Non-autoregressive neural machine translation

ICLR 2018

Jiatao Gu†∗, James Bradbury‡ , Caiming Xiong‡ , Victor O.K. Li† & Richard Socher‡

‡Salesforce Research

†The University of Hong Kong

Part 1.

Introduction

- **Neural network based models**
  - Recent neural network models are much slower than statistical mahcine translation approaches at inference time (Wu et al., 2016)
    - Model families use autoregressive decoders that operate one step at a time
    - Generate each token conditioned on the sequence of tokens previously generated
    - This process is not parallelizable and particularly slow
  - Some models avoid recurrence at train time by leveraging convolutions or self-attention
    - More-parallelizable alternatives to RNNs
    - But still use of autoregressive decoding makes it impossible to take full advantage of parallelism during inference

Y. Wu et al. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144, 2016.

Part 1.

Introduction

- **A non-autoregressive translation model based on the Transformer network**
  - Modify the encoder of the original Transformer network
    - Add a module that predicts fertilities, sequences of numbers that form an important component of many traditional machine translation models(Brown et al., 1993)
  - Fertility
    - Supervised training
    - Provide the decoder at inference time with a globally consistent plan on which to condition its simultaneously computed outputs

Peter Brown et al. The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics, 19(2):263–311, 1993.

Background: Autoregressive Neural Machine Translation

- **Autoregressive Neural Machine Translation**
  - A model factors the distribution over possible output sentences into a chain of conditional probabilities with a left-to-right causal structure

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

  - Where the special tokens y_0 (e.g. $\langle \text{bos} \rangle$) and y_{T+1} (e.g. $\langle \text{eos} \rangle$) are used to represent the beginning and end of all target sentences

  - An encoder-decoder architecture (Sutskever et al., 2014) with a unidirectional RNN-based decoder is used to capture the causal structure of the output distribution

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

Ilya Sutskever et al. Sequence to sequence learning with neural networks. NIPS. 2014.

Background: Autoregressive Neural Machine Translation

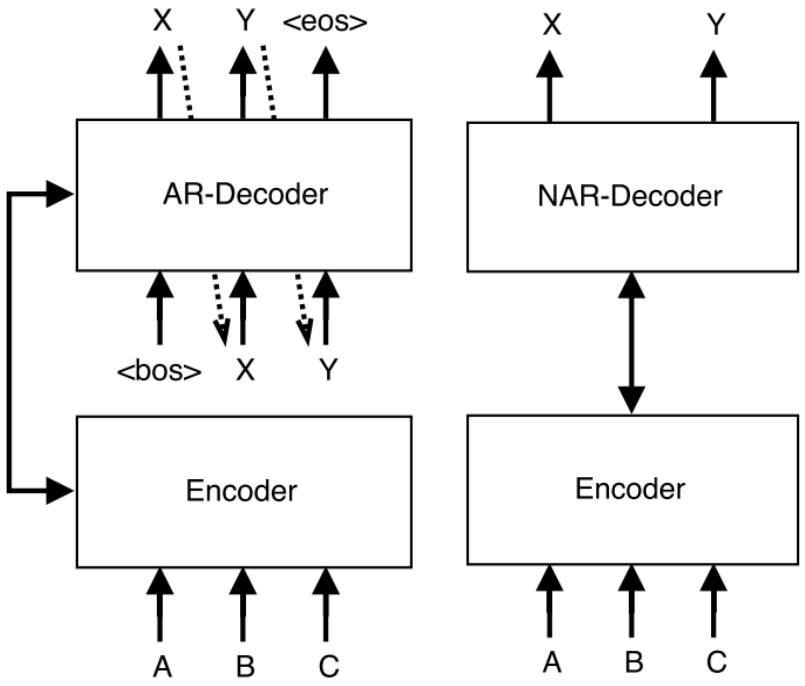
- **Maximum Likelihood training**
  - Factorize the machine translation output distribution autoregressively
  - Straightforward maximum likelihood training with a cross-entropy loss applied at each decoding step
  - This loss provides direct supervision for each conditional probability prediction

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

- **Autoregressive NMT without RNNs**
  - Even though decoding must remain entirely sequential during inference, models can take advantage of this parallelism during training
  - Transformer network which reduces sequential computation
    - Allow information to flow in the decoder across arbitrarily long distances in a constant number of operations, asymptotically fewer than required by convolutional architectures

Background: Non-autoregressive Decoding

- **Pros and cons of autoregressive decoding**



Pros

- Effectively captures the distribution of real translations according to the word-by-word nature of human language production
- Easy to train with SOTA performance
- Beam search: an effective local search method for finding approximately-optimal output translations

Cons

- Sequential decoding:
  Autoregressive decoding prevents architectures from fully realizing train-time performance advantage during inference
- Beam search:
  - Suffer from diminishing returns with respect to beam size
  - Exhibit limited search parallelism because of computational dependence between beams

