - Lookahead decoding, Viterbi decoding, Beam search decoding

Huang F, Zhou H, Liu Y, et al. Directed Acyclic Transformer for Non-Autoregressive Machine Translation. ICML 2022.

Performance

- Performance for CTC and DA-Transformer

Model	Vanilla	CTC	DAT
Output space	VT	$VL + L$	$VL + L^2$
BLEU (raw)	11.4	18.4	26.5
BLEU (kd)	19.5	25.5	27.5

- State-of-the-art NAT models
  - With knowledge distillation, based on CTC
  - Without knowledge distillation, based on DA-Transformer

Overview

- Multi-modality problem
- Solutions:

Methods	Descriptions
Knowledge distillation	Reduce multi-modality of training data
Latent modeling	Reduce multi-modality with latent information
Expressive ability	More expressive NAR models
Training objectives	Replace MLE with cross-word objectives
Decoding approaches	Decoding approaches besides fully NAR decoding

Training Objectives

- MLE loss:
  - word-by-word matching
  - improper evaluation

Reference:	I	ate	pizza	this	afternoon
NAT output:	this	afternoon	I	ate	pizza

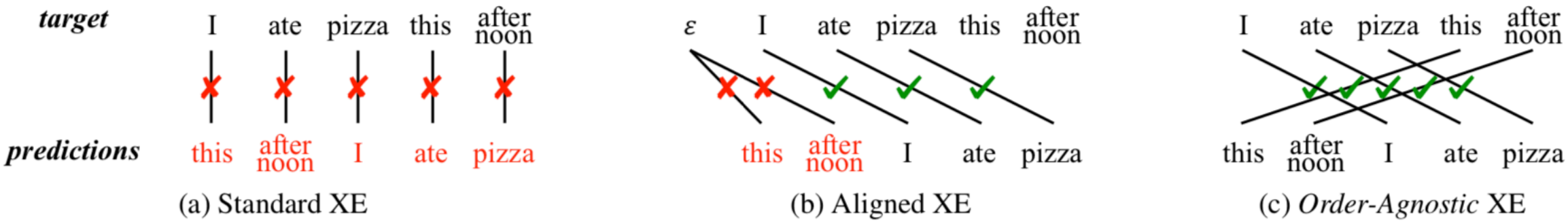
Training Objectives

- Standard of good training objectives for NAR model
  - Easy to optimize
  - Feasible to evaluate

- Training objectives:
  - Token-level
  - Ngram-level
  - Sequence-level

Token-Level Training

- Relax the alignment restriction of MLE loss
  - Aligned cross-entropy (AXE)
    - monotonic alignment with dynamic programming
  - Order-agnostic cross-entropy (OAXE)
    - non-monotonic alignment with Hungarian algorithm



Ghazvininejad M, Karpukhin V, Zettlemoyer L, et al. Aligned cross entropy for non-autoregressive machine translation. ICML 2020.
Du C, Tu Z, Jiang J. Order-agnostic cross entropy for non-autoregressive machine translation. ICML 2021.

Ngram-Level Training

- Bag-of-Ngrams
  - Encourage to generate the same n-grams as the reference
  - Minimize the bag-of-Ngram distance between model output and reference.

$$\mathrm{BoN}_\theta(\boldsymbol{g}) = \sum_{\boldsymbol{Y}} P(\boldsymbol{Y}|\boldsymbol{X},\theta) \cdot \mathrm{BoN}_{\boldsymbol{Y}}(\boldsymbol{g})$$

Shao C, Zhang J, Feng Y, et al. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation. AAAI 2020.

Ngram-Level Training

- Ngram-OAXE
  - Ngram-level MLE
  - Search with the Hungarian algorithm

Vocabulary	Output Probability Distribution				
	Pos:1	Pos:2	Pos:3	Pos:4	Pos:5
I	0.2	0.1	0.4	0.1	0.1
ate	0.1	0.1	0.1	0.4	0.1
pizza	0.1	0.1	0.3	0.1	0.5
this	0.5	0.1	0.1	0.3	0.1
afternoon	0.1	0.6	0.1	0.1	0.3

Word Distribution

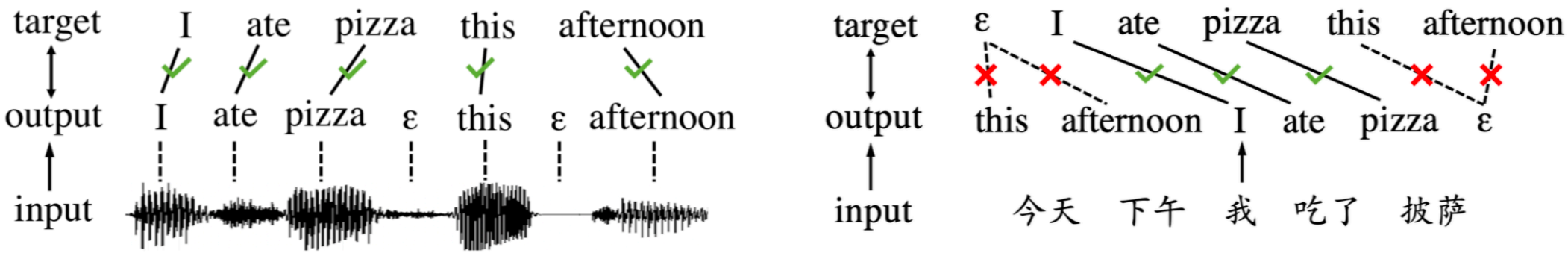
Bigram List	Output Probability Distribution			
	Pos:1,2	Pos:2,3	Pos:3,4	Pos:4,5
I ate	0.02	0.01	**0.16**	0.01
ate pizza	0.01	0.03	0.01	**0.20**
pizza this	0.01	**0.01**	0.09	0.01
this afternoon	**0.30**	0.01	0.01	0.09

Bigram Distribution

Du C, Tu Z, Wang L, et al. ngram-OAXE: Phrase-Based Order-Agnostic Cross Entropy for Non-Autoregressive Machine Translation. COLING 2022.

Ngram-Level Training

- Non-monotonic alignments for CTC
  - Bipartite matching,
  - N-gram matching



(a) CTC for ASR (b) CTC for NAT

Shao C, Feng Y. Non-Monotonic Latent Alignments for CTC-Based Non-Autoregressive Machine Translation. NeurIPS 2022.

Ngram-Level Training

- Fuzzy alignments for DA-Transformer
  - Ngram-level fuzzy alignments
  - Expected Ngram counts
  - The n-step transition probability of a Markov chain

CTC $$C_g^s(\theta) = \sum_{i_1=1}^{T_a} \sum_{i_2=1}^{T_a} \dots \sum_{i_n=1}^{T_a} p_{i_1}(g_1|x,\theta) \times \prod_{k=2}^{n} A_{i_{k-1},i_k} p_{i_k}(g_k|x,\theta)$$

$O(nT^n) \rightarrow O(nT^2)$

DAT $$\mathbb{E}_{\boldsymbol{y}'}[C_{\boldsymbol{g}}(\boldsymbol{y}')] = \sum_{v_1,\dots,v_n} P(\mathbb{1}(v_1)|\boldsymbol{x}) \prod_{i=1}^{n-1} \mathbf{E}_{v_i,v_{i+1}} \prod_{i=1}^{n} P_\theta(g_i|v_i)$$

Shao C, Feng Y. Non-Monotonic Latent Alignments for CTC-Based Non-Autoregressive Machine Translation. NeurIPS 2022.
Anonymous. Fuzzy Alignments in Directed Acyclic Graph for Non-Autoregressive Machine Translation. Under review at ICLR 2023.

Sequence-Level Training

- Sequence-level objectives with Reinforcement learning
  - Expected BLEU as reward

$$\mathrm{L}_\theta = -\sum_{\mathbf{Y}=y_{1:T}} p(\mathbf{Y}|\mathbf{X},\theta)\cdot r(\mathbf{Y})$$

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

- Variance reduction: Reinforce-Step, Reinforce-Topk, Traverse-Ref

Shao C, Feng Y, Zhang J, et al. Retrieving sequential information for non-autoregressive neural machine translation. ACL 2019.
Shao C, Feng Y, Zhang J, et al. Sequence-level training for non-autoregressive neural machine translation. Computational Linguistics 2021.

Sequence-Level Training

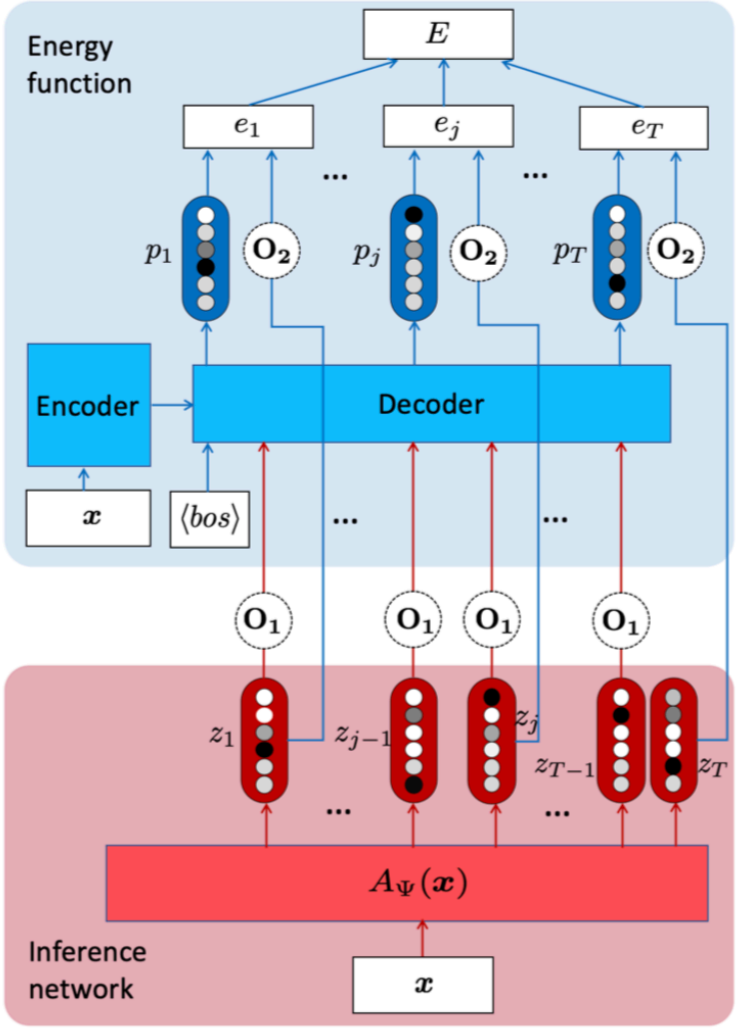
- Energy-Based Inference Networks
  - optimizing the sequence-level energy

- Autoregressive teacher energy

$$E(x, y) = -\log p_{ar}(y|x)$$

- Optimal distribution is also one-hot

$$y^* = \underset{y}{\operatorname{argmax}}\, p_{ar}(y|x)$$



Tu L, Pang R Y, Wiseman S, et al. ENGINE: Energy-based inference networks for non-autoregressive machine translation. EMNLP 2020

Overview

- Multi-modality problem
- Solutions:

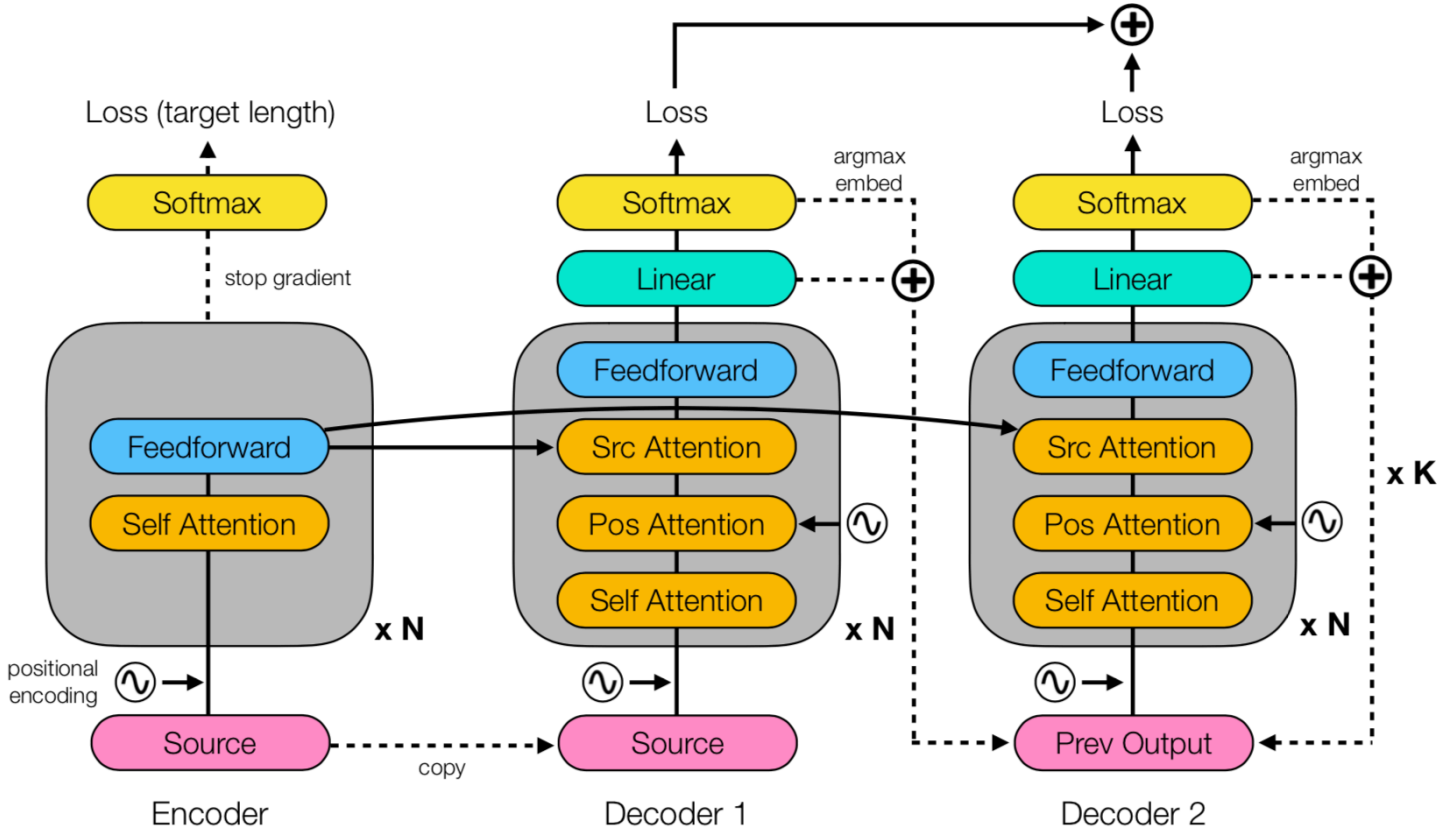
Methods	Descriptions
Knowledge distillation	Reduce multi-modality of training data
Latent modeling	Reduce multi-modality with latent information
Expressive ability	More expressive NAR models
Training objectives	Replace MLE with cross-word objectives
Decoding approaches	Decoding approaches besides fully NAR decoding

Decoding Approaches

- AR decoding
  - slow but accurate
- NAR decoding
  - fast but the naive version has low quality
- Alternative decoding approaches
  - Iterative decoding: iteratively refine model outputs
  - Semi-autoregressive decoding: generate multiple tokens at a time
  - Blockwise decoding: predict by NAR and verify by AR
  - NAR generation + AR rescoring

Iterative Decoding

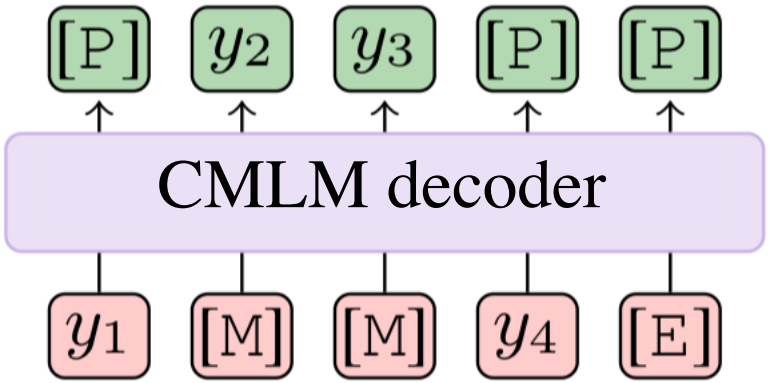
- Iteration t decoder output → iteration t+1 decoder input



Lee J, Mansimov E, Cho K. Deterministic non-autoregressive neural sequence modeling by iterative refinement. EMNLP 2018.

Mask-Predict

- Conditional masked language model (CMLM)
  - Training:
    - randomly mask 0~100% tokens
  - Decoding:
    - mask tokens with lowest probability scores

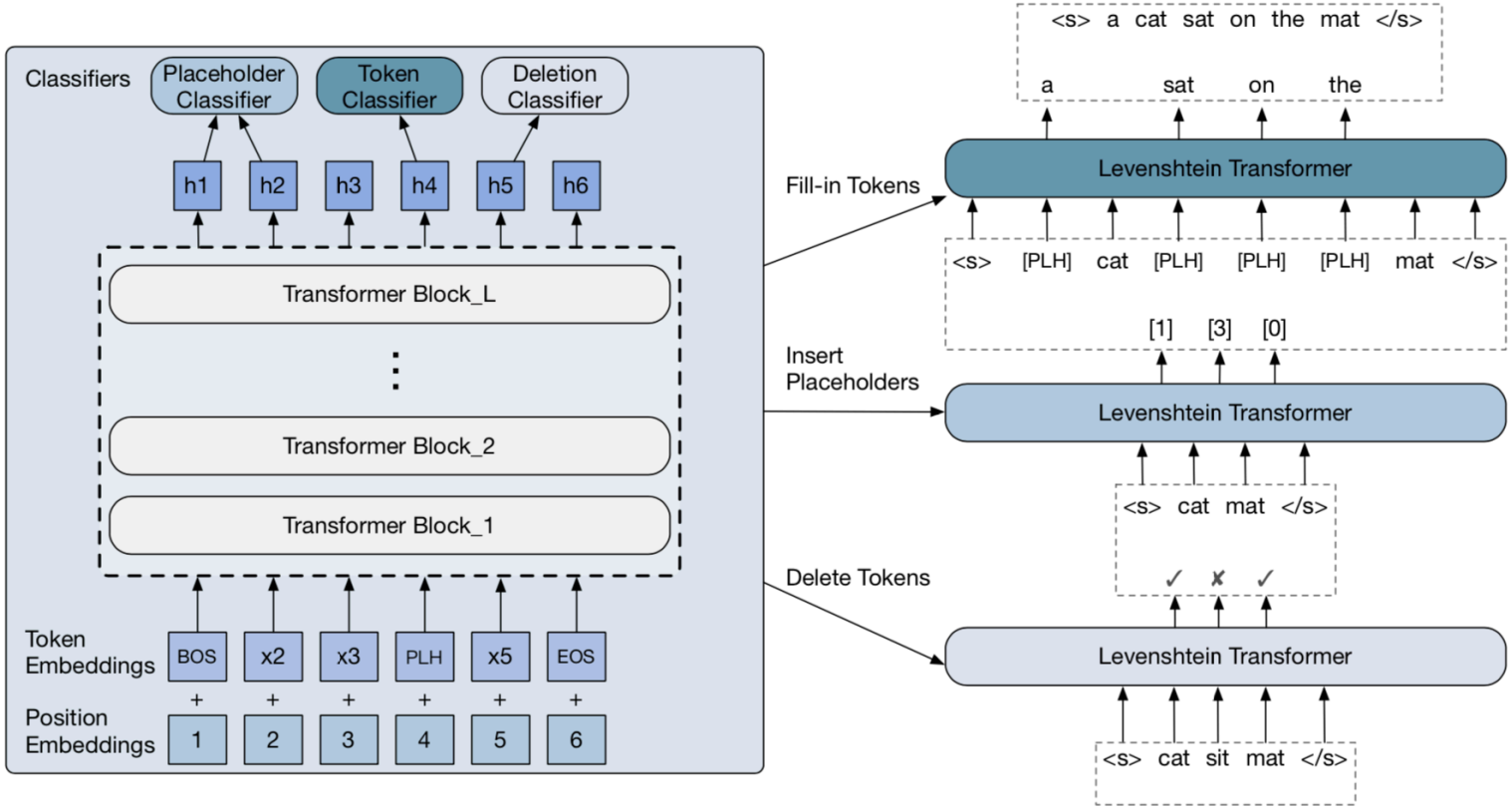


src	Der Abzug der franzsischen Kampftruppen wurde am 20. November abgeschlossen .
$t = 0$	The departure of the French combat completed completed on 20 November .
$t = 1$	The departure of French combat troops was completed on 20 November .
$t = 2$	The withdrawal of French combat troops was completed on November 20th .

Ghazvininejad M, Levy O, Liu Y, et al. Mask-predict: Parallel decoding of conditional masked language models. EMNLP 2019.

Levenshtein Transformer

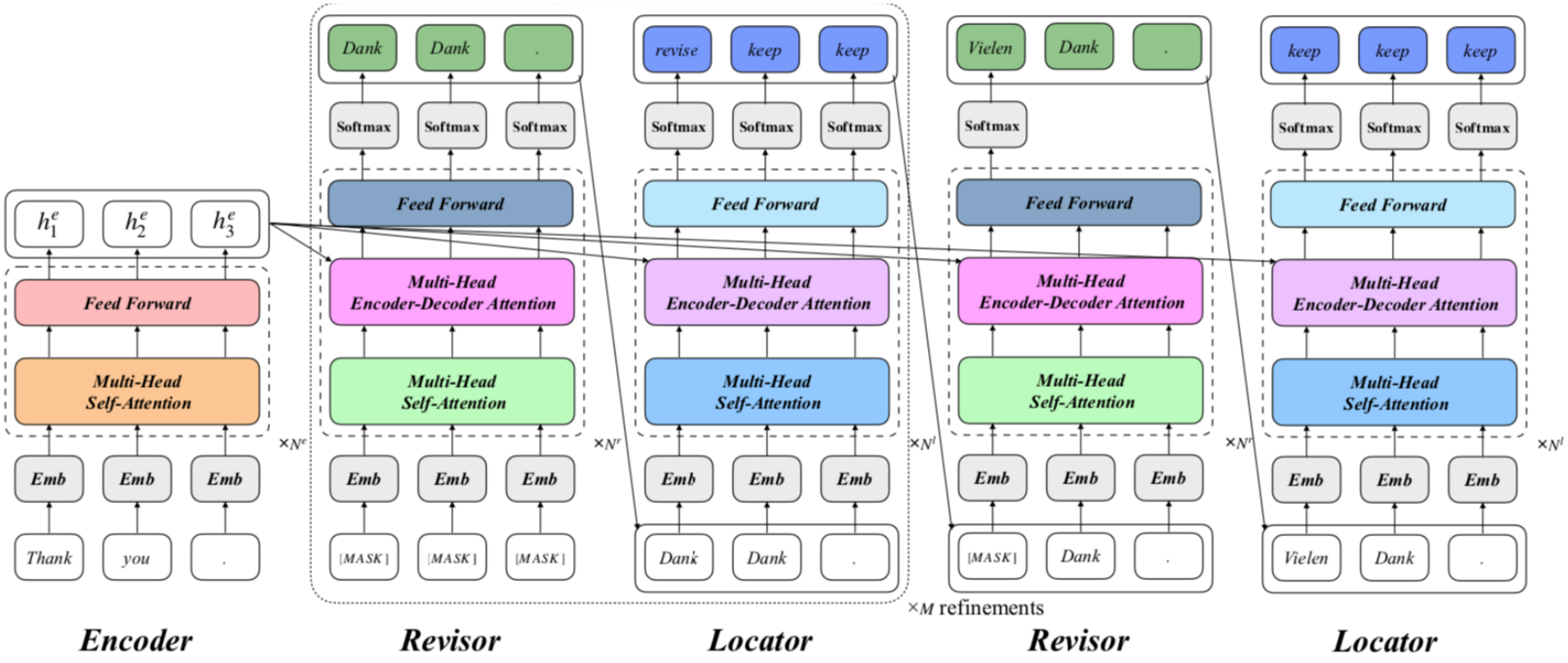
- Actions to take:
  - delete tokens
  - insert placeholders
  - predict tokens



Gu J, Wang C, Zhao J. Levenshtein transformer. NIPS 2019.

RewriteNAT

- RewriteNAT
  - Introduce Locator module and Revisor module
  - locate errors and revise



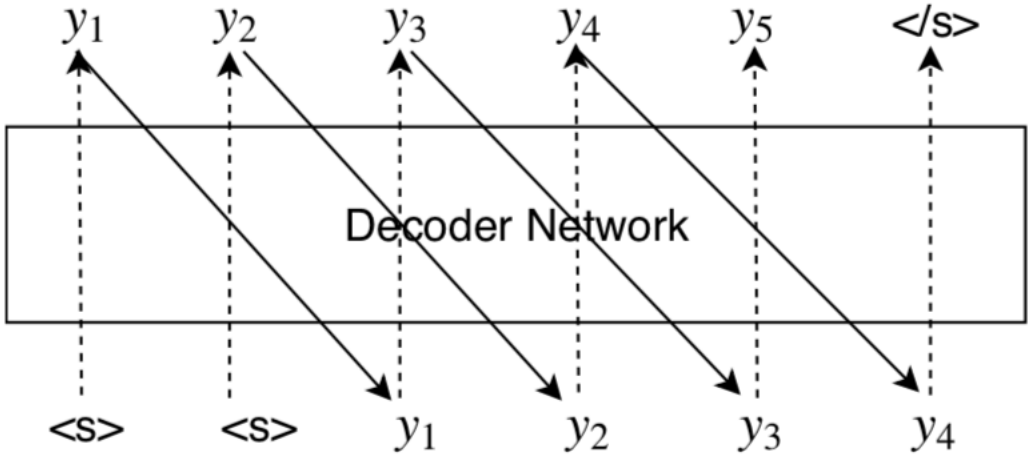
Geng X, Feng X, Qin B. Learning to rewrite for non-autoregressive neural machine translation. EMNLP 2021.

Semi-Autoregressive

- Semi-autoregressive decoding
  - Generate K tokens at a time
  - Group-level chain rule, each group contains K tokens

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

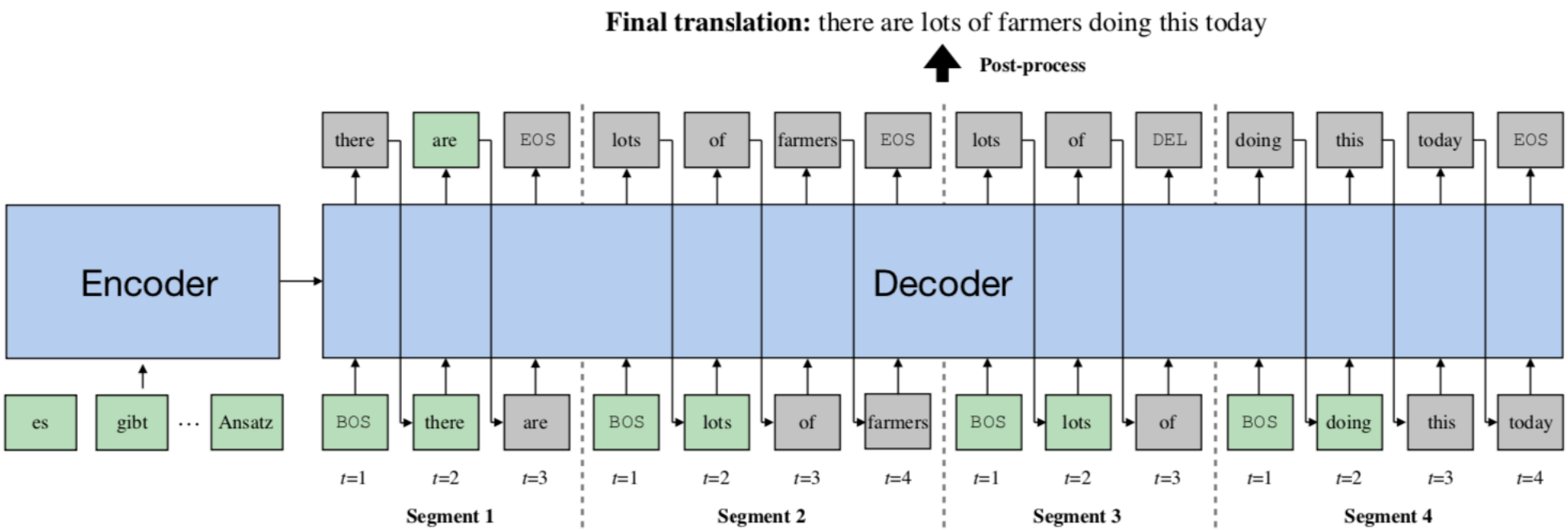
- Speedup: at most K times to AR decoding



Wang C, Zhang J, Chen H. Semi-autoregressive neural machine translation. EMNLP 2018.

Multi-Thread

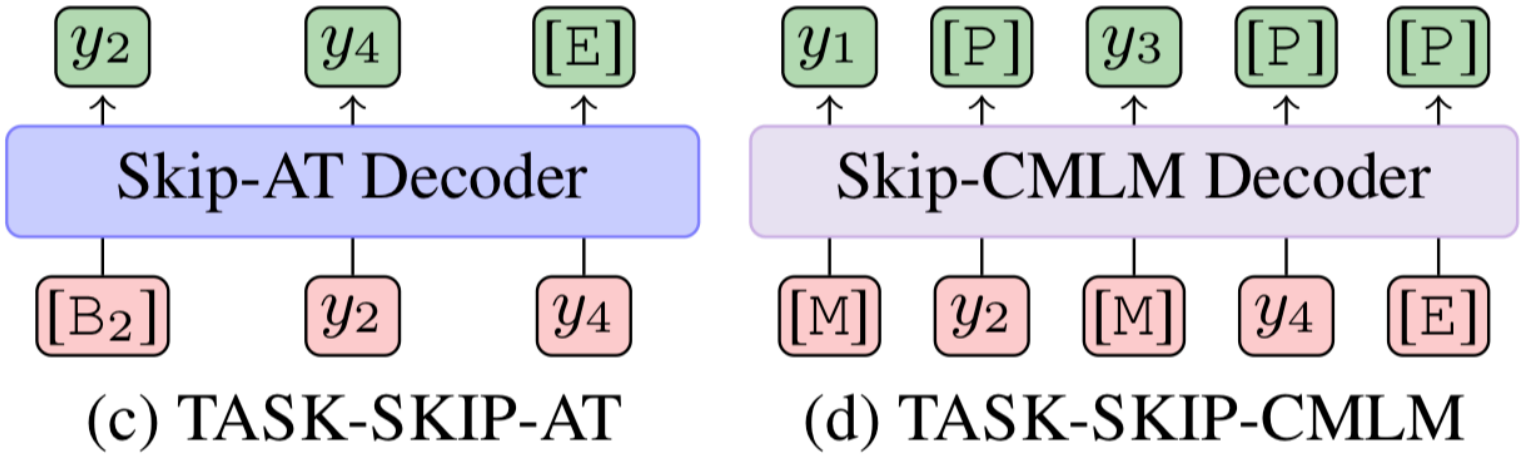
- Multi-thread decoding
  - generate K tokens at a time
  - Semi-autoregressive: $y_1, y_2, y_3 \rightarrow y_4, y_5, y_6 \rightarrow y_7, y_8, y_9$
  - Multi-thread: $y_1, y_4, y_7 \rightarrow y_2, y_5, y_8 \rightarrow y_3, y_6, y_9$



Ran Q, Lin Y, Li P, et al. Learning to recover from multi-modality errors for non-autoregressive neural machine translation. ACL 2020.

Hybrid-Regressive

- Hybrid-regressive: combine AR and NAR models
- Skip-autoregressive: each group contains K tokens, generate the first token in each group by an AR model
- Skip-CMLM: fill in the blanks with a NAR model

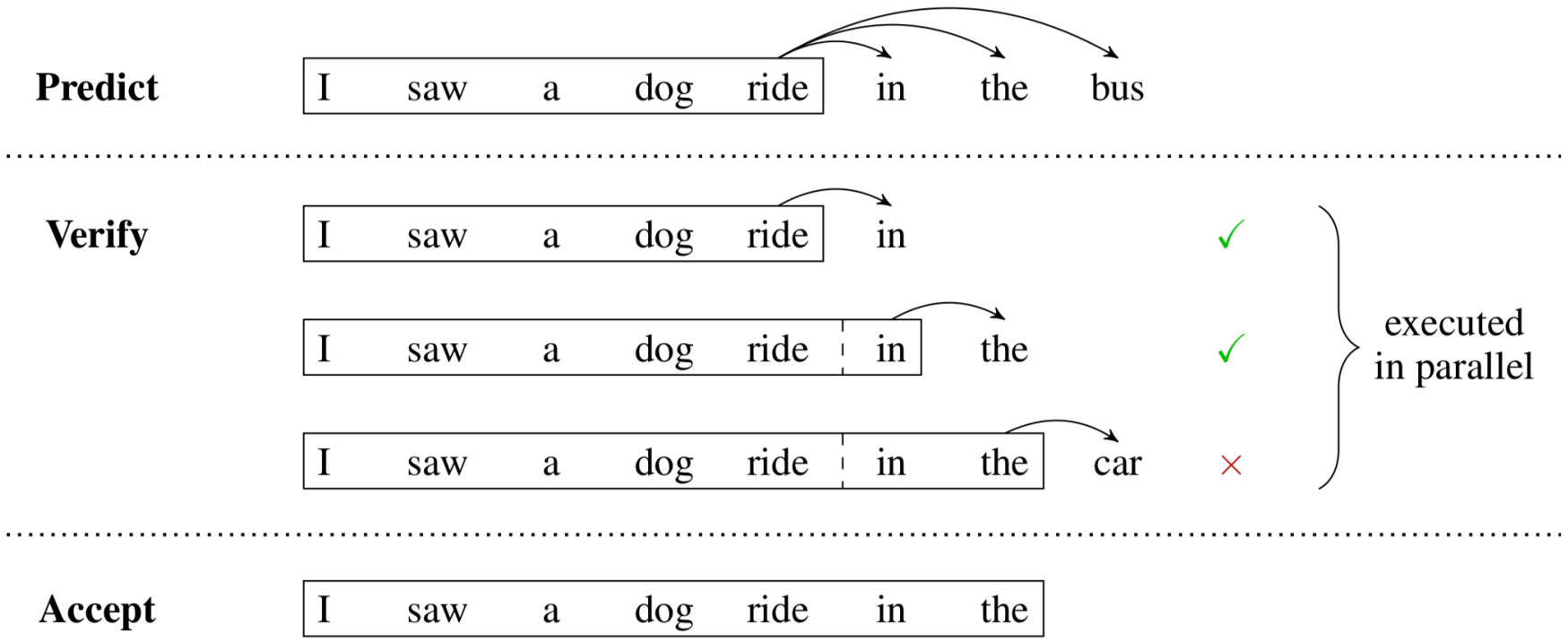


(c) TASK-SKIP-AT (d) TASK-SKIP-CMLM

Wang Q, Hu X, Chen M. Hybrid-regressive neural machine translation. arXiv preprint arXiv:2210.10416, 2022.

Blockwise decoding

- Combine NAR and AR models to generate the same result as AR greedy decoding, while being much faster
- Predict (NAR), verify (AR, parallelizable), accept



Stern M, Shazeer N, Uszkoreit J. Blockwise parallel decoding for deep autoregressive models. NIPS 2018.