Background: Non-autoregressive Decoding

- **Towards non-autoregressive decoding**
  - Remove the autoregressive connection directly from an existing encoder-decoder model
  - Assumption: Target sequence length T can be modeled with a separate conditional distribution

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

  - An explicit likelihood function: still use independent cross-entropy losses on each output distribution during training
  - Parallelism: These distributions can be computed in at inference time

Background: Non-autoregressive Decoding

- **Multimodality Problem: Conditional independence assumption**
  - The assumption prevents a model from properly capturing the highly multimodal distribution of target translations
  - Complete conditional independence assumption : Naive approach doesn't yield good results
    - Each token's distribution $p(y_t)$ depends only on the source sentence X
      - ➡ A poor approximation to the true target distribution
      - ➡ True target distribution exhibits strong correlation across time
    - Intuitively, such a decoder is akin to a panel of human translators
      - Ask to provide a single word independently of the words their colleagues choose

Source Sentence

"Thank you."

➡

Target distribution

"Danke."	"Danke schön."	"Vielen Dank."

Source Sentence

"Thank you."

➡

A conditionally independent distribution

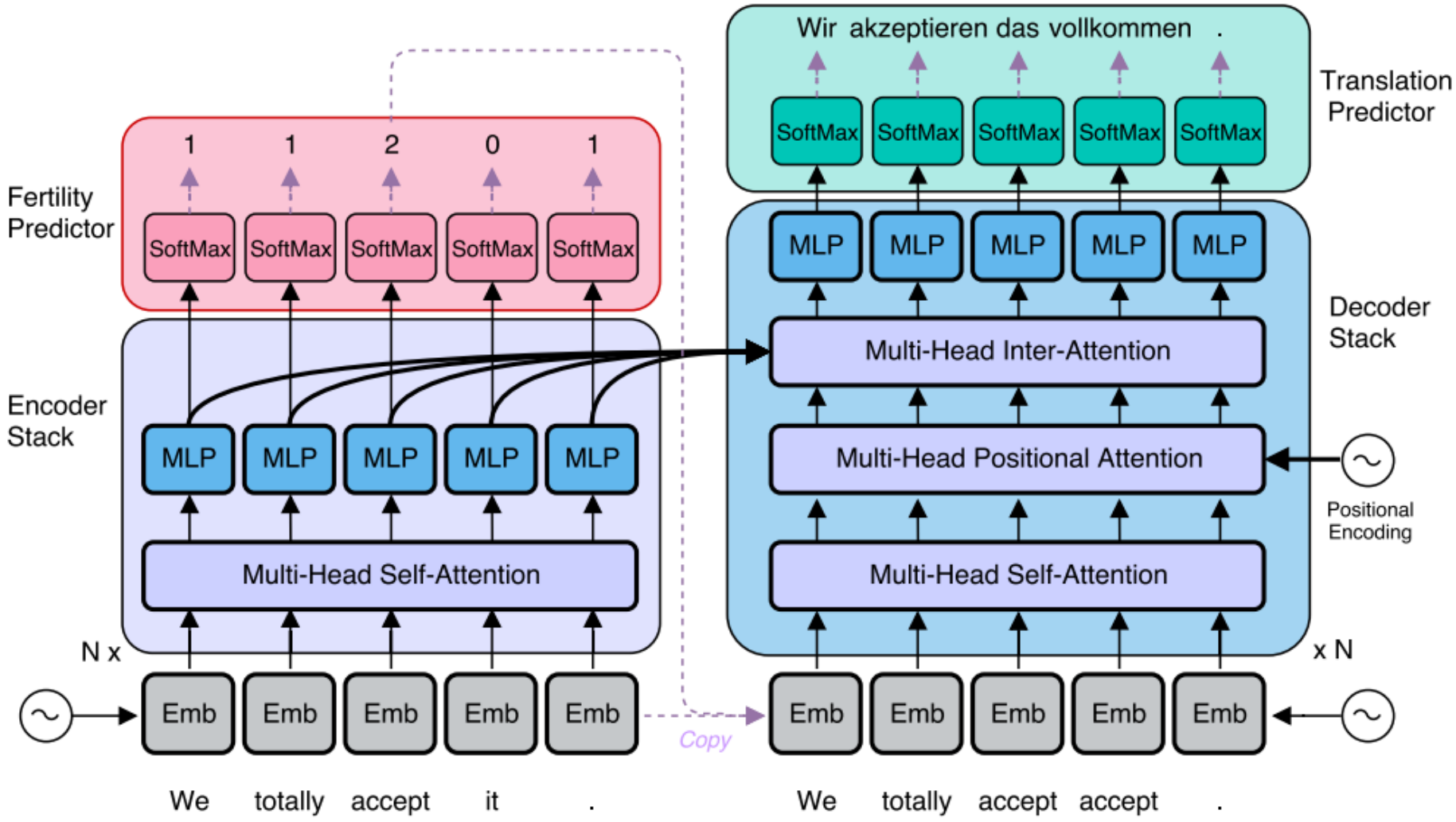
"Danke."	"Danke Dank."	"Vielen schön."