NAR Generation + AR Rescoring

- NAR generation + AR rescoring
- Both NAR generation and AR rescoring are parallelizable

- How to generate multiple candidates: sample latent variables
  - Predicted length L, generate 2B+1 candidates from length beam [L-B, L+B]
  - Other latent modeling methods, sample from prior distribution

$$z_i \sim p_{NAR}(z|x), y_i = \text{argmax}_y \, p_{NAR}(y|x,z)$$
$$\text{y} = \text{argmax}_i \, p_{AR}(y_i|x)$$

Gu J, Bradbury J, Xiong C, et al. Non-autoregressive neural machine translation. ICLR 2018.

Beam Search with LM

- NAR models with upsampling: CTC and DA-Transformer
- Support beam search, optionally combined with a n-gram LM

CTC $\log p_\theta(\boldsymbol{y}|\boldsymbol{x}) + \alpha \cdot \log p_{\text{LM}}(\boldsymbol{y}) + \beta \log |\boldsymbol{y}|$

DA-Transformer $\frac{1}{|Y|^\alpha}\left[\log P_\theta(Y|X) + \gamma \log P_{n\text{-gram}}(Y)\right]$

- Without neural computation, implemented in C++

Gu J, Kong X. Fully non-autoregressive neural machine translation: Tricks of the trade. Findings of ACL 2021.
Huang F, Zhou H, Liu Y, et al. Directed Acyclic Transformer for Non-Autoregressive Machine Translation. ICML 2022.

Combination

- Build state-of-the-art NAT system by combining methods from different perspectives

Methods	Descriptions
Knowledge distillation	Reduce multi-modality of training data
Latent modeling	Reduce multi-modality with latent information
Expressive ability	More expressive NAR models
Training objectives	Replace MLE with cross-word objectives
Decoding approaches	Decoding approaches besides fully NAR decoding

Combination

- Combination: Fully-NAT
  - On WMT14 En-De
  - AR Transformer: 27.48 BLEU
  - Fully-NAT: 27.49 BLEU, 16.5× speedup

Methods	Descriptions
Knowledge distillation	Standard KD
Latent modeling	GLAT or VAE
Expressive ability	CTC
Training objectives	None
Decoding approaches	None

Models		Iter.	Speed	WMT'14 EN-DE	WMT'14 DE-EN	WMT'16 EN-RO	WMT'16 RO-EN
AT	Transformer *base* (teacher)	N	1.0×	**27.48**	**31.39**	**33.70**	**34.05**
	Transformer *base* (12-1)	N	2.4×	26.21	30.80	33.17	33.21
	+ KD	N	2.5×	27.34	30.95	33.52	34.01
Fully NAT	***Ours (Fully NAT)***	1	17.6×	11.40	16.47	24.52	24.79
	+ KD	1	17.6×	19.50	24.95	29.91	30.25
	+ KD + CTC	1	16.8×	26.51	30.46	33.41	34.07
	+ KD + CTC + VAE	1	16.5×	**27.49**	31.10	**33.79**	33.87
	+ KD + CTC + GLAT	1	16.8×	27.20	**31.39**	33.71	**34.16**

Gu J, Kong X. Fully non-autoregressive neural machine translation: Tricks of the trade. Findings of ACL 2021.

Combination

- Combination: NMLA
  - Non-Monotonic Latent Alignments
  - Current SOTA NAT system
  - WMT14 En-De, +1 BLEU to AR
  - WMT16 Ro-En, +2 BLEU to AR

Methods	Descriptions
Knowledge distillation	Diverse distillation
Latent modeling	None
Expressive ability	CTC
Training objectives	N-gram level
Decoding approaches	N-gram LM

Models		Iter.	Speed	WMT14		WMT16	
				EN-DE	DE-EN	EN-RO	RO-EN
AT	Transformer (teacher)	N	1.0×	27.54	**31.57**	**34.26**	**33.87**
	Transformer (12-1)	N	2.6×	26.09	30.30	32.76	32.39
	+ KD	N	2.7×	**27.61**	31.48	33.43	33.50
Our work	CTC w/o finetune	1	14.7×	26.34	29.58	33.45	33.32
	CTC w/ NMLA	1	14.7×	27.57	31.28	33.86	33.94
	+ beam&lm	1	5.0×	**28.35**	**32.27**	**34.72**	**34.95**
	DDRS w/o finetune	1	14.7×	27.18	30.91	34.42	34.31
	DDRS w/ NMLA	1	14.7×	28.02	31.80	34.73	34.76
	+ beam&lm	1	5.0×	**28.63**	**32.65**	**35.51**	**35.85**

Shao C, Feng Y. Non-Monotonic Latent Alignments for CTC-Based Non-Autoregressive Machine Translation. NeurIPS 2022.

Combination

- Combination: FA-DAT
  - Fuzzy Alignments for DA-Transformer
  - Current SOTA NAT system without knowledge distillation

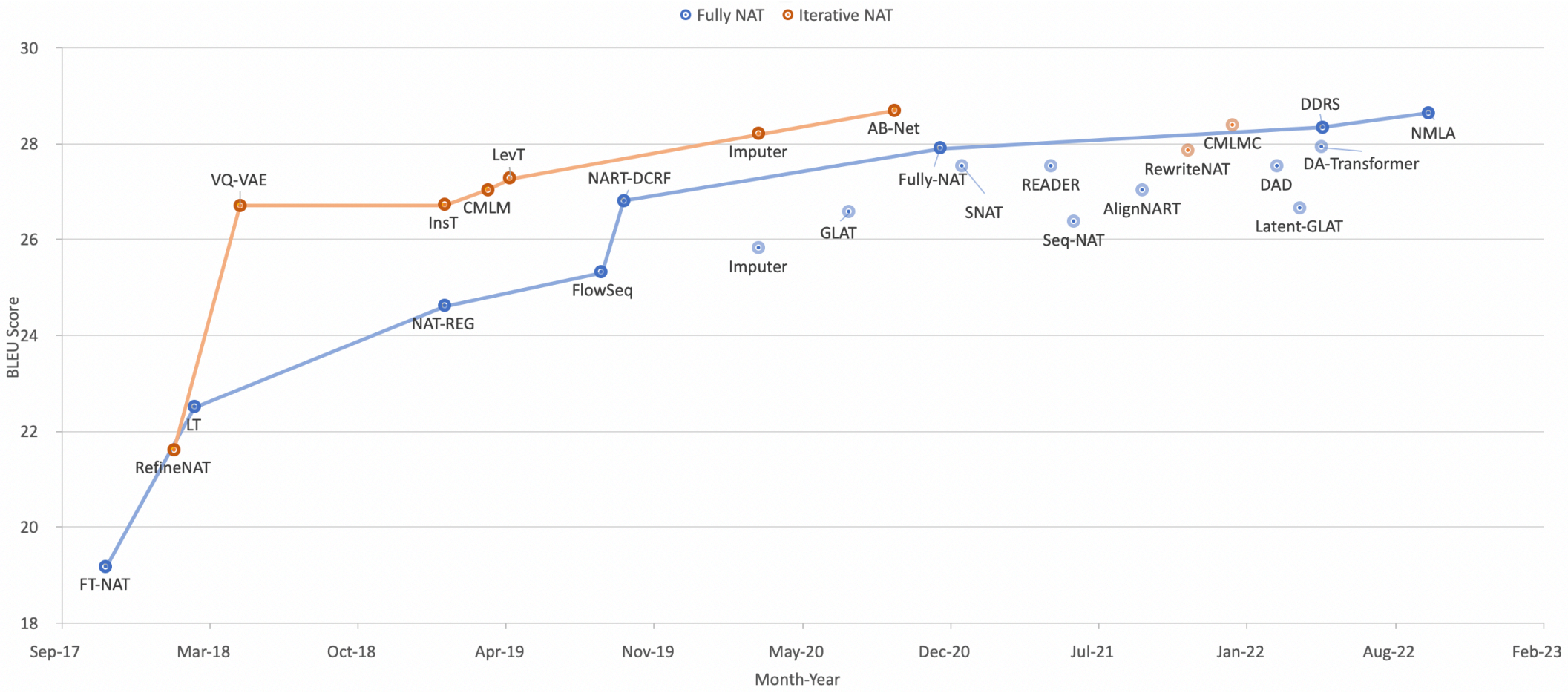
Methods	Descriptions
Knowledge distillation	None
Latent modeling	GLAT
Expressive ability	DA-Transformer
Training objectives	N-gram level
Decoding approaches	Lookahead

Model		Iter.	Speedup	WMT14 EN-DE	WMT14 DE-EN	WMT17 ZH-EN	WMT17 EN-ZH
Transformer(Vaswani et al., 2017)		N	1.0×	27.54	31.55	24.23	35.19
DA-Transformer (Huang et al., 2022b)	+ *Greedy*	1	14.2×	26.08	30.48	22.66	33.27
	+ *Lookahead*	1	14.0×	26.57	30.68	22.82	33.83
DA-Transformer (Our implementation)	+ *Greedy*	1	14.2×	26.07	30.69	22.35	33.58
	+ *Lookahead*	1	14.0×	26.56	30.81	22.65	33.62
FA-DAT	+ *Greedy*	1	14.2×	27.49**	31.36**	23.78**	33.97*
	+ *Lookahead*	1	14.0×	**27.53****	**31.37****	**23.81****	**34.02****

Anonymous. Fuzzy Alignments in Directed Acyclic Graph for Non-Autoregressive Machine Translation. Under review at ICLR 2023.

Summary

- Evolution of NAT



Summary

- NAR translation models: multi-modality
- Combining methods from different perspectives to build state-of-the-art NAT

Methods	Descriptions
Knowledge distillation	Reduce multi-modality of training data
Latent modeling	Reduce multi-modality with latent information
Expressive ability	More expressive NAR models
Training objectives	Replace MLE with cross-word objectives
Decoding approaches	Decoding approaches besides fully NAR decoding

Outline

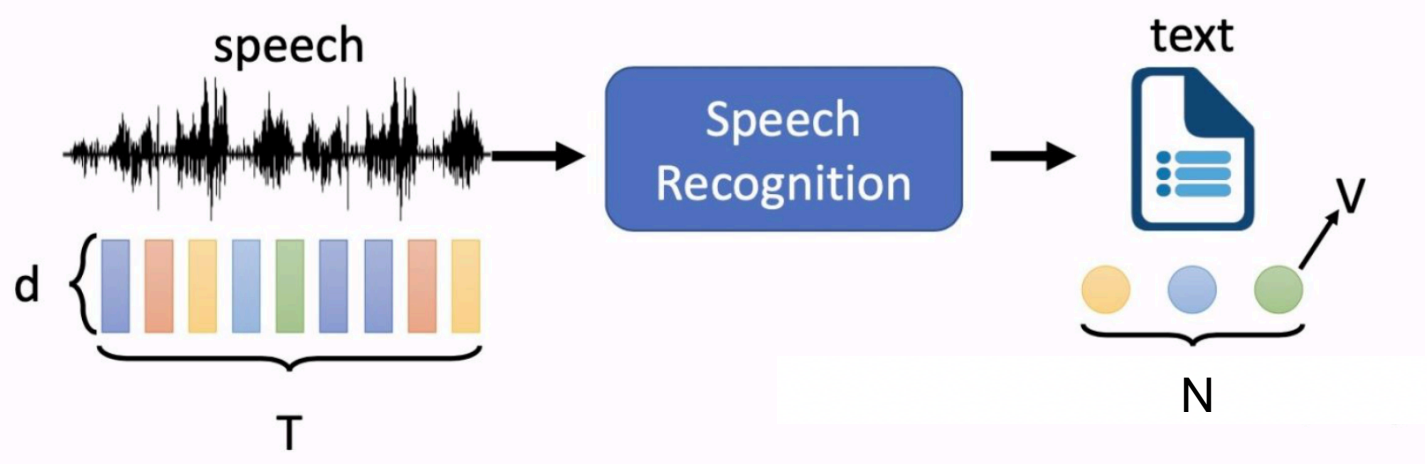
- Introduction (Yang Feng)
- Non-Autoregressive Translation (Yang Feng)
- Non-Autoregressive Generation (Chenze Shao)
- Conclusion and Future Direction (Chenze Shao)

Overview

- NAR generation is very popular on speech-related generation tasks
  - automatic speech recognition (ASR), text to speech (TTS)
- Are some tasks more difficult for NAR generation than others?
  - multi-modality, target-token dependency
- Is unconditional NAR generation possible?
  - language modeling, extremely high multi-modality
- More applications of NAR generation
  - dialogue generation, image captioning, object detection

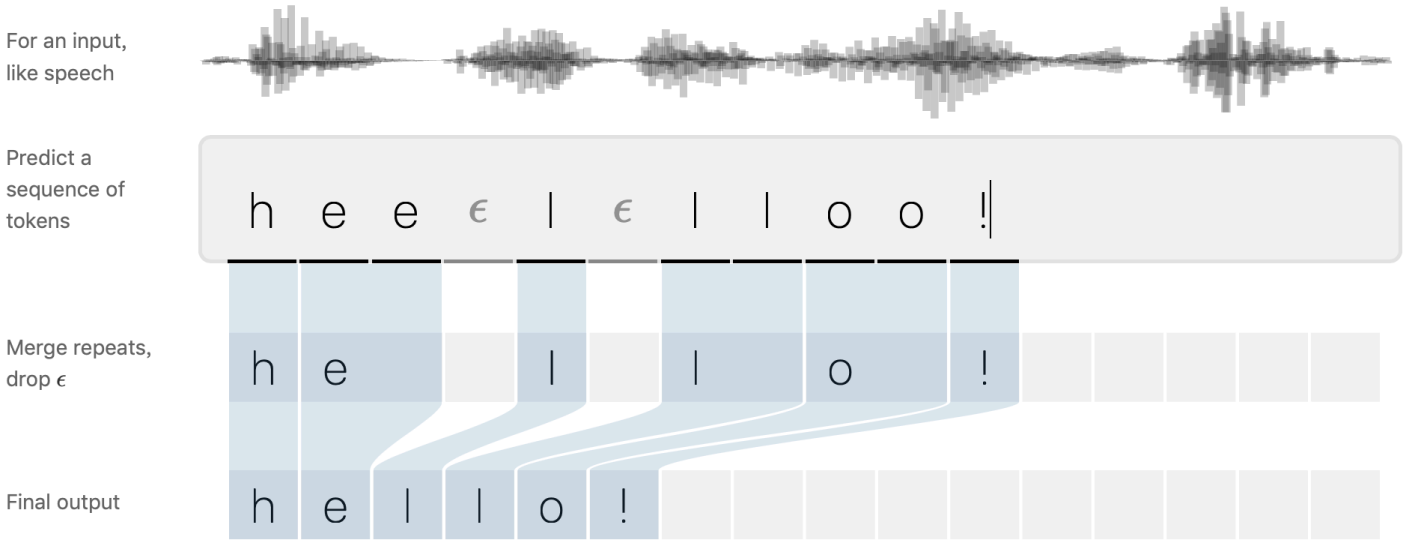
Automatic Speech Recognition

- Automatic speech recognition: speech to text
  - Input: speech (usually spectrogram); Output: text (usually character-level)
  - Speech input is much longer than text output, text output is much longer than machine translation output (subword-level)
  - NAR generation can bring more significant decoding speedup in ASR
- Specific property: monotonic alignment between speech and text



CTC

- In 2006, Connectionist Temporal Classification (CTC)
- Early exploration of NAR Generation, unparallelizable at that time (RNN)
- Properties of ASR: input is longer than output, monotonic alignment



Graves A, Fernández S, Gomez F, et al. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. ICML 2006.

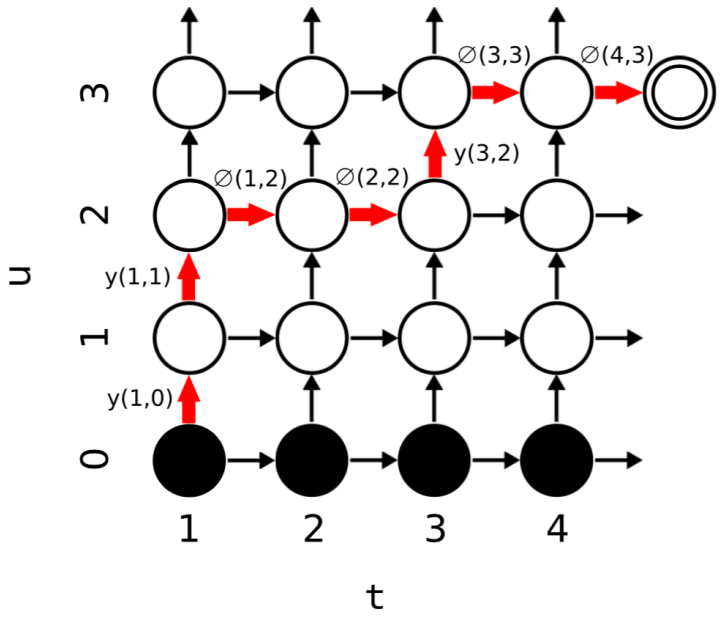
Transducer

- In 2012, Transducer, combines AR and NAR generation
  - NAR model f, bidirectional RNN, scans input sequence $(x_1, \dots, x_T)$
  - AR model g, unidirectional RNN, scans output sequence $(\phi, y_1, \dots, y_U)$
- Probability of output $y_{u+1} = k$ at position t

$$h(k,t,u) = \exp\left(f_t^k + g_u^k\right)$$

$$\Pr(k \in \bar{\mathcal{Y}} | t, u) = \frac{h(k,t,u)}{\sum_{k' \in \bar{\mathcal{Y}}} h(k',t,u)}$$

- Training: dynamic programming
- Decoding: beam search



Graves A. Sequence transduction with recurrent neural networks. arXiv preprint arXiv:1211.3711, 2012.

Training of CTC

- In 2014, train CTC with sequence-level objectives (WER)
- WER: word error rate defined as edit distance
- Optimizing the expected value of WER

$$\mathcal{L}(\boldsymbol{x}) = \sum_{\boldsymbol{y}} \sum_{\boldsymbol{a} \in \mathcal{B}^{-1}(\boldsymbol{y})} \Pr(\boldsymbol{a}|\boldsymbol{x}) \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) \longrightarrow \text{WER}$$

$$= \sum_{\boldsymbol{a}} \Pr(\boldsymbol{a}|\boldsymbol{x}) \mathcal{L}(\boldsymbol{x}, \mathcal{B}(\boldsymbol{a}))$$

- Estimate the gradient on each step

$$\frac{\partial \mathcal{L}(\boldsymbol{x})}{\partial \Pr(k, t|\boldsymbol{x})} = \sum_{\boldsymbol{a}:\boldsymbol{a}_t = k} \Pr(\boldsymbol{a}|\boldsymbol{x}, \boldsymbol{a}_t = k) \mathcal{L}(\boldsymbol{x}, \mathcal{B}(\boldsymbol{a}))$$

expected WER when fix $a_t = k$

Graves A, Jaitly N. Towards end-to-end speech recognition with recurrent neural networks. ICML 2014.

A-CMLM

- CTC-based models are non-autoregressive, but their decoding are still unparallelizable in pre-Transformer era
- Parallel generation of ASR with non-autoregressive Transformer
- Audio-Conditional Masked Language Model (A-CMLM)
- T_M: masked tokens, T_U: unmasked tokens

$$P(\mathbf{y}_{T_\mathrm{M}}|\mathbf{y}_{T_\mathrm{U}}, \mathbf{x}) = \prod_{t \in T_\mathrm{M}} P_{\mathsf{dec}}(\mathbf{y}_t|\mathbf{y}_{T_\mathrm{U}}, f_t(\mathbf{h}))$$

- Decoding: mask tokens with lowest probability scores in each iteration

Chen N, Watanabe S, Villalba J, et al. Non-autoregressive transformer for speech recognition. IEEE Signal Processing Letters 2020.

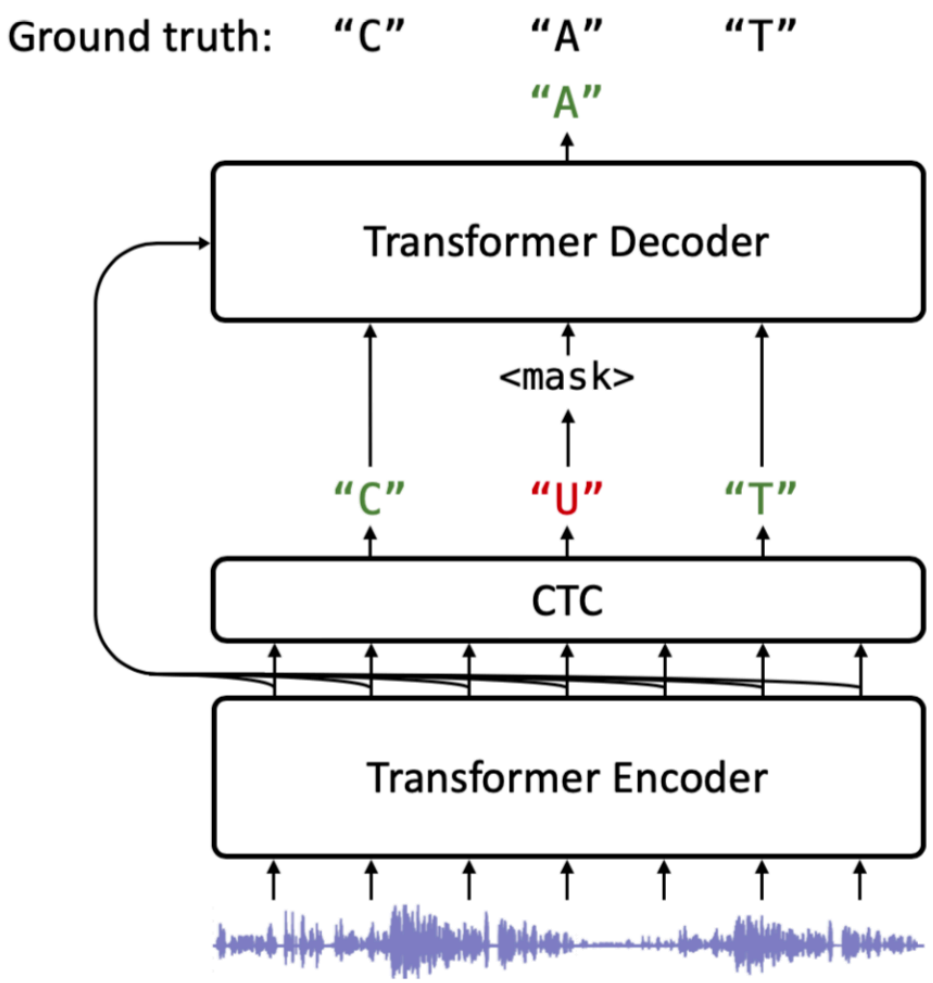
Mask CTC

- Mask CTC: combine CTC and mask-predict, joint training

$$\mathcal{L}_{\mathrm{NAR}} = \gamma \log P_{\mathrm{ctc}}(Y|X) + (1-\gamma) \log P_{\mathrm{cmlm}}(Y_{\mathrm{mask}}|Y_{\mathrm{obs}}, X)$$

- CTC: generate the outputs as the initial output sequence

- Mask-predict: iteratively refine the CTC outputs using the CMLM decoder



Higuchi Y, Watanabe S, Chen N, et al. Mask CTC: Non-autoregressive end-to-end ASR with CTC and mask predict. INTERSPEECH 2020.

Imputer

- Imputer: directly perform mask-predict on the CTC sequence
- Difference between Mask CTC and Imputer
  - Mask CTC, merge the CTC output, token-level: $(A, B, \phi, D) \rightarrow (A, B, C, D)$
  - Imputer, without merging, frame-level: $(A, -, B, \phi, \phi, \phi, D) \rightarrow (A, -, B, C, -, -, D)$
- Frame-level is more flexible but has higher computational cost

1. Sample an alignment $a \sim q_{\phi'}$ from our alignment policy q'_{ϕ}.

$$a = (A, _, B, _, _, C, D)$$

2. Sample a masked-out alignment $\tilde{a} \sim r(a)$ from our masking policy r. Imputer conditions on the masked-out alignment $\tilde{a}$.

$$\tilde{a} = (A, _, B, \varnothing, \varnothing, \varnothing, D)$$

3. Imputer marginalizes over all compatible alignments to the conditioning $\tilde{a}$. This can be solved exactly and efficiently via dynamic programming.

$$A \;\; - \;\; B \left\{ \begin{matrix} - & - & C \\ - & C & - \\ C & - & - \end{matrix} \right\} D$$

Imputer

$$A \;\; - \;\; B \;\; \varnothing \;\; \varnothing \;\; \varnothing \;\; D$$

Chan W, Saharia C, Hinton G, et al. Imputer: Sequence modelling via imputation and dynamic programming. ICML 2020.

Comparison

- Without CTC
  - 17, 20, 21
- CTC as regularization loss
  - 22, 39
- Single iteration CTC
  - 33, 35, 36
- CTC + token-level refine
  - 19, 31
- CTC + frame-level refine
  - 18, 23

Model	#iter	Processing unit	CTC
A-CMLM [17]	3	token	
Imputer [18]	8	frame	✓
LASO [20]	1	token	
Spike-Triggered [22]	1	token	✓
Mask-CTC [19]	10	token	✓
Improved Mask-CTC [31]	5	token	✓
Align-Refine [23]	5	frame	✓
Align-Denoise [33]	1	frame	✓
Insertion Transformer [21]	$\simeq \log_2(L)$	token	
KERMIT [21]	$\simeq \log_2(L)$	token	✓
Intermediate CTC [35]	1	frame	✓
Self-conditioned CTC [36]	1	frame	✓
CIF-NA [39]	1	token	✓

Higuchi Y, Chen N, Fujita Y, et al. A comparative study on non-autoregressive modelings for speech-to-text generation. IEEE ASRU Workshop 2021.

Comparison

- NAR models can achieve competitive results on clean data, but resulted in worse performances than AR model on noisy data

Model		#iter	Inference speed		LS-100 (WER)				TED2 (WER)		CSJ-APS (CER)		
			RTF	Speedup	dev clean	dev other	test clean	test other	dev	test	eval1	eval2	eval3
AR	CTC/attention	L	0.341	1.00×	6.8	18.8	7.4	19.0	11.6	8.7	5.4	4.0	9.8
	+ beam-search	$> L$	3.419	0.10×	**6.3**	**18.2**	**6.8**	**18.5**	10.4	8.4	**5.1**	**3.8**	**9.0**
	Transducer	L	0.069	4.94×	7.3	19.9	7.6	19.9	9.6	9.2	6.3	4.5	10.6
	+ beam-search	$> L$	0.234	1.46×	6.4	18.8	**6.8**	18.9	**8.6**	**8.2**	5.2	4.1	10.0
NAR	CTC	1	0.059	5.78×	7.4	20.5	7.8	20.8	8.9	8.6	5.4	4.0	9.6
	Mask-CTC	10	0.063	5.41×	7.2	20.3	7.5	20.6	8.9	8.5	5.6	4.0	9.6
	Improved Mask-CTC	5	0.072	4.74×	7.0	19.8	7.3	20.2	8.8	8.3	5.5	4.0	9.5
	Align-Denoise	1	0.073	4.67×	8.0	22.3	8.4	22.5	9.0	8.7	5.4	**3.7**	**9.1**
	Intermediate CTC	1	0.059	5.78×	6.9	19.7	7.1	20.2	**8.5**	8.3	5.6	4.1	9.8
	Self-conditioned CTC	1	0.059	5.78×	**6.6**	**19.4**	**6.9**	**19.7**	8.7	**8.0**	**5.3**	**3.7**	**9.1**
	KERMIT	$\simeq \log_2(L)$	0.361	1.06×	7.1	19.7	7.4	20.2	9.1	8.2	5.4	**3.7**	9.5
	Insertion Transformer	$\simeq \log_2(L)$	0.083	4.11×	16.0	27.3	16.2	27.4	–	–	–	–	–
	CIF-NA†	1	0.073	4.67×	15.4	34.0	15.7	34.6	–	–	–	–	–

Text to Speech

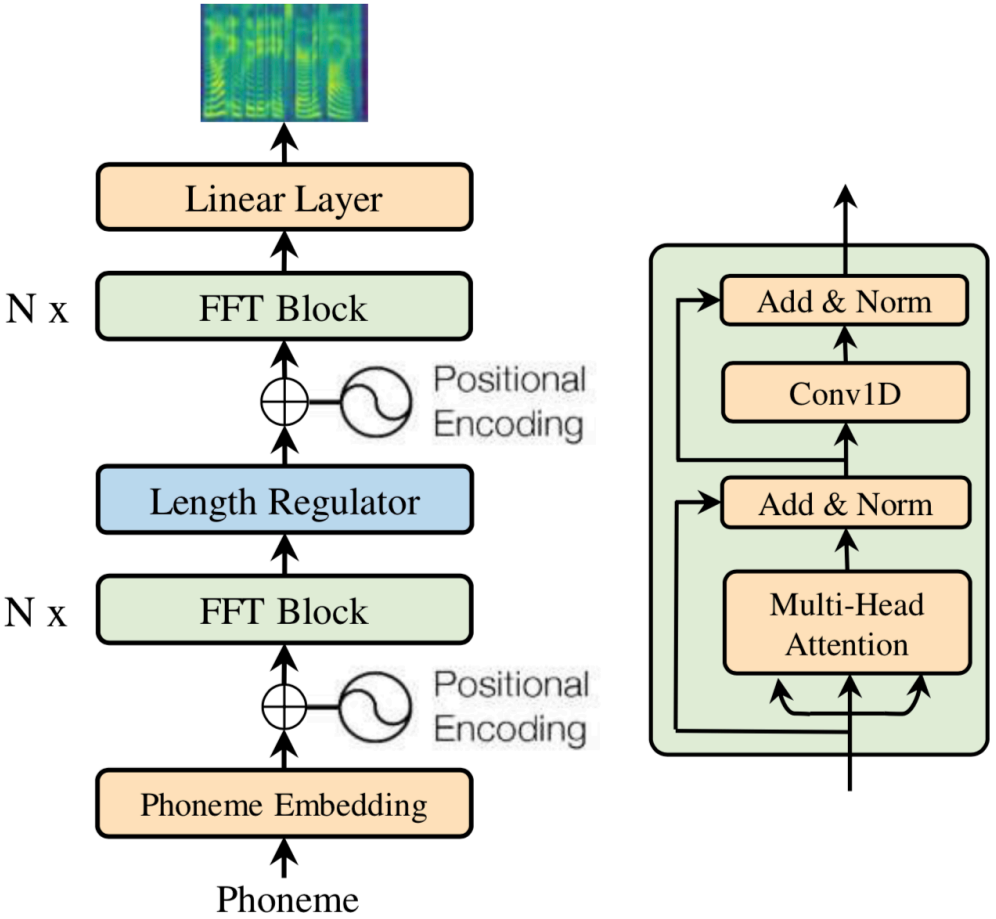
- Text to speech (TTS): synthesize intelligible and natural speech from text
- Text Analysis + Acoustic Model + Vocoder, or end-to-end

- The length of TTS output is very long, needs NAR generation
  - 500 mel-spectrogram for 1s speech, the waveform is even much longer
- We focus on the acoustic model here

Tan X, Qin T, Soong F, et al. A survey on neural speech synthesis. arXiv preprint arXiv:2106.15561, 2021.

FastSpeech

- FastSpeech: fast speech generation with a novel NAR Transformer
- Architecture: feed-forward Transformer with length regulator
- Remove text-speech attention: wrong attention alignments cause problems of words skipping and repeating

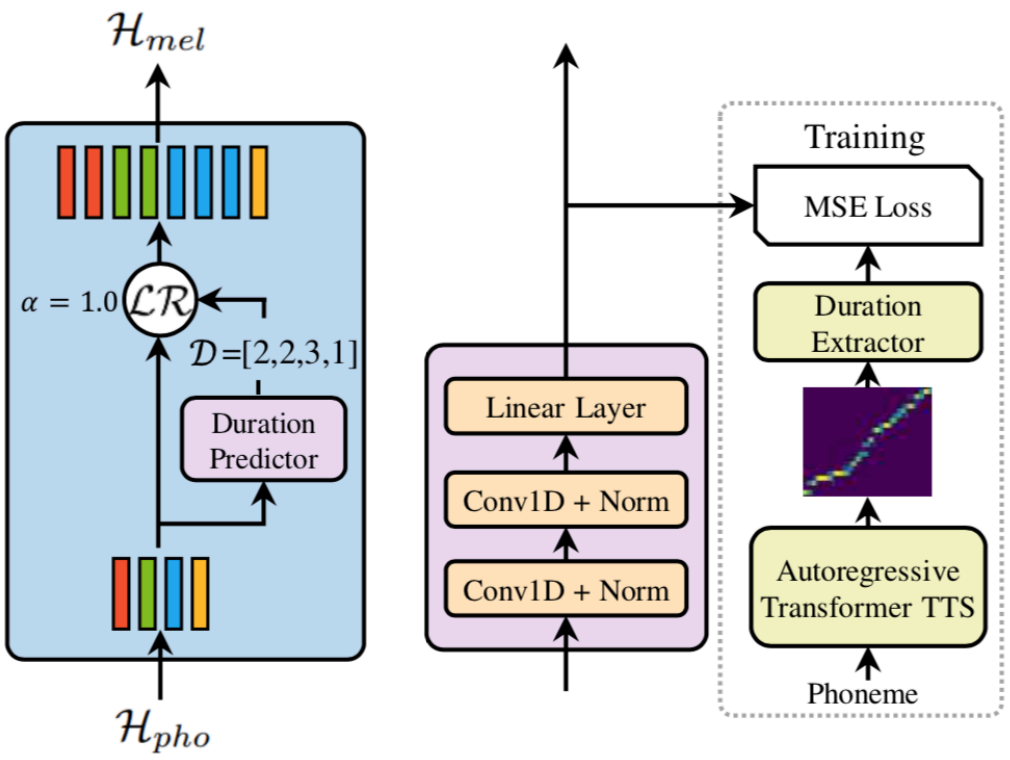


(a) Feed-Forward Transformer (b) FFT Block

Ren Y, Ruan Y, Tan X, et al. Fastspeech: Fast, robust and controllable text to speech. NIPS 2019.