Vocabulary

- schön: 정말
- Vielen: 매우
- Danke: 감사합니다

The Non-autoregressive Transformer (NAT)

- **Multimodality Problem**



The Non-autoregressive Transformer (NAT)

- **Encoder Stack**
  - Similar to the autoregressive Transformer
    - Both the encoder & decoder stacks are composed entirely of FFN (MLPs) & MHN modules
    - Since no RNNs are used, there is no inherent requirement for sequential execution

The Non-autoregressive Transformer (NAT)

- **Decoder Inputs**
  - Before decoding starts
    - The NAT needs to know how long the target sentence will be for generation in parallelable
  - The first decoder layer
    - Cannot use time-shifted target outputs as the inputs (during training)
    - Cannot use previously predicted outputs as the inputs (during inference)
  - Initialize the decoding process using copied source inputs from the encoder side
    - Omitting inputs to the first decoder layer entirely, or using only positional embeddings, resulted in very poor performance

The Non-autoregressive Transformer (NAT)

- **Initialize the decoding process for Decoder Inputs**

  As the source and target sentences are often of different lengths

  - Copy source inputs uniformly:
    - Each decoder input t is a copy of the $\text{Round}(T't/T)$-th encoder input
    - Equivalent to "scanning" source inputs from left to right with a constant "speed"
    - Results in a decoding process that is deterministic given a (predicted) target length

  - Copy source inputs using fertilities (A more powerful way):
    - Copy each encoder input as a decoder input zero or more times, with the number of times each input is copied referred to as that input word's "fertility."
    - "scanning" source inputs from left to right with a "speed" that varies inversely with the fertility of each input
    - The decoding process is now conditioned on the sequence of fertilities
    - The resulting output length is determined by the sum of all fertility values

The Non-autoregressive Transformer (NAT)

- **Non-causal self-attention**
  - Without the constraint of an autoregressive factorization of the output distribution
    - No longer need to prevent earlier decoding steps from accessing information from later steps
  - Avoid the causal mask used in the self-attention module of Transformer's decoder
    - Instead, mask out each query position only from attending to itself
    - Improve decoder performance relative to unmasked self-attention

- **Positional attention**
  - Include an additional positional attention module in each decoder layer
  - Incorporate positional information directly into the attention process
  - Provide a stronger positional signal than the embedding layer alone
  - Hypothesize that this additional information improves the decoder's ability to perform local reordering

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

The Non-autoregressive Transformer (NAT)

- **Modeling Fertility to Tackle The Multimodality Problem**
  - Multimodality problem can be attacked by introducing a latent variable z to directly model the non-determinism in the translation process
    - First sample z from a prior distribution and then condition on z to non-autoregressively generate a translation

One way to interpret this latent variable:

a sentence-level "plan" akin to those discussed in the language production literature (Martin et al., 2010)

  - Desirable properties for this latent variable
    - It should be simple to infer a value for the latent variable given a particular input-output pair, as this is needed to train the model end-to-end
    - Adding z to the conditioning context should account as much as possible for the correlations across time between different outputs, so that the remaining marginal probabilities at each output location are as close as possible to satisfying conditional independence
    - It should not account for the variation in output translations so directly that $p(y|x,z)$ becomes trivial to learn, since that is the function our decoder neural network wi

Randi Martin et al. Planning in sentence production: Evidence for the phrase as a default planning scope. Cognition, 116 (2):177–192, 2010.

The Non-autoregressive Transformer (NAT)

- **Modeling Fertility to Tackle The Multimodality Problem**
  - The factorization by length provides a very weak example of a latent variable model, satisfying the first and third property but not the first

    Property:
    - It should be simple to infer a value for the latent variable given a particular input-output pair
    - Adding to the conditioning context should account for the correlations between outputs
    - It shouldn't account for the variation in outputs so the function our decoder will approximate

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

  - Propose the use of fertilities instead
    - The number of words in the source & target sentence can be aligned to that source word using a hard alignment algorithm like IBM Model 2 (Brown et al., 1993)

Peter Brown et al. The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics, 19(2):263–311, 1993.

The Non-autoregressive Transformer (NAT)

- **Modeling Fertility to Tackle The Multimodality Problem**
  - One of the most important properties of the proposed NAT
    - Naturally introduces an informative latent variable
      when we choose to copy the encoder inputs based on predicted fertilities

$$p_{\mathcal{NA}}(Y|X;\theta) = \sum_{f_1,\ldots,f_{T'}\in\mathcal{F}} \left(\prod_{t'=1}^{T'} p_F(f_{t'}|x_{1:T'};\theta) \cdot \prod_{t=1}^{T} p(y_t|x_1\{f_1\},..,x_{T'}\{f_{T'}\};\theta) \right)$$

$$\mathcal{F} = \{f_1,\ldots,f_{T'} | \textstyle\sum_{t'=1}^{T'} f_{t'} = T, f_{t'} \in \mathbb{Z}^*\}$$

Sentence Length

$l = 6, m = 7$

- The set of all fertility sequences -
  One fertility value per source word - that sum to the length of Y
- $x\{f\}$ denotes the token x repeated f times

Source Sentence

$e = And\ the\ programme\ has\ been\ implemented$

Target Setntence

$f = Le\ programme\ a\ ete\ mis\ en\ application$

The Non-autoregressive Transformer (NAT)

- **Alignment Models: IBM Model 2**
  - $t(f|e)$: Conditional probability of generating target word f from source word e
  - $q(j|i,l,m)$: Probability of alignment variable a_i (The value j , source & target l & m)

$$p(f_1 \dots f_m, a_1 \dots a_m | e_1 \dots e_l, m) = \prod_{i=1}^{m} q(a_i|i,l,m) t(f_i|e_{a_i})$$

Alignment

Le	$\Rightarrow$	the
Programme	$\Rightarrow$	program
a	$\Rightarrow$	has
ete	$\Rightarrow$	been
mis	$\Rightarrow$	implemented
en	$\Rightarrow$	implemented
application	$\Rightarrow$	implemented

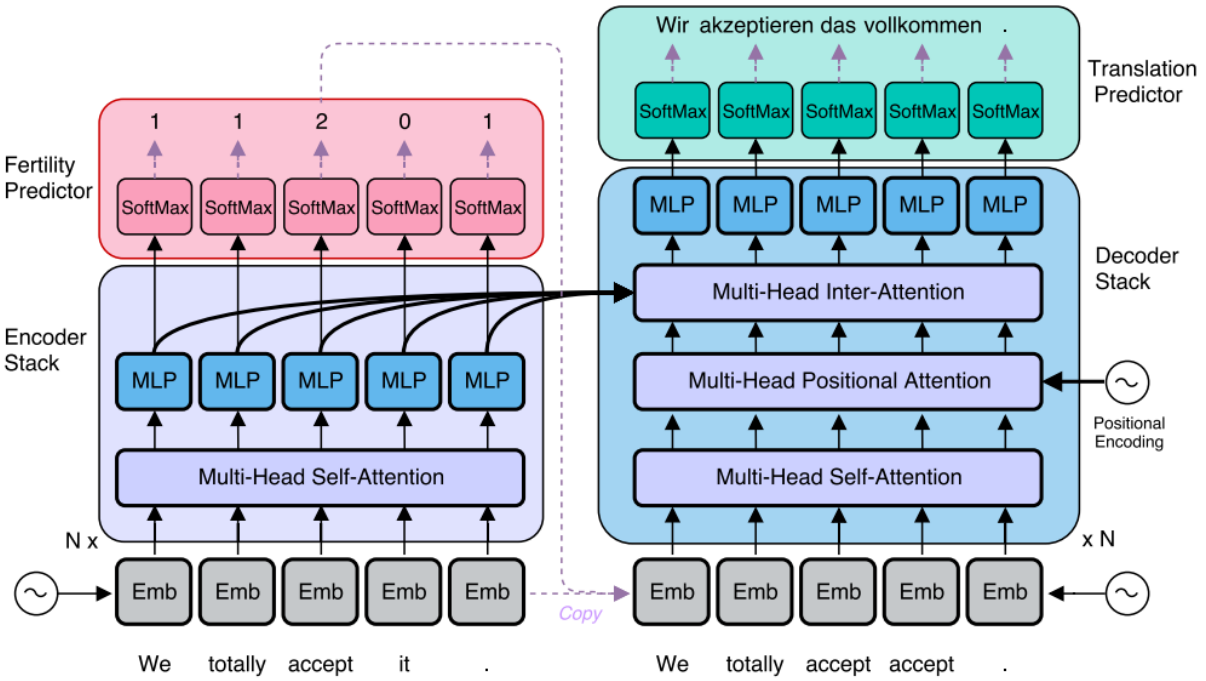
$$\begin{aligned}
& p(f_1 \dots f_m, a_1 \dots a_m | e_1 \dots e_l, m) \\
= \; & q(2|1,6,7) \times t(Le|the) \\
& \times q(3|2,6,7) \times t(Programme|program) \\
& \times q(4|3,6,7) \times t(a|has) \\
& \times q(5|4,6,7) \times t(ete|been) \\
& \times q(6|5,6,7) \times t(mis|implemented) \\
& \times q(6|6,6,7) \times t(en|implemented) \\
& \times q(6|7,6,7) \times t(application|implemented)
\end{aligned}$$

Michael Collins. Statistical Machine Translation: IBM Models 1 and 2. 2013.