FastSpeech

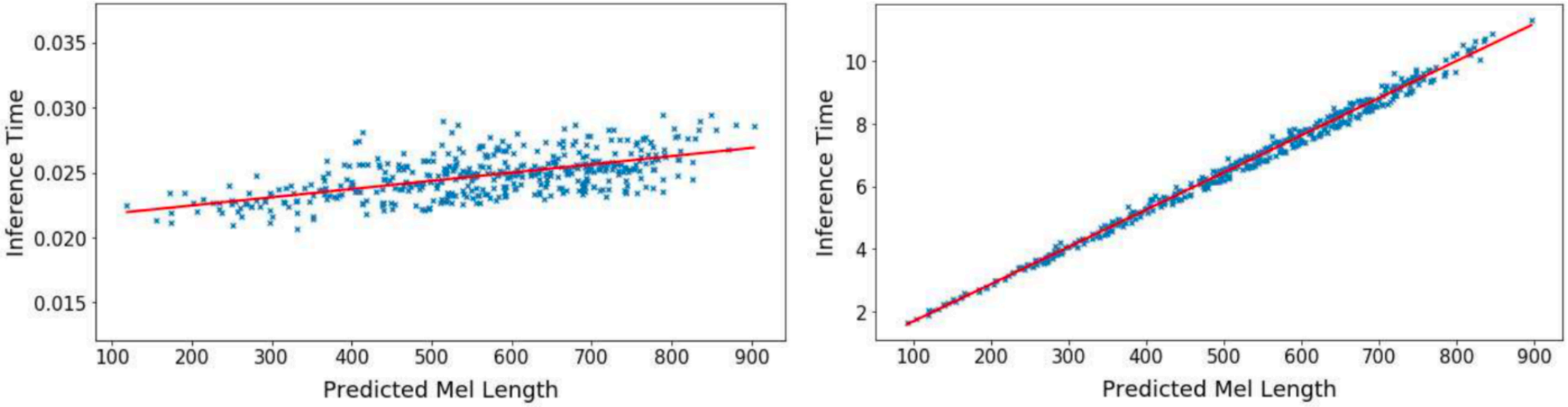
- Length regulator: upsample encoder outputs according to duration sequence D and voice speed α
- Duration predictor: predicts the duration for each phoneme
  - Similar to fertility in NAT
- Pretrain an AR TTS model, extract the golden duration and perform knowledge distillation
- Reduce multi-modality: use golden duration during training



(c) Length Regulator (d) Duration Predictor

FastSpeech

- Performance: nearly match the quality of the AR TTS model
- Speed: very significant speedup

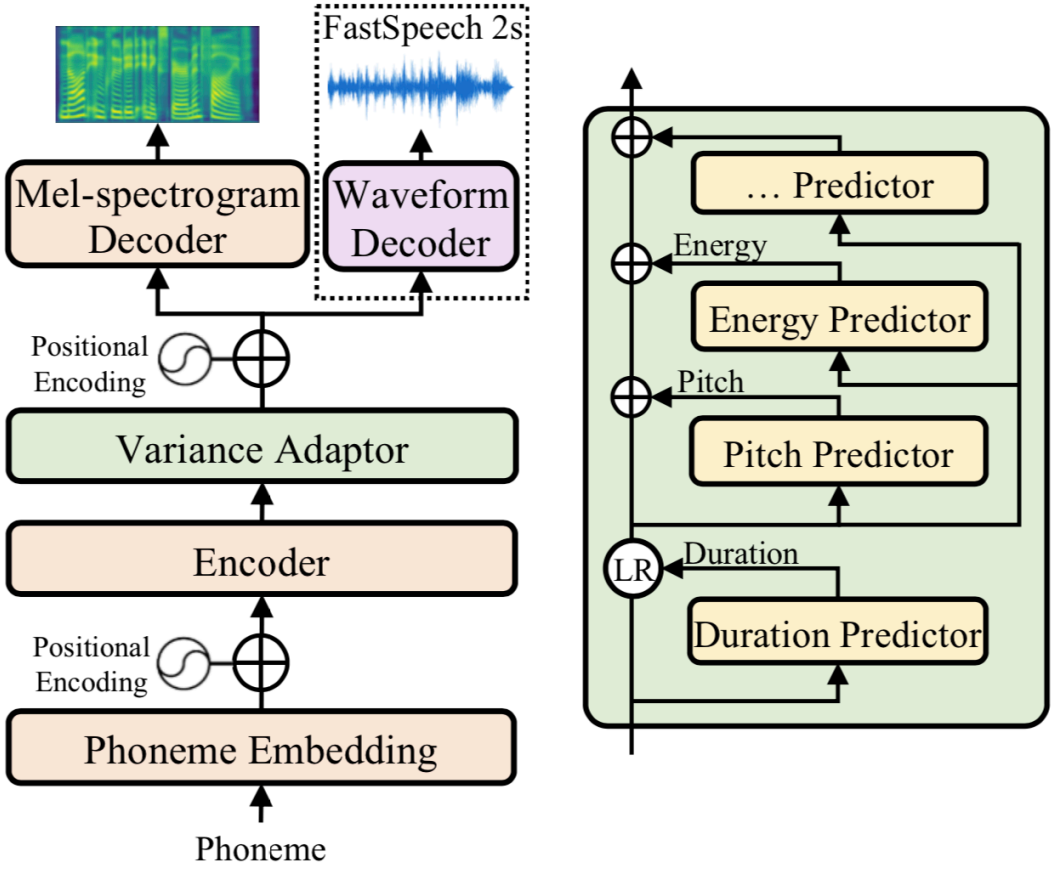


(a) FastSpeech (b) Transformer TTS

Figure 2: Inference time (second) vs. mel-spectrogram length for FastSpeech and Transformer TTS.

FastSpeech 2

- FastSpeech relies on an AR TTS model for KD and duration prediction
  - Duration is not accurate enough
  - Some information is lost in KD
- FastSpeech 2: train the model without KD, extract duration with a speech-text alignment tool
- TTS is also one-to-many mapping, without KD, need to further reduce multi-modality

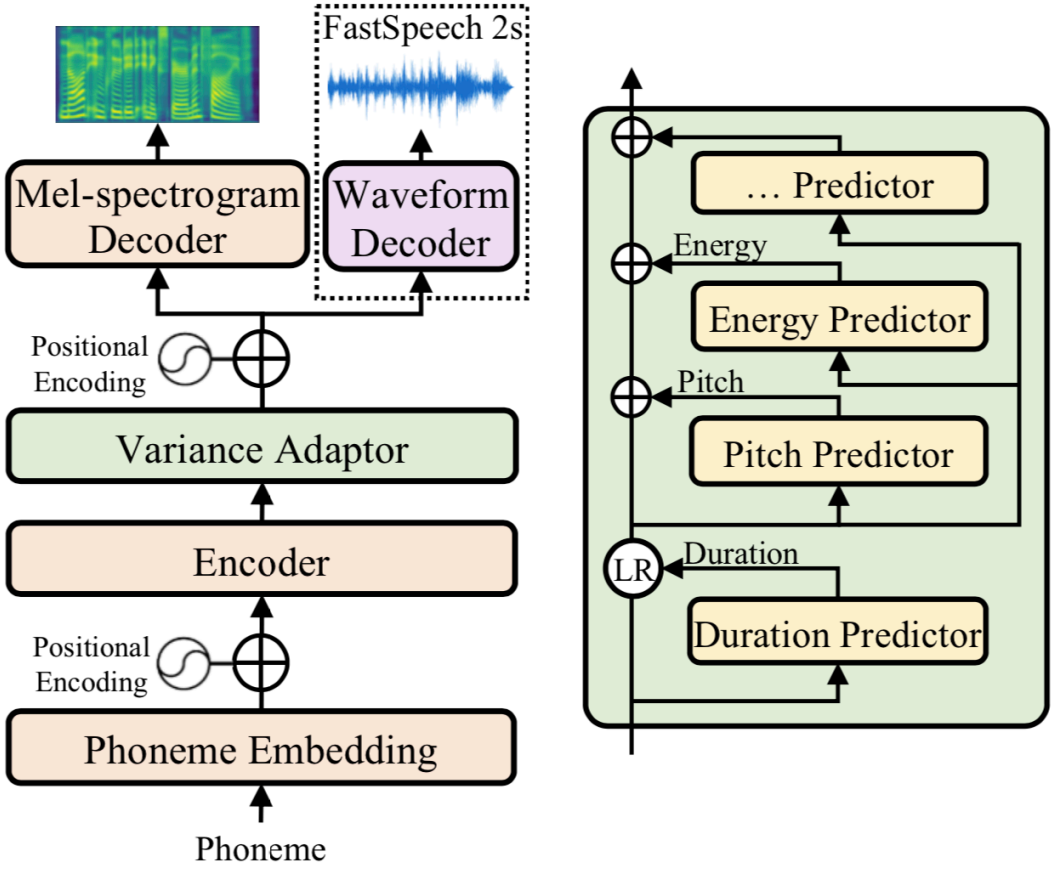


(a) FastSpeech 2 (b) Variance adaptor

Ren Y, Hu C, Tan X, et al. FastSpeech 2: Fast and High-Quality End-to-End Text to Speech. ICLR 2021.

FastSpeech 2

- Introduce more latent information to reduce multi-modality
  - duration, pitch, energy
  - duration: control the sequence length
  - pitch and energy: embeddings, added to decoder inputs
- More information can be added: emotion, style and speaker…
- Reduce multi-modality: use golden latent information in training, only use the predictor in inference



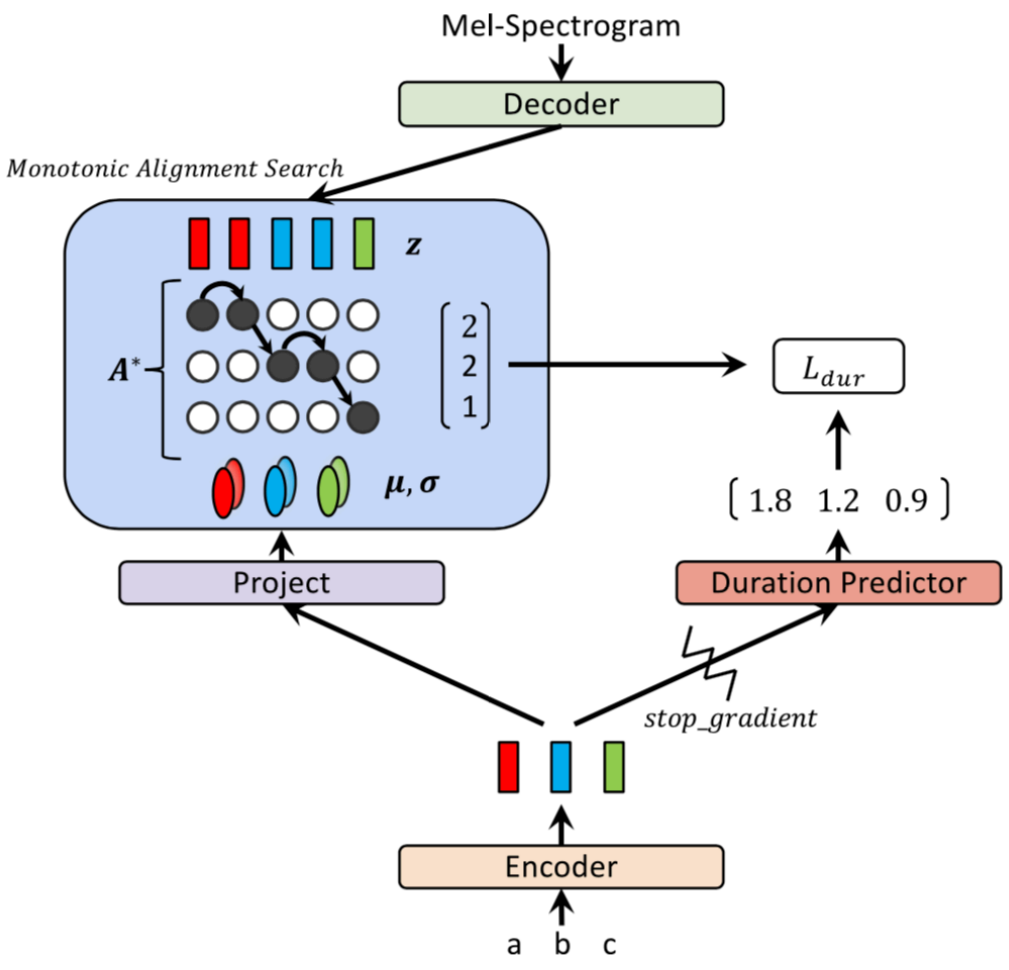
(a) FastSpeech 2 (b) Variance adaptor

Glow-TTS

- Glow-TTS: sequence generation with flow-based generative model
- Input text c, output spectrogram x, flow-based invertible decoder $f_{dec}: z \leftrightarrow x$
- Encoder predicts $z = f_{dec}^{-1}(x)$, decoder recovers $x = f_{dec}(z)$

$$\log P_X(x|c) = \log P_Z(z|c) + \log \left| \det \frac{\partial f_{dec}^{-1}(x)}{\partial x} \right|$$

encoder loss decoder loss



Kim J, Kim S, Kong J, et al. Glow-tts: A generative flow for text-to-speech via monotonic alignment search. NeurIPS 2020.

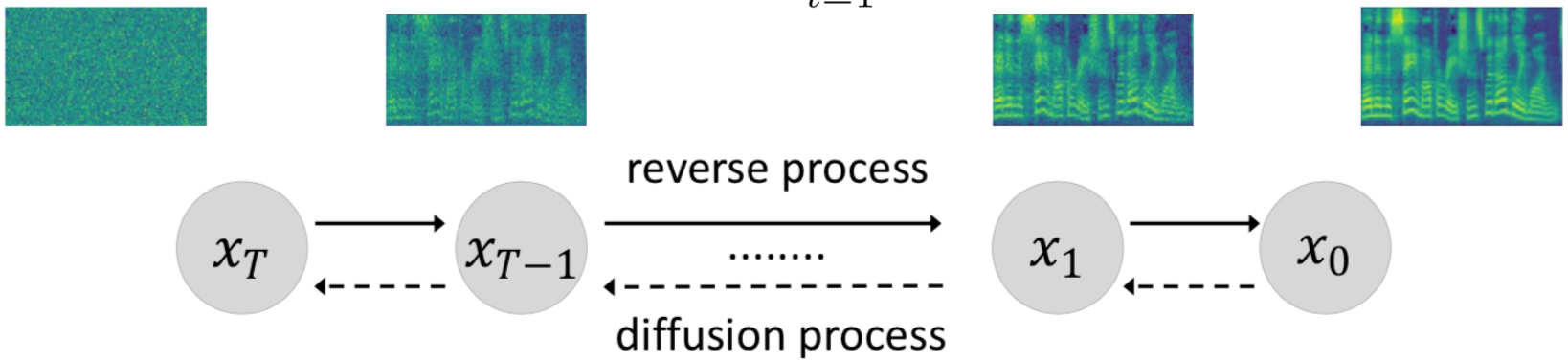
Diff-TTS

- Diff-TTS: sequence generation with diffusion process
- Forward diffusion process: training target x, add Gaussian noise to x in T steps, a sequence of noisy samples $x_0 \rightarrow x_1 \rightarrow \cdots \rightarrow x_T$:

$$q(x_t|x_{t-1}, c) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

- Reverse diffusion process: recreate the sample from the noisy input x_T:

$$p_\theta(x_0 \ldots, x_{T-1}|x_T, c) = \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t, c)$$



Jeong M, Kim H, Cheon S J, et al. Diff-tts: A denoising diffusion model for text-to-speech. arXiv preprint arXiv:2104.01409, 2021.

Vocoder

- Vocoder: generate waveform from mel-spectrogram
- Waveform is much longer than mel-spectrogram, AR generation is very slow
- Generative models for NAR generation: Flow/GAN/VAE /Diffusion

Vocoder	Input	AR/NAR	Modeling	Architecture
WaveNet [254]	Linguistic Feature	AR	/	CNN
SampleRNN [233]	/	AR	/	RNN
WaveRNN [150]	Linguistic Feature	AR	/	RNN
LPCNet [363]	BFCC	AR	/	RNN
Univ. WaveRNN [215]	Mel-Spectrogram	AR	/	RNN
SC-WaveRNN [265]	Mel-Spectrogram	AR	/	RNN
MB WaveRNN [418]	Mel-Spectrogram	AR	/	RNN
FFTNet [145]	Cepstrum	AR	/	CNN
Par. WaveNet [255]	Linguistic Feature	NAR	Flow	CNN
WaveGlow [279]	Mel-Spectrogram	NAR	Flow	Hybrid/CNN
FloWaveNet [163]	Mel-Spectrogram	NAR	Flow	Hybrid/CNN
WaveFlow [271]	Mel-Spectrogram	AR	Flow	Hybrid/CNN
SqueezeWave [433]	Mel-Spectrogram	NAR	Flow	CNN
WaveGAN [68]	/	NAR	GAN	CNN
GELP [149]	Mel-Spectrogram	NAR	GAN	CNN
GAN-TTS [23]	Linguistic Feature	NAR	GAN	CNN
MelGAN [178]	Mel-Spectrogram	NAR	GAN	CNN
Par. WaveGAN [402]	Mel-Spectrogram	NAR	GAN	CNN
HiFi-GAN [174]	Mel-Spectrogram	NAR	GAN	Hybrid/CNN
VocGAN [408]	Mel-Spectrogram	NAR	GAN	CNN
GED [96]	Linguistic Feature	NAR	GAN	CNN
Fre-GAN [161]	Mel-Spectrogram	NAR	GAN	CNN
Wave-VAE [268]	Mel-Spectrogram	NAR	VAE	CNN
WaveGrad [41]	Mel-Spectrogram	NAR	Diffusion	Hybrid/CNN
DiffWave [176]	Mel-Spectrogram	NAR	Diffusion	Hybrid/CNN
PriorGrad [185]	Mel-Spectrogram	NAR	Diffusion	Hybrid/CNN

Tan X, Qin T, Soong F, et al. A survey on neural speech synthesis. arXiv preprint arXiv:2106.15561, 2021.

Summary

- Why NAR generation is popular on speech related tasks?
  - Both the text (character-level) and speech (mel-spectrogram or waveform) are very long, NAR brings large speedup
  - Monotonic alignment between speech and text, easier for NAR generation (CTC in ASR, duration prediction in TTS)
- Techniques for the NAR generation of ASR and TTS
  - CTC is the core technique in non-autoregressive ASR
  - Generative models (GAN/Flow/VAE/Diffusion) are important in TTS
- Performance of NAR generation
  - NAR generation can match AR performance in TTS, but the gap still exists in ASR

Overview

- NAR generation is very popular on speech-related generation tasks
  - automatic speech recognition (ASR), text to speech (TTS)
- Are some tasks more difficult for NAR generation than others?
  - multi-modality, target-token dependency
- Is unconditional NAR generation possible?
  - language modeling, extremely high multi-modality
- More applications of NAR generation
  - dialogue generation, image captioning, object detection

Difficulty of NAR Generation

- So far, we have introduced three applications of NAR generation
  - NMT, ASR, TTS
- Are some tasks more difficult for NAR generation than others?
  - ASR and TTS seem to be easier than NMT: they do not heavily rely on knowledge distillation, and the performance gap with AR is small
- Intuitive multi-modality comparision
  - Monotonic alignment between speech and text: weak multi-modality in ASR, multi-modality in TTS can be reduced with duration
  - Non-monotonic alignment between source language and target language: strong multi-modality in NMT, complex source-target alignment

Multi-Modality and Dependency

- Difficult to compare the multi-modality of different tasks numerically

- Target-target dependency vs target-source dependency
  - If predicting a target token depends more on other target information, then the multi-modality is probably strong, difficult for NAR generation
  - If predicting a target token depends more on source information, then the multi-modality is probably weak, suitable for NAR generation

- Dependency can be measured by target-source and target-target attention

Ren Y, Liu J, Tan X, et al. A study of non-autoregressive model for sequence generation. ACL 2020.