The Non-autoregressive Transformer (NAT)

- **Modeling Fertility to Tackle The Multimodality Problem**

$$p_{\mathcal{NA}}(Y|X;\theta) = \sum_{f_1,\ldots,f_{T'}\in\mathcal{F}} \left(\prod_{t'=1}^{T'} p_F(f_{t'}|x_{1:T'};\theta) \cdot \prod_{t=1}^{T} p(y_t|x_1\{f_1\},..,x_{T'}\{f_{T'}\};\theta) \right)$$



Fertility prediction

- Model the fertility $p_F(f_{t'}|x_{1:T'})$ at each position independently using a one-layer neural network
- Use a softmax classifier $(F = 50 \text{ in our experiments})$ on top of the output of the last encoder layer
- Fertility values are a property of each input word depend on information & context from the entire sentence

The Non-autoregressive Transformer (NAT)

- **Benefits of fertility as a latent variable**
  - An external aligner provides a simple and fast approximate inference model that effectively reduces the unsupervised training problem to two supervised ones
  - Using fertilities as a latent variable: Solve the multimodality problem by providing a natural factorization of the output space
    - Restricting the output distribution to those target sentences consistent with a particular fertility sequence dramatically reduces the mode space
    - The global choice of mode is factored into a set of local mode choices: namely, how to translate each input word
    - Local mode choices can be effectively supervised because the fertilities provide a fixed "scaffold"
  - Including both fertilities and reordering in the latent variable would provide complete alignment statistics
    - Decoding function trivially easy to approximate given the latent variable and force all of the modeling complexity into the encoder
    - Using fertilities alone allows the decoder to take some of this burden off of the encoder

The Non-autoregressive Transformer (NAT)

- **Benefits of fertility as a latent variable**
  - There is no need to have a separate means of explicitly modeling the length of the translation, which is simply the sum of fertilities
    - Provide a powerful way to condition the decoding process
    - Allow the model to generate diverse translations by sampling over the fertility space

Translation Predictor And The Decoding Process

- **Benefits of fertility as a latent variable**

$$p_{\mathcal{NA}}(Y|X;\theta) = \sum_{f_1,\ldots,f_{T'} \in \mathcal{F}} \left(\prod_{t'=1}^{T'} p_F(f_{t'}|x_{1:T'};\theta) \cdot \prod_{t=1}^{T} p(y_t|x_1\{f_1\},..,x_{T'}\{f_{T'}\};\theta) \right)$$

  - At inference time, the model can identify the translation with the highest conditional probability by marginalizing over all possible latent fertility sequences
  - Identifying the optimal translation only requires independently maximizing the local probability for each output position

Translation Predictor And The Decoding Process

- **Benefits of fertility as a latent variable**

$$Y = G(x_{1:T'}, f_{1:T'}; \theta)$$

  - Represent the optimal translation given a source sentence and a sequence of fertility values
  - But searching and marginalizing over the whole fertility space is still intractable
  - Propose three heuristic decoding algorithms to reduce the search space of the NAT model

Translation Predictor And The Decoding Process

- **Heuristic decoding algorithms to reduce the search space of the NAT model**

$$\hat{Y}_{\text{argmax}} = G(x_{1:T'}, \hat{f}_{1:T'}; \theta) \text{ where } \hat{f}_{t'} = \underset{f}{\text{argmax}}\, p_F(f_{t'}|x_{1:T'}; \theta)$$

Argmax decoding

- The fertility sequence is modeled with a conditionally independent factorization
- So simply estimate the best translation by choosing the highest-probability fertility for each input word

$$\hat{Y}_{\text{average}} = G(x_{1:T'}, \hat{f}_{1:T'}; \theta) \text{ where } \hat{f}_{t'} = \text{Round}\left(\sum_{f_{t'}=1}^{L} p_F(f_{t'}|x_{1:T'}; \theta) f_{t'}\right)$$

Average decoding

- Estimate each fertility as the expectation of its corresponding softmax distribution

Translation Predictor And The Decoding Process

- **Heuristic decoding algorithms to reduce the search space of the NAT model**

The autoregressive teacher to identify the best overall translation

$$\hat{Y}_{\text{NPD}} = G(x_{1:T'}, \underset{f_{t'} \sim p_F}{\operatorname{argmax}}\, p_{\mathcal{AR}}(G(x_{1:T'}, f_{1:T'}; \theta)|X; \theta); \theta)$$

Noisy parallel decoding (NPD)

- A more accurate approximation of the true optimum of the target distribution
  - Inspired by Cho (2016), draw samples from the fertility space and compute the best translation for each fertility sequence
- A scoring function: it can run as fast as it does at train time because it can be provided with all decoder inputs in parallel
- A stochastic search method
- Increase the computational resources required linearly by the sample size
  - The process only doubles the latency compared to computing a single translation if sufficient parallelism is available

Kyunghyun Cho. Noisy parallel approximate decoding for conditional recurrent language model. arXiv preprint arXiv:1605.03835, 2016.