Dependency

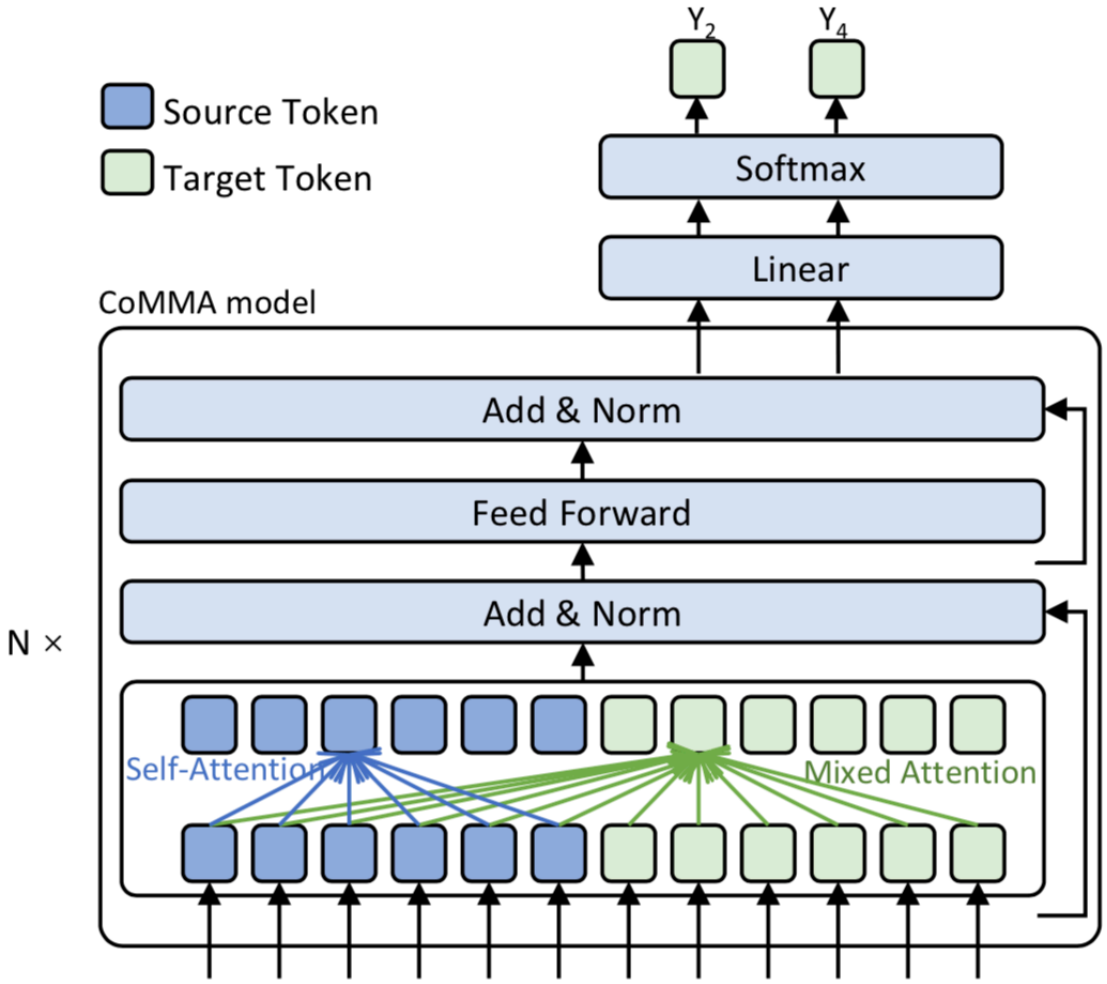
- Conditional masked prediction model with mix-attention (CoMMA)

$$\mathcal{L}_\theta = \log P(Y_{mask}|X, Y_{obs})$$

- Mixed attention in target-side:

$$\sum_{j=1}^{N+M} A_{i,j} = 1$$

- When predicting Y_{mask}, is it more dependent on the source sequence X, or the observed target Y_{obs}?

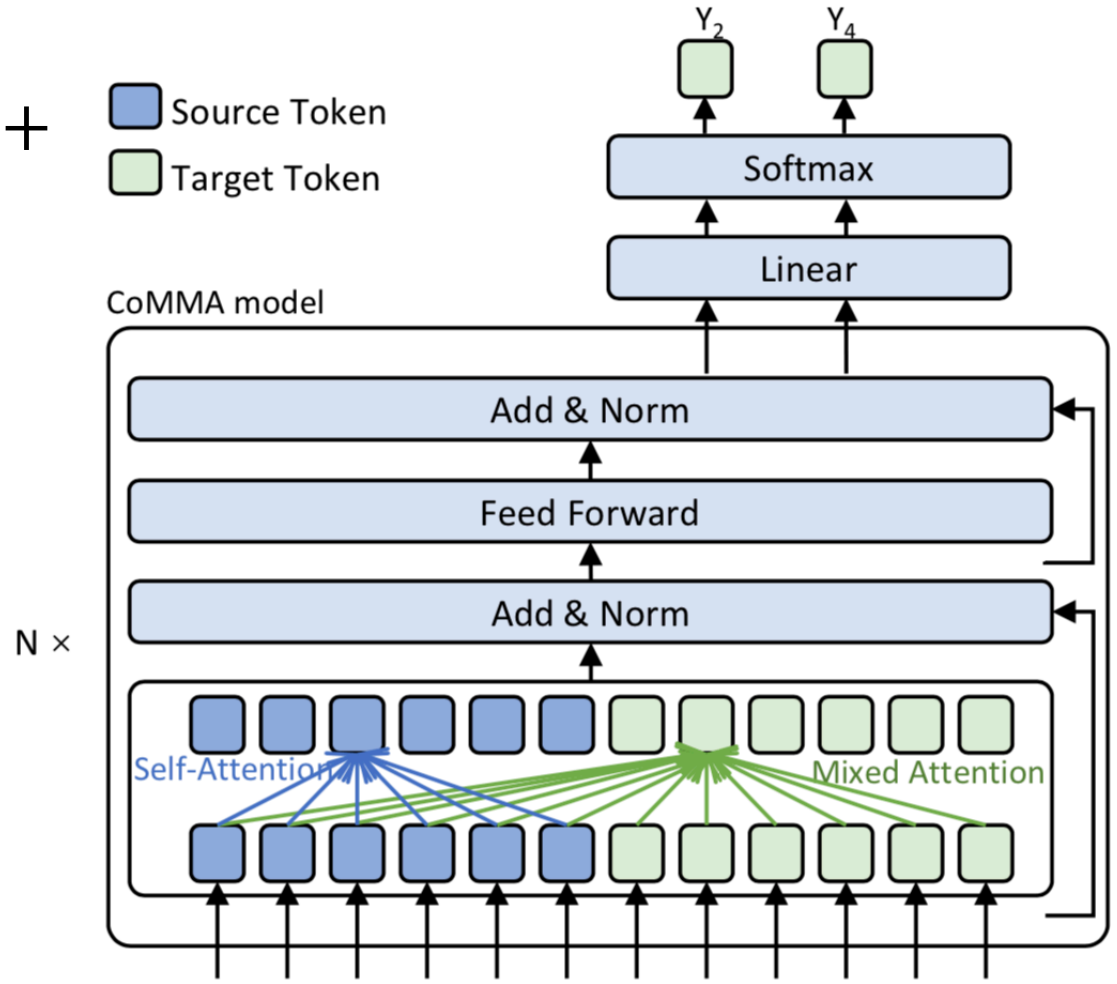


Dependency

- Attention matrix A, target token $i, j \in [1, N]$ represents target tokens, $j \in [N+1, N+M]$ represents source tokens
- Ratio of attention density on target context when predicting y_i

$$\alpha_i = \frac{\frac{1}{N}\sum_{j=1}^{N} A_{i,j}}{\frac{1}{N}\sum_{j=1}^{N} A_{i,j} + \frac{1}{M}\sum_{j=N+1}^{N+M} A_{i,j}}$$

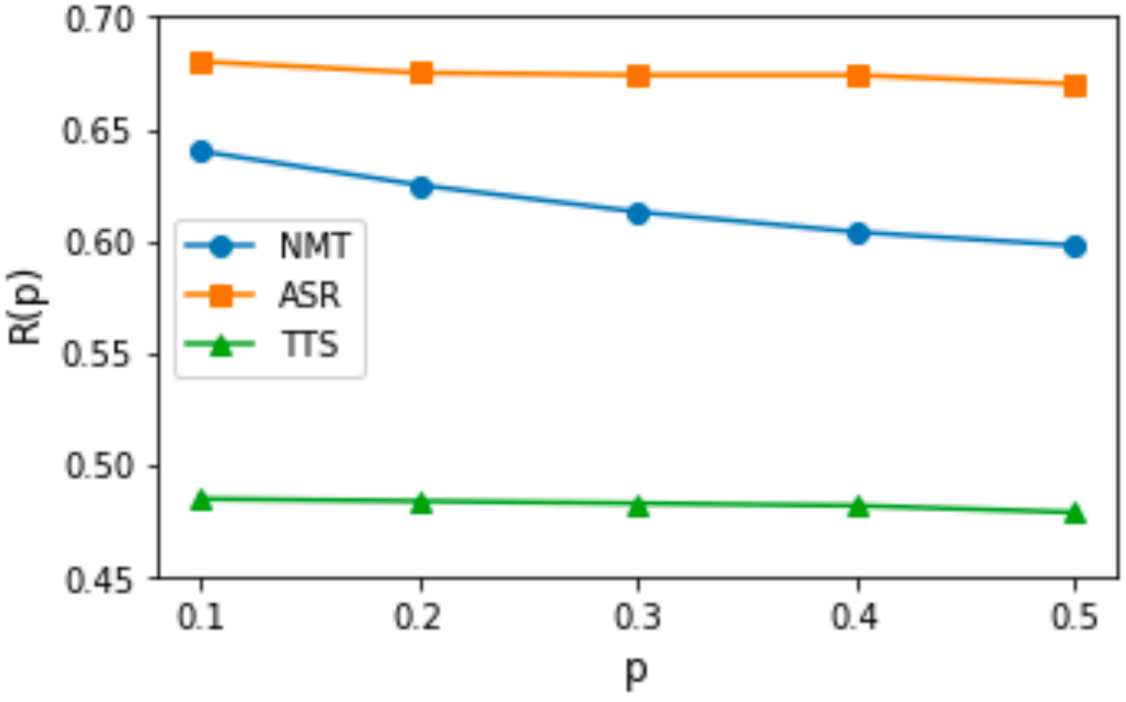
- α_i represents the proportion of target-target dependency



Dependency

- $R(p)$: average of α_i under mask ratio p
- Target-target dependency: ASR > NMT > TTS?
  - seems to be counter-intuitive
- Some concerns
  - Use knowledge distillation in NMT and TTS, but not in ASR
  - Attention score is averaged by length, α_i tends to be high if target length N << source length M (ASR), otherwise α_i tends to be small (TTS)

$$\alpha_i = \frac{\frac{1}{N}\sum_{j=1}^{N} A_{i,j}}{\frac{1}{N}\sum_{j=1}^{N} A_{i,j} + \frac{1}{M}\sum_{j=N+1}^{N+M} A_{i,j}}$$



Summary

- Are some tasks more difficult for NAR generation than others?
  - Intuitive multi-modality comparision: NMT > ASR, TTS
- Difficult to directly compare multi-modality, measure target-target dependency by attention

$$\alpha_i = \frac{\frac{1}{N}\sum_{j=1}^{N} A_{i,j}}{\frac{1}{N}\sum_{j=1}^{N} A_{i,j} + \frac{1}{M}\sum_{j=N+1}^{N+M} A_{i,j}}$$

- ASR > NMT > TTS, seems to be counter-intuitive
  - Maybe this metric is biased, larger when target length N << source length M?
- Still challenging to measure the NAR difficulty

Overview

- NAR generation is very popular on speech-related generation tasks
  - automatic speech recognition (ASR), text to speech (TTS)
- Are some tasks more difficult for NAR generation than others?
  - multi-modality, target-token dependency
- Is unconditional NAR generation possible?
  - language modeling, extremely high multi-modality
- More applications of NAR generation
  - dialogue generation, image captioning, object detection

Language Modeling

- NAR model can handle conditional sequence generation
  - NMT, ASR, TTS…
  - Difficulty may depend on multi-modality, target-source dependency
- Is unconditional NAR generation possible?
  - Extremely strong multi-modality
  - No target-source dependency
- Image generation is usually non-autoregressive, generative models
- NAR language modeling: GAN / Flow / Diffusion

GAN-LM

- Generative Adversarial Networks (GAN) for NAR generation
- Prior distribution $p(z)$, generator G maps latent space to data space:

$$\mathbf{z} \sim p(\mathbf{z}); \qquad \mathbf{x} = G(\mathbf{z})$$

- The generator G_θ and the discriminator D_ϕ play a minimax game:

$$\min_\theta \max_\phi V(D_\phi, G_\theta) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log D_\phi(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(1 - D_\phi(G_\theta(\mathbf{z})))]$$

- For image, the above objective can be directly optimized, but text is discrete, gradient cannot be passed from D_ϕ to G_θ

Huang F, Guan J, Ke P, et al. A text GAN for language generation with non-autoregressive generator. 2020.

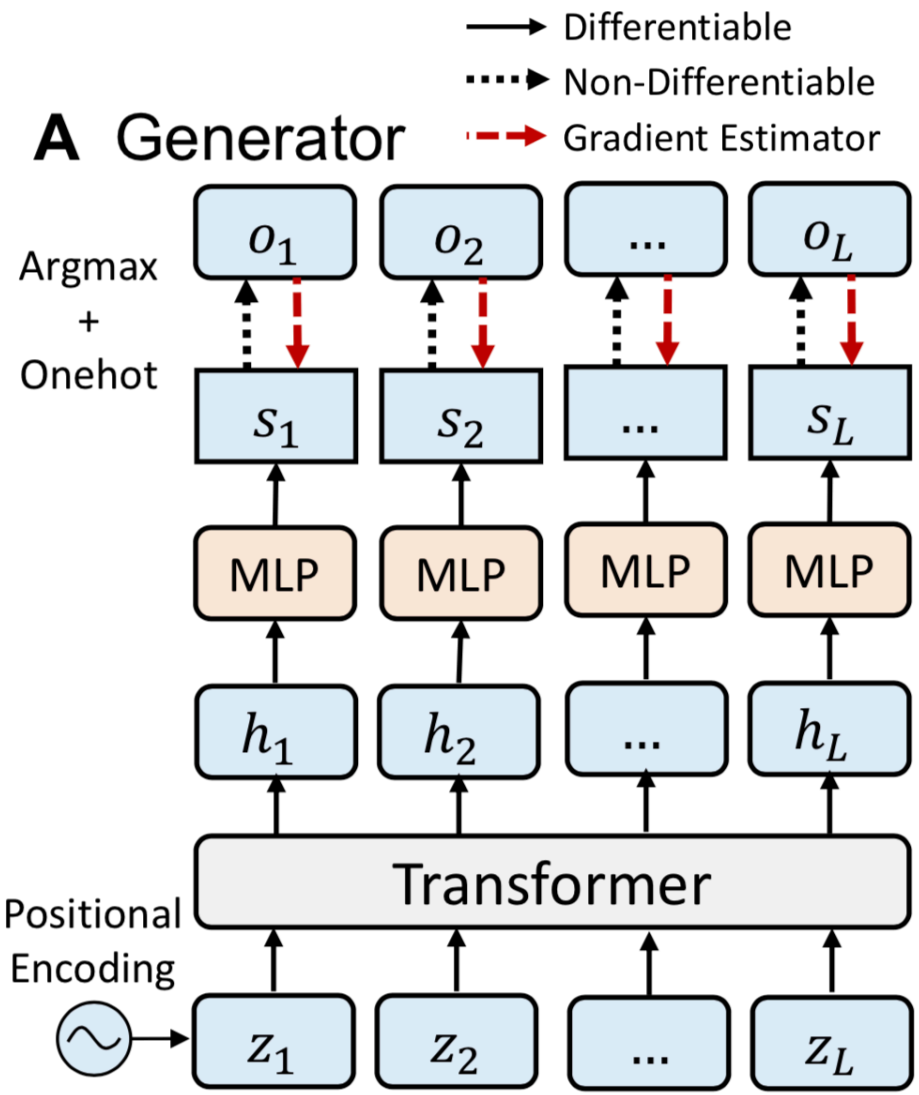
GAN-LM

- Usually apply RL in text GAN, but the training becomes very unstable
- For NAR model, directly use argmax operation to get the generator output o

$$[\boldsymbol{h}_1 \boldsymbol{h}_2 \cdots \boldsymbol{h}_L] = \text{Transformer}([\boldsymbol{z}_1 \boldsymbol{z}_2 \cdots \boldsymbol{z}_L]),$$

$$\boldsymbol{s}_t = \text{MLP}(\boldsymbol{h}_t) \in \mathbb{R}^V, \qquad \boldsymbol{o}_t = \text{onehot}(\arg\max_v(\boldsymbol{s}_{t,v}))$$

- Argmax is non-differentiable, use straight-through gradient estimator



GAN-LM

- Unconditional NAR generator can produce fluent sentences

Table 10: Cases of unconditional generation on the COCO dataset.

A group of people walking in the grass covered field .
A man riding a motorcycle on a snowy day .
A young man riding a skate board down a hill .
A woman is sitting on a table with a brown laptop .
A group of people standing next to a bunch of bananas .
Two people are sitting on the side of a road .
A big teddy bear laying on a rock in a field .
A group of young man riding a horse in the rain .
A stop sign with a city street .
A plate filled with chicken , and a salad in a basket sitting on a table .

Diff-LM

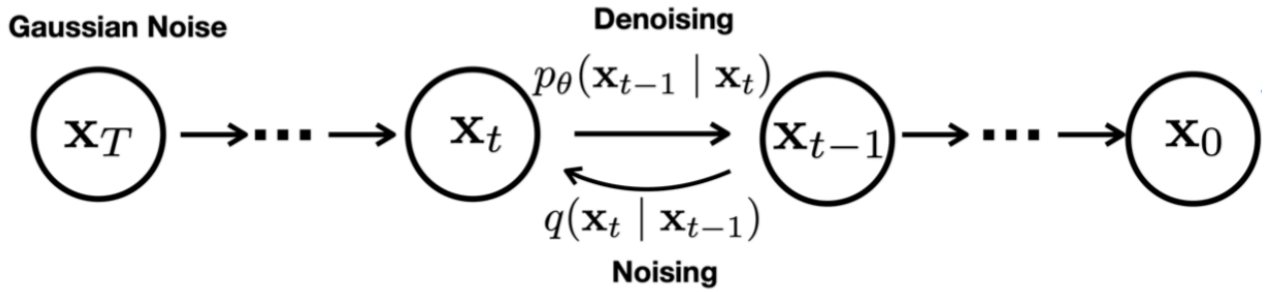
- Diffusion model: data $x_0 \in R^d$, forward diffusion process adds Gaussian noise to the sample in T steps $x_0 \to x_1 \to \cdots \to x_T$

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

- Sample from the diffusion model: reverse diffusion process, initial state $p_\theta(x_T) \sim N(0, \mathbf{I})$, denoising transition $x_t \to x_{t-1}$ is parametrized by model:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$$

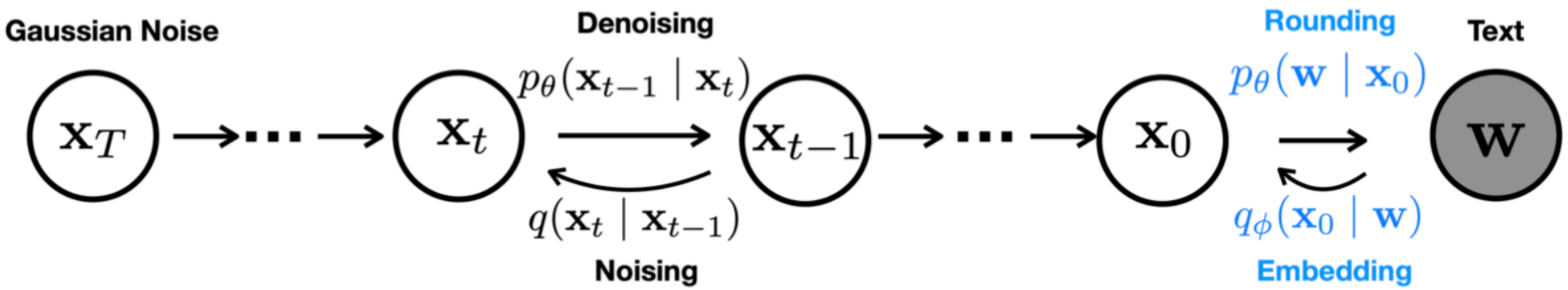
- Training: minimize KL-divergence between $p_\theta(x_t|x_{t+1})$ and $q(x_t|x_{t+1}, x_0)$



Li X L, Thickstun J, Gulrajani I, et al. Diffusion-LM Improves Controllable Text Generation. arXiv preprint arXiv:2205.14217, 2022.

Diff-LM

- Diffusion model is continuous, but text is discrete
- Input sequence $\mathbf{w}$ of length n, embedding $Emb(\mathbf{w}) \in R^{nd}$, forward process $q_\phi(x_0|w) = N(Emb(\mathbf{w}), \sigma_0 I)$
- Reverse process: add a trainable rounding step $p_\theta(w|x_0) = \prod_{i=1}^{n} p_\theta(w_i|x_i)$, probability of each step is a softmax distribution
- Diffusion model p_θ: Transformer encoder



Diff-LM

ROCStories+Aug	Matt was at the store . He was looking at a new toothbrush . He found the perfect one . When he got home , he bought it . It was bright and he loved it .
	I and my friend were hungry . We were looking for some to eat . We went to the grocery store . We bought some snacks . We decided to pick up some snacks .
	I was at the store . I had no money to buy milk . I decided to use the restroom . I went to the register . I was late to work .
	The man wanted to lose weight . He did n't know how to do weight . He decided to start walking . He ate healthy and ate less . He lost ten pounds in three months .
	I went to the aquarium . I wanted to feed something . I ordered a fish . When it arrived I had to find something . I was disappointed .

Summary

- Unconditional NAR generation is challenging but possible
- With generative models like GAN and Diffusion, NAR language model can produce fluent sentences
- There are also NAR LM based on normalizing flows
- In conditional NAR generation, these generative models are widely used in TTS (waveform generation), but less used in NMT and ASR

Overview

- NAR generation is very popular on speech-related generation tasks
  - automatic speech recognition (ASR), text to speech (TTS)
- Are some tasks more difficult for NAR generation than others?
  - multi-modality, target-token dependency
- Is unconditional NAR generation possible?
  - language modeling, extremely high multi-modality
- More applications of NAR generation
  - dialogue generation, image captioning, object detection

Dialogue Generation

- Dialogue generation: generate the response of dialog contexts

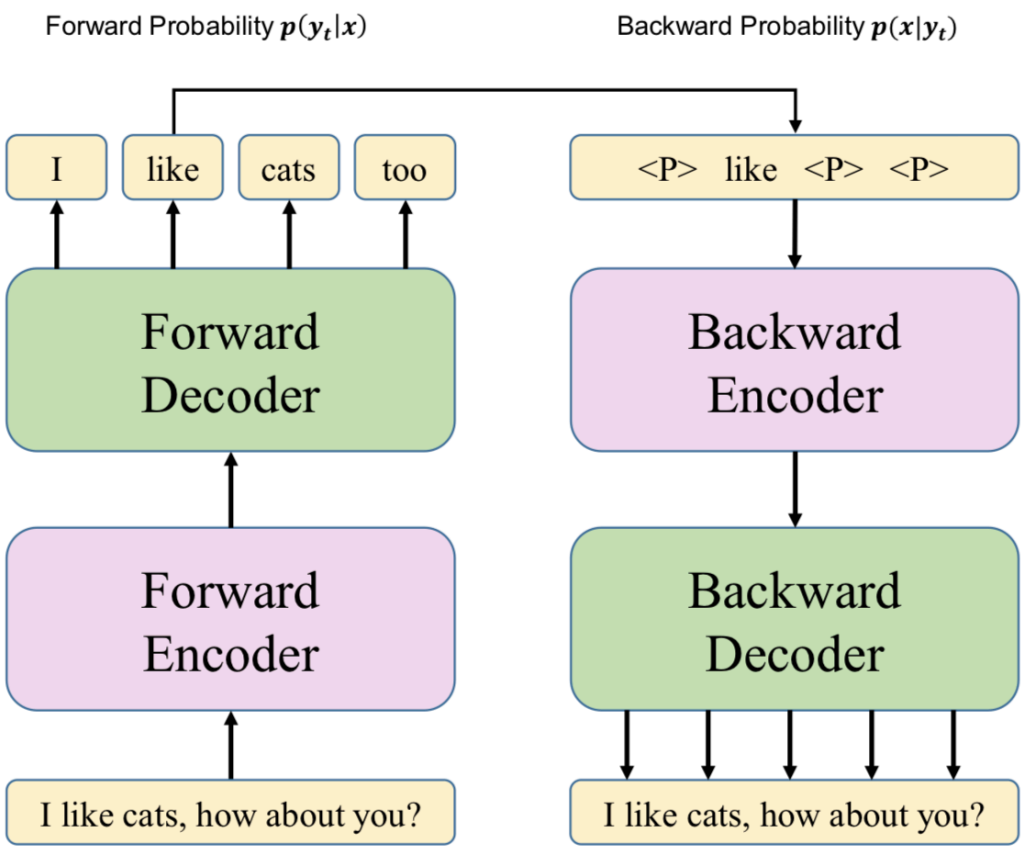
$$\text{forward_prob} = \prod_{t=1}^{t=L_y} p(y_t|x)$$

- Reconstruction task: recover the dialog contexts from response

$$\text{backward_prob} = [\prod_{t=1}^{t=L_y} p(x|y_t)]^{1/L_y}$$

- Decoding: maximum mutual information

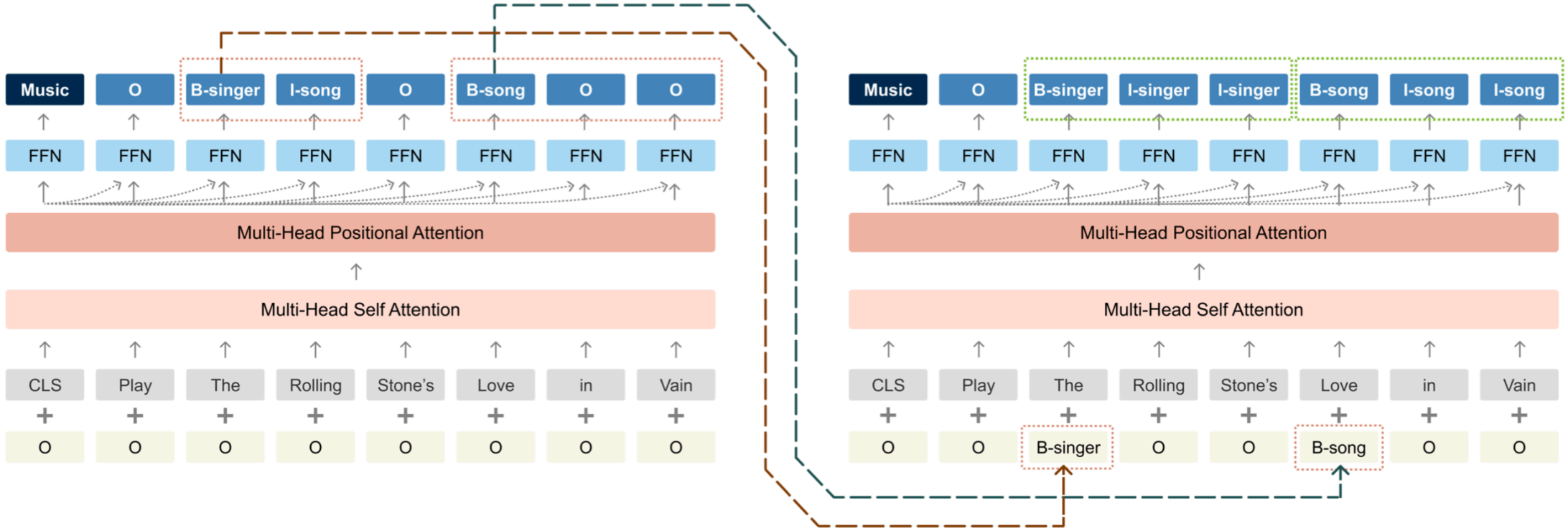
$$\hat{y} = \arg\max_{y} \left\{ (1-\lambda) \log p(y|x) + \lambda \log p(x|y) \right\}$$



Han Q, Meng Y, Wu F, et al. Non-autoregressive neural dialogue generation. arXiv preprint arXiv:2002.04250, 2020.

Dialogue Generation

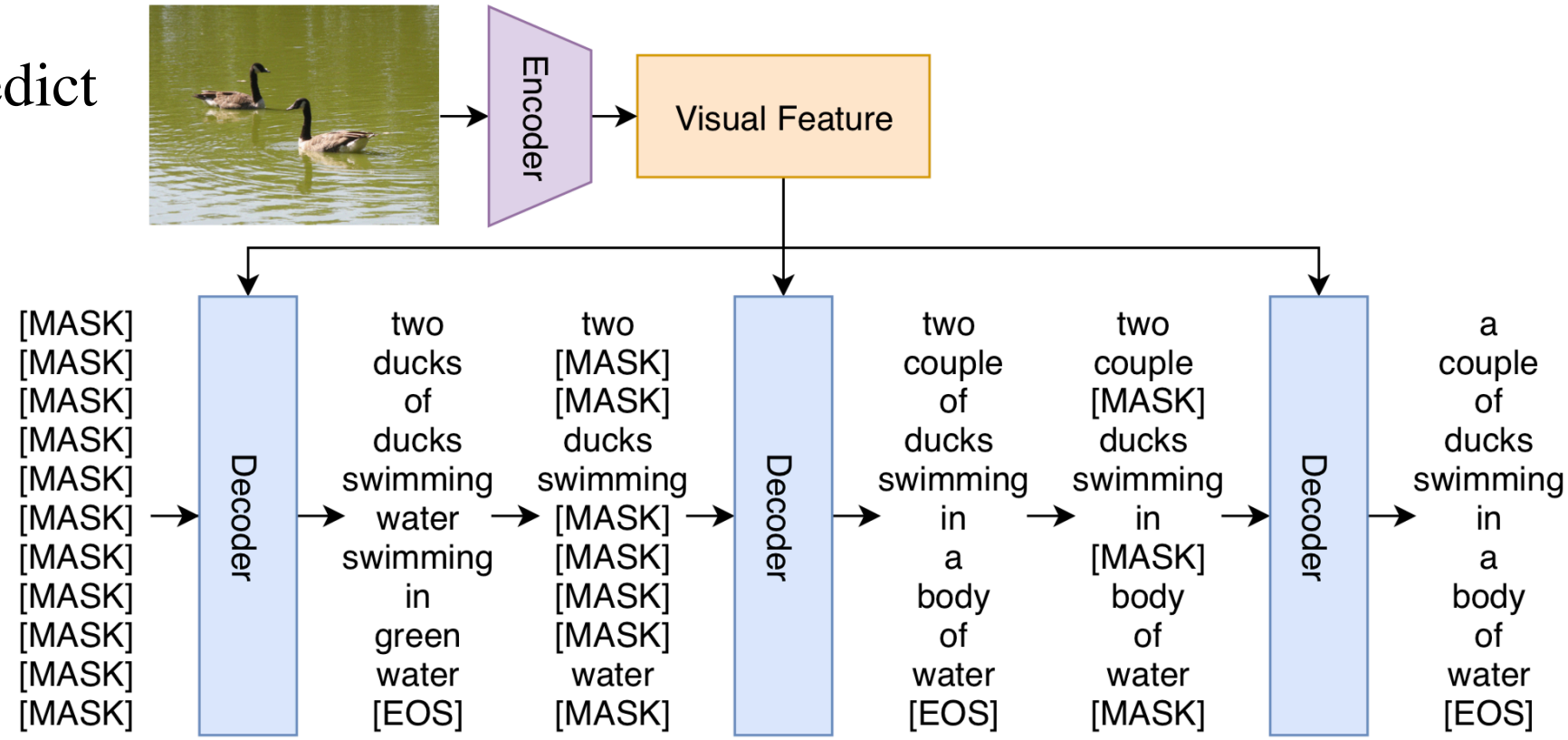
- Slot filling: identify from a running dialog different slots, a sequence labeling task with little multi-modality
- Two-pass refine: the output of begin-tags are fed to the next iteration



Wu D, Ding L, Lu F, et al. SlotRefine: A Fast Non-Autoregressive Model for Joint Intent Detection and Slot Filling. EMNLP 2020.

Image captioning

- Image captioning: generate the caption of image, task similar to machine translation, techniques in NAT are useful
- Mask-predict

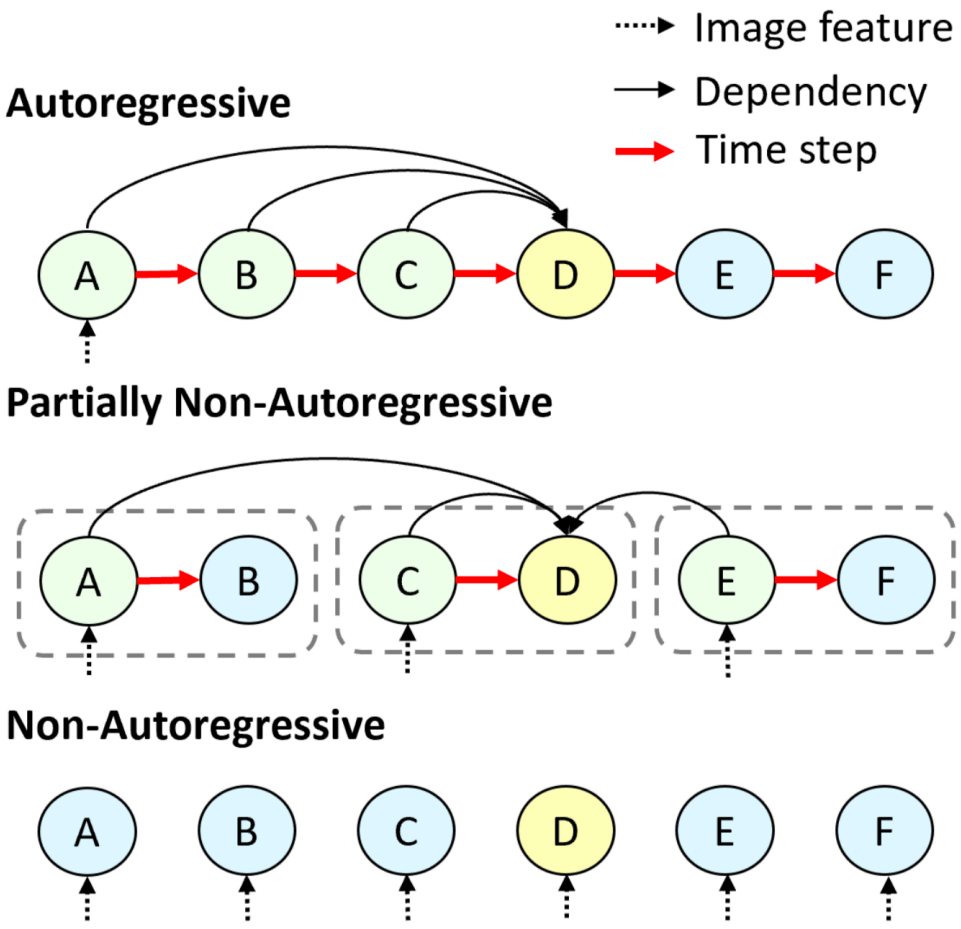


Gao J, Meng X, Wang S, et al. Masked non-autoregressive image captioning. arXiv preprint arXiv:1906.00717, 2019.