Training

- **A variational bound for the overall maximum likelihood loss**

$$\mathcal{L}_{\mathrm{ML}} = \log p_{\mathcal{NA}}(Y|X;\theta) = \log \sum_{f_{1:T'} \in \mathcal{F}} p_F(f_{1:T'}|x_{1:T'};\theta) \cdot p(y_{1:T}|x_{1:T'}, f_{1:T'};\theta)$$

$$\geq \mathop{\mathbb{E}}_{f_{1:T'} \sim q} \left(\underbrace{\sum_{t=1}^{T} \log p(y_t|x_1\{f_1\}, .., x_{T'}\{f_{T'}\};\theta)}_{\text{Translation Loss}} + \underbrace{\sum_{t'=1}^{T'} \log p_F(f_{t'}|x_{1:T'};\theta)}_{\text{Fertility Loss}} \right) + \mathcal{H}(q)$$

- A discrete sequential latent variable
- Conditional posterior distribution
- Approximate using a proposal distribution

$f_{1:T'}$

$q(f_{1:T'}|x_{1:T'}, y_{1:T})$

$p(f_{1:T'}|x_{1:T'}, y_{1:T};\theta)$

Training

- **A variational bound for the overall maximum likelihood loss**

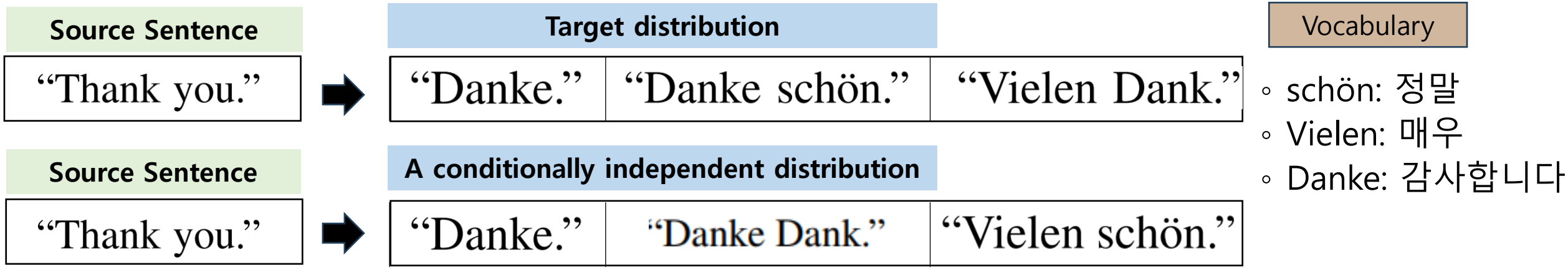
$$\mathcal{L}_{\mathbf{ML}} \geq \mathop{\mathbb{E}}_{f_{1:T'} \sim q} \left(\underbrace{\sum_{t=1}^{T} \log p(y_t | x_1\{f_1\}, .., x_{T'}\{f_{T'}\}; \theta)}_{\text{Translation Loss}} + \underbrace{\sum_{t'=1}^{T'} \log p_F(f_{t'} | x_{1:T'}; \theta)}_{\text{Fertility Loss}} \right) + \mathcal{H}(q)$$

  - Simplifies the inference process considerably, as the expectation over q is deterministic
    - Choose a proposal distribution q defined by a separate, fixed fertility model
    - Possible options include the output of an external aligner, which produces a deterministic sequence of integer fertilities in our fixed autoregressive teacher model
  - Allow us to train the entire model in a supervised fashion, using the inferred fertilities to simultaneously train the translation model p and supervise the fertility neural network model p_F

Training

- **Sequence-level Knowledge Distillation**
  - Improves the ability of the non-autoregressive output distribution to approximate the multimodal target distribution
    - It does not completely solve the problem of non-determinism in the training data
    - There are multiple correct translations consistent with a single sequence of fertilities

.g., fertility sequence $[2, 0, 1]$, because "you" is not directly translated in German sentence



Training

- **Sequence-level Knowledge Distillation**
  - Previous work: sequence-level knowledge distillation (Kim & Rush, 2016)
    - Apply the distillation to construct a new corpus by training an autoregressive machine translation model, known as the teacher, on an existing training corpus
    - Use that model's greedy outputs as the targets for training the non-autoregressive student
    - The resulting targets are less noisy and more deterministic, as the trained model will consistently translate a sentence like "Thank you." into the same German translation every time
    - On the other hand, they are also lower in quality than the original dataset

Training

- **Fine-Tuning**
  - Some drawbacks from a decomposition of the overall maximum likelihood loss into translation and fertility terms
    - Heavily relies on the deterministic, approximate inference model provided by the external alignment system
  - It would be desirable to train entire model, including the fertility predictor, end to end

Additional loss term consisting of the reverse K-L divergence with the teacher model (word-level knowledge distillation)

$$\mathcal{L}_{\mathrm{RKL}}\left(f_{1:T'};\theta\right)=\sum_{t=1}^{T}\sum_{y_t}\left[\log p_{\mathcal{AR}}\left(y_t|\hat{y}_{1:t-1},x_{1:T'}\right)\cdot p_{\mathcal{NA}}\left(y_t|x_{1:T'},f_{1:T'};\theta\right)\right] \qquad \hat{y}_{1:T}=G(x_{1:T'},f_{1:T'};\theta)$$

$$\mathcal{L}_{\mathrm{FT}}=\lambda\left(\underbrace{\mathop{\mathbb{E}}_{f_{1:T'}\sim p_F}\left(\mathcal{L}_{\mathrm{RKL}}\left(f_{1:T'}\right)-\mathcal{L}_{\mathrm{RKL}}\left(\bar{f}_{1:T'}\right)\right)}_{\mathcal{L}_{\mathrm{RL}}}+\underbrace{\mathop{\mathbb{E}}_{f_{1:T'}\sim q}\left(\mathcal{L}_{\mathrm{RKL}}\left(f_{1:T'}\right)\right)}_{\mathcal{L}_{\mathrm{BP}}}\right)+\left(1-\lambda\right)\mathcal{L}_{\mathrm{KD}}$$

  - Such a loss is more favorable towards highly peaked student output distributions than a standard cross-entropy error would be

Training

- **Fine-Tuning**

Additional loss term consisting of the reverse K-L divergence with the teacher model (word-level knowledge distillation)

$$\mathcal{L}_{\text{RKL}}\left(f_{1:T'};\theta\right)=\sum_{t=1}^{T}\sum_{y_t}\left[\log p_{\mathcal{AR}}\left(y_t|\hat{y}_{1:t-1},x_{1:T'}\right)\cdot p_{\mathcal{NA}}\left(y_t|x_{1:T'},f_{1:T'};\theta\right)\right] \qquad \hat{y}_{1:T}=G(x_{1:T'},f_{1:T'};\theta)$$

$$\mathcal{L}_{\text{FT}}=\lambda\left(\underbrace{\mathop{\mathbb{E}}_{f_{1:T'}\sim p_F}\left(\mathcal{L}_{\text{RKL}}\left(f_{1:T'}\right)-\mathcal{L}_{\text{RKL}}\left(\bar{f}_{1:T'}\right)\right)}_{\mathcal{L}_{\text{RL}}}+\underbrace{\mathop{\mathbb{E}}_{f_{1:T'}\sim q}\left(\mathcal{L}_{\text{RKL}}\left(f_{1:T'}\right)\right)}_{\mathcal{L}_{\text{BP}}}\right)+\left(1-\lambda\right)\mathcal{L}_{\text{KD}}$$

$$\bar{f}_{1:T'}=\hat{Y}_{\text{average}}=G(x_{1:T'},\hat{f}_{1:T'};\theta)\ \text{where}\ \hat{f}_{t'}=\text{Round}\left(\sum_{f_{t'}=1}^{L}p_F(f_{t'}|x_{1:T'};\theta)f_{t'}\right)$$

- Train the whole model jointly with a weighted sum of the original distillation loss and two such terms, one an expectation over the predicted fertility distribution, normalized with a baseline, and the other based on the external fertility inference model

Experiment

- **Datasets**
  - IWSLT16 En–De2
    - Use separate English and German vocabulary and embeddings
  - WMT14 En–De,3 and WMT16 En–
    - Use shared BPE vocabulary and additionally share encoder and decoder word embeddings

- **Teacher**
  - Sequence-level knowledge distillation is applied to alleviate multimodality in the training dataset, using autoregressive models as the teacher
  - The same teacher model used for distillation is also used as a scoring function for fine-tuning and noisy parallel decodig
  - Transformer architecture
    - Use the same sizes and hyperparameters for each student and its respective teacher, with the exception of the newly added positional self-attention and fertility prediction module

Experiment

- **Preparation for knowledge distillation**
  - Train all teacher models using maximum likelihood
  - Freeze their parameters
  - To avoid the redundancy of running fixed teacher models repeatedly on the same data, we decode the entire training set once using each teacher to create a new training dataset for its respective student
- **Encoder initialization**
  - Initialize the weights in student's & teacher's encoder to share the same encoder input
- **Fertility supervision during training**
  - Supervise the fertility predictions at train time by using a fixed aligner as a fertility inference function
- **Evaluation metrics**
  - Tokenized and cased BLEU scores

Part 7.