Image captioning

- Multi-thread decoding, generate K groups simultaneously

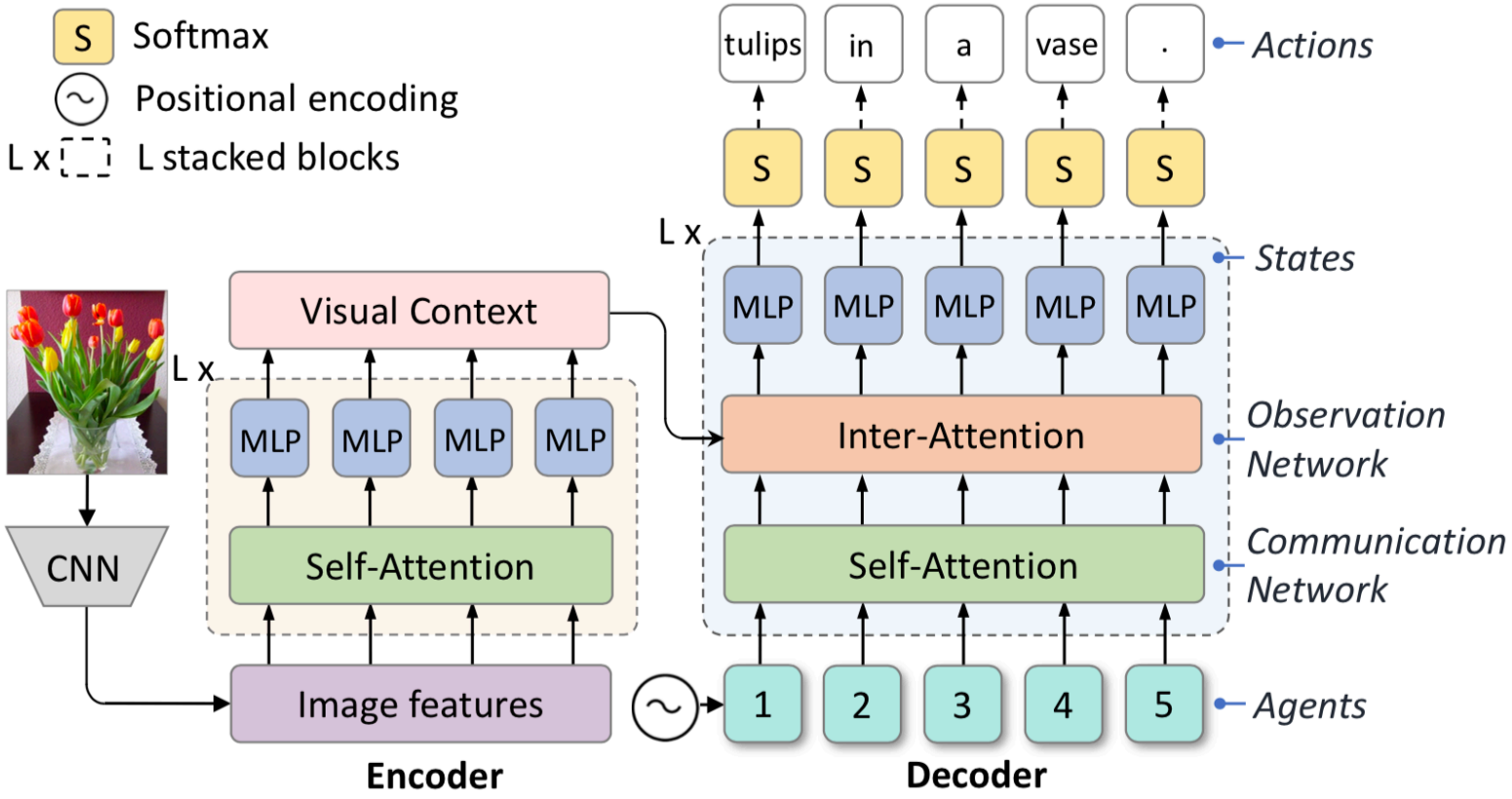
$$P(S|I) = \prod_{t=1}^{L_g} \prod_{i=1}^{K} P(w_t^i | G_{<t}^1, \ldots, G_{<t}^K; I_F)$$



Fei Z. Partially non-autoregressive image captioning. AAAI 2021.

Image captioning

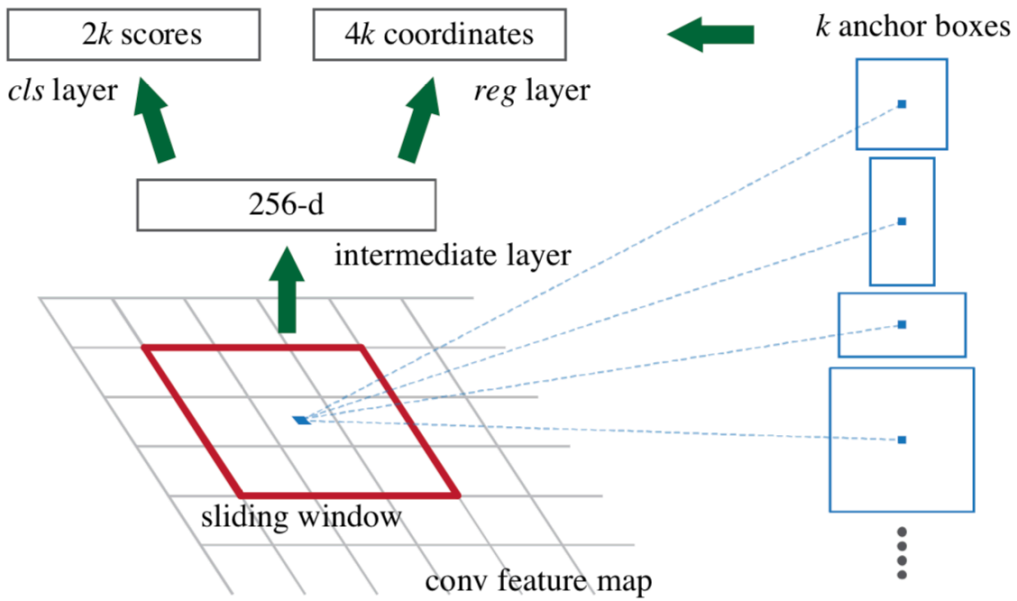
- Reinforcement learning, directly optimize the evaluation metric
- Positions as agents, predictions as actions, multi-agent RL

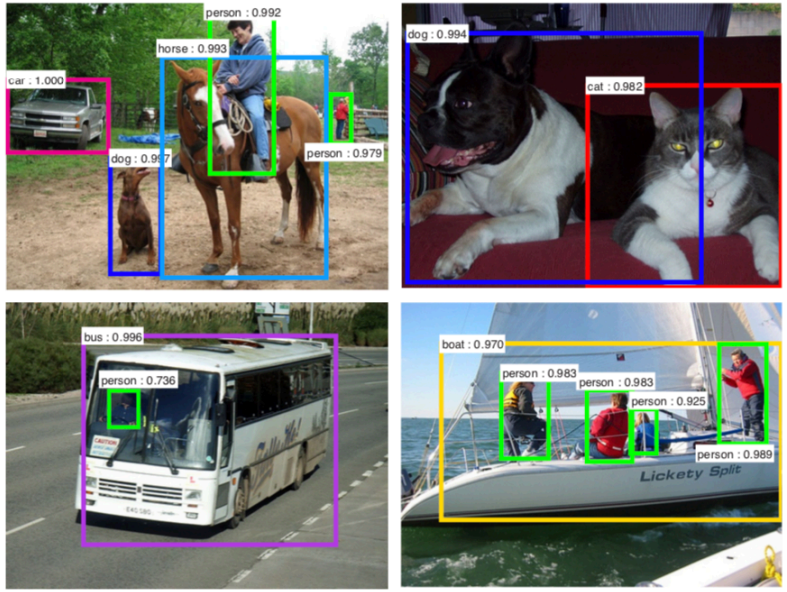


Guo L, Liu J, Zhu X, et al. Non-Autoregressive Image Captioning with Counterfactuals-Critical Multi-Agent Learning. IJCAI 2020.

Object Detection

- Object detection: a fundamental problem in computer vision
- Common approach: region proposal + classification
  - Slide a small network over the conv feature map to detect objects
- Yolo: divide the image to a 7 * 7 grids, has limitations

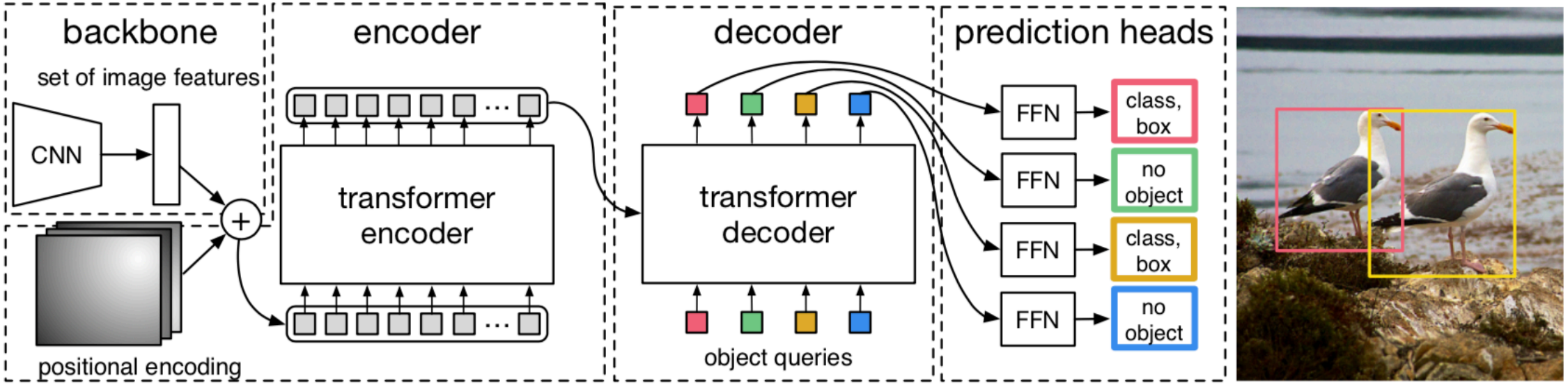




Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. NIPS 2015.
Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection. CVPR 2016.

Object Detection

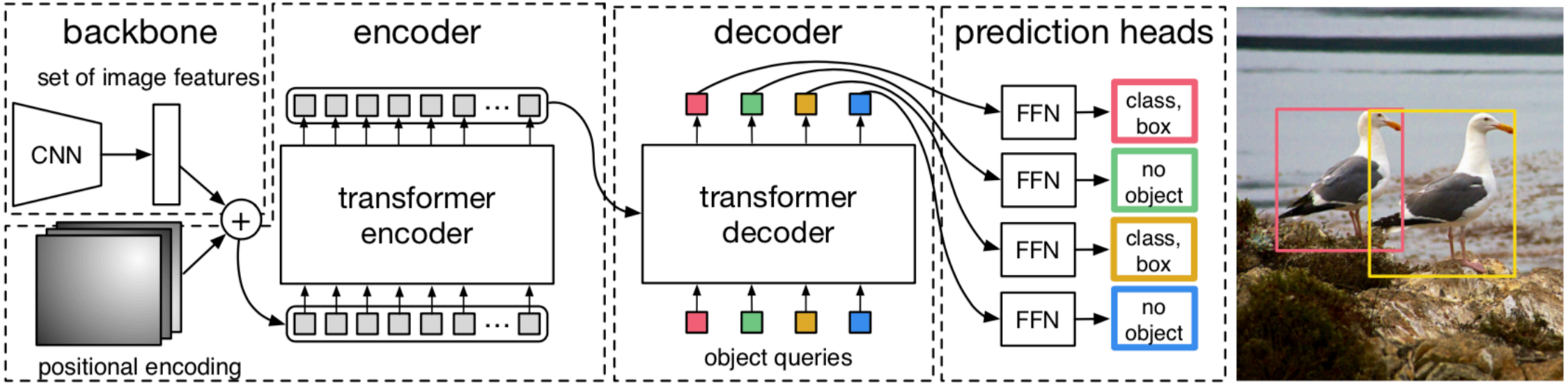
- End-to-End Object Detection with Transformers (DETR)
- A conventional CNN backbone, produce the image embedding for encoder
- The length of Transformer decoder is fixed (N=100, maximum number of objects), decoder inputs are purely positional embeddings



Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers. ECCV 2020.

Object Detection

- Decoder output $y_i = (c_i, b_i)$, c_i is class label and $b_i \in [0,1]^4$ defines the bounding box. Allow the prediction of "no object"
- There is no specific order between objects, any output order is legal
- Similar to the multi-modality problem, how to train the model?



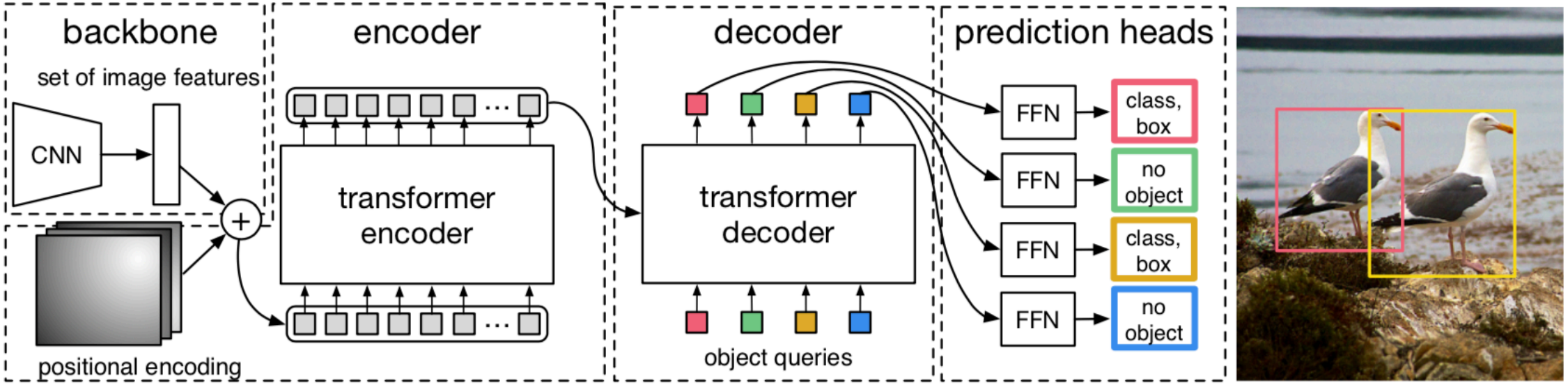
Object Detection

- Find the best matching to train the model, Hungarian algorithm

$$\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) = -\mathbb{1}_{\{c_i \neq \varnothing\}} \hat{p}_{\sigma(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\sigma(i)})$$

$$\hat{\sigma} = \arg\min_{\sigma \in \mathfrak{S}_N} \sum_{i}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)})$$

- The Hungarian loss is also useful in NAT (OAXE)



Summary

- NAR generation has wide applications
  - Dialogue generation, image captioning, object detection…
- Many techniques are transferable among tasks
  - The same problem: multi-modality
- Vanilla NAR model may not perform well, try these techniques when applying NAR generation to a new task

Methods	Descriptions
Knowledge distillation	Reduce multi-modality of training data
Latent modeling	Reduce multi-modality with latent information
Expressive ability	More expressive NAR models
Training objectives	Replace MLE with cross-word objectives
Decoding approaches	Decoding approaches besides fully NAR decoding

Outline

- Introduction (Yang Feng)
- Non-Autoregressive Translation (Yang Feng)
- Non-Autoregressive Generation (Chenze Shao)
- Conclusion and Future Direction (Chenze Shao)

Conclusion

- Multi-modality problem: multiple correct outputs for the same input
- Solutions from different perspectives

Methods	Descriptions
Knowledge distillation	Reduce multi-modality of training data
Latent modeling	Reduce multi-modality with latent information
Expressive ability	More expressive NAR models
Training objectives	Replace MLE with cross-word objectives
Decoding approaches	Decoding approaches besides fully NAR decoding

- Combining these methods in machine translation, the performance of state-of-the-art NAT systems are comparable with AR model

Conclusion

- NAR generation is popular on speech related tasks
  - CTC is the core technique in non-autoregressive ASR
  - Generative models (GAN/Flow/VAE/Diffusion) are important in TTS
- Some tasks more difficult for NAR generation than others
  - Still challenging to measure the NAR difficulty, a current solution: measure target-target dependency by attention
- Unconditional NAR generation is possible with generative models
- Other applications: dialogue generation, image captioning, object detection…

Future Direction

- Multi-modality is the major problem of NAR generation, need further improvements in the future
- KD provides a simplified reference but sets an upper bound for student
  - Diverse KD and rephraser explore the solution of dynamic reference
  - Use raw data, provide better dynamic reference, or other soultions?
- It seems to be a promising direction to further improve the expressiveness of NAR models

Model	Vanilla	CTC	DAT
Output space	VT	$VL + L$	$VL + L^2$
BLEU (raw)	11.4	18.4	26.5
BLEU (kd)	19.5	25.5	27.5

Future Direction

- Training objectives: mainly focus on finding a better (explicit or implicit) alignment between output and reference, are there better solutions with strong theoretical supports?

- VAE is widely used in latent modeling of NAR, are other generative models helpful (GAN, Flow, Diffusion)? Manually designed or automatically learned latent information, which one is better?

- The relationship between diffusion models and autoregressive or iterative models? Can we gain insights from diffusion and design better decoding approaches?

Future Direction

- How can we use NAR model to help AR generation, Transducer or a more efficient way?
- Parallel decoding but $O(T^2)$ attention complexity, maybe still slow when output sequence is long, need a linear complexity Transformer?
- Exploring the application of NAR generation on more tasks

Thank You for Listening!