Experiment

- **Results**
  - The NAT performs between 2-5 BLEU points worse than its autoregressive teacher
  - NPD improves the performance of our non-autoregressive model
  - shows a speedup of more than a factor of 10 over greedy autoregressive decoding
  - Latencies for decoding with NPD, regardless of sample size, could be reduced to about 80ms by parallelizing across multiple GPUs because

BLEU score & Latency

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

Latency is computed as the time to decode a single sentence without minibatching, averaged over the whole test set

Experiment

- **Ablation Study**
  - Fails to train when provided with only positional embeddings as input to decoder
  - Training on the distillation corpus provides a fairly consistent improvement
  - Switching from uniform copying of source inputs to fertility-based copying improves performance

Distillation		Decoder Inputs			Fine-tuning			BLEU	BLEU (T)
b=1	b=4	+uniform	+fertility	+PosAtt	+$\mathcal{L}_{\text{KD}}$	+$\mathcal{L}_{\text{BP}}$	+$\mathcal{L}_{\text{RL}}$		
				✓				≈ 2	
		✓		✓				16.51	
			✓	✓				18.87	
✓		✓		✓				20.72	
	✓	✓		✓				21.12	
✓			✓					24.02	43.91
✓			✓	✓				25.20	45.41
✓		✓		✓	✓	✓		22.44	
✓			✓	✓			✓	×	×
✓			✓	✓		✓		×	×
✓			✓	✓	✓	✓		25.76	46.11
✓			✓	✓	✓	✓	✓	**26.52**	**47.38**

An X indicates that fine-tuning caused that model to get worse

Experiment

- **Ablation Study**
  - Fine-tuning doesn't converge with reinforcement learning alone, or with $\mathcal{L}_{\mathrm{BP}}$
  - Use of all three fine-tuning terms together leads to an improvement
  - Training the student model from a distillation corpus produced using beam search is similar to training from the greedily-distilled corpus
  - The translations produced by the NAT with NPD are also noticeably more literal.
    - Instances of repeated words or phrases, highlighted in gray, are most prevalent in the non-autoregressive output for the relatively complex first example sentence
    - A pair in the second, are not present in the versions with noisy parallel decoding
    - NPD scoring using the teacher model can filter out such mistakes

Source:	politicians try to pick words and use words to shape reality and control reality , but in fact , reality changes words far more than words can ever change reality .
Target:	Politiker versuchen Worte zu benutzen , um die Realität zu formen und die Realität zu kontrollieren , aber tatsächlich verändert die Realität Worte viel mehr , als Worte die Realität jemals verändern könnten .
AR:	Politiker versuchen Wörter zu wählen und Wörter zur Realität zu gestalten und Realität zu steuern , aber in Wirklichkeit verändert sich die Realität viel mehr als Worte , die die Realität verändern können .
NAT:	Politiker versuchen , Wörter wählen und zu verwenden , um Realität zu formen und Realität zu formen , aber tatsächlich ändert Realität Realität viel mehr als Worte die Realität Realität verändern .
NAT+NPD:	Politiker versuchen , Wörter wählen und zu verwenden , um Realität Realität formen und die Realität zu formen , aber tatsächlich ändert die Realität Worte viel mehr als Worte jemals die Realität verändern können .

Experiment

- **Ablation Study**
  - Noisy parallel decoding process demonstrates the diversity of translations that can be found by sampling from the fertility space
    - Left: Sampled fertility sequences from the encoder, represented with their corresponding decoder input sequences
    - Right: Values for the latent variable leads to a different possible output translation
    - Transformer picks the best translation (red), a process which is faster than directly using it to generate output

se lucreaza la solutii de genul acesta .	
se la solutii de genul acesta .	solutions on this kind are done .
se lucreaza la solutii de acesta .	work done on solutions like this .
se lucreaza solutii de genul acesta .	solutions on this kind is done .
se se lucreaza la solutii de acesta .	work is done on solutions like this .
se lucreaza lucreaza la solutii de acesta .	work is done on solutions like this .
se se lucreaza lucreaza la solutii de acesta .	**work is being done on solutions like this .**
se se lucreaza lucreaza la solutii de de acesta .	work is being done on solutions such as this .
se se lucreaza lucreaza la solutii de genul acesta .	work is being done on solutions such this kind .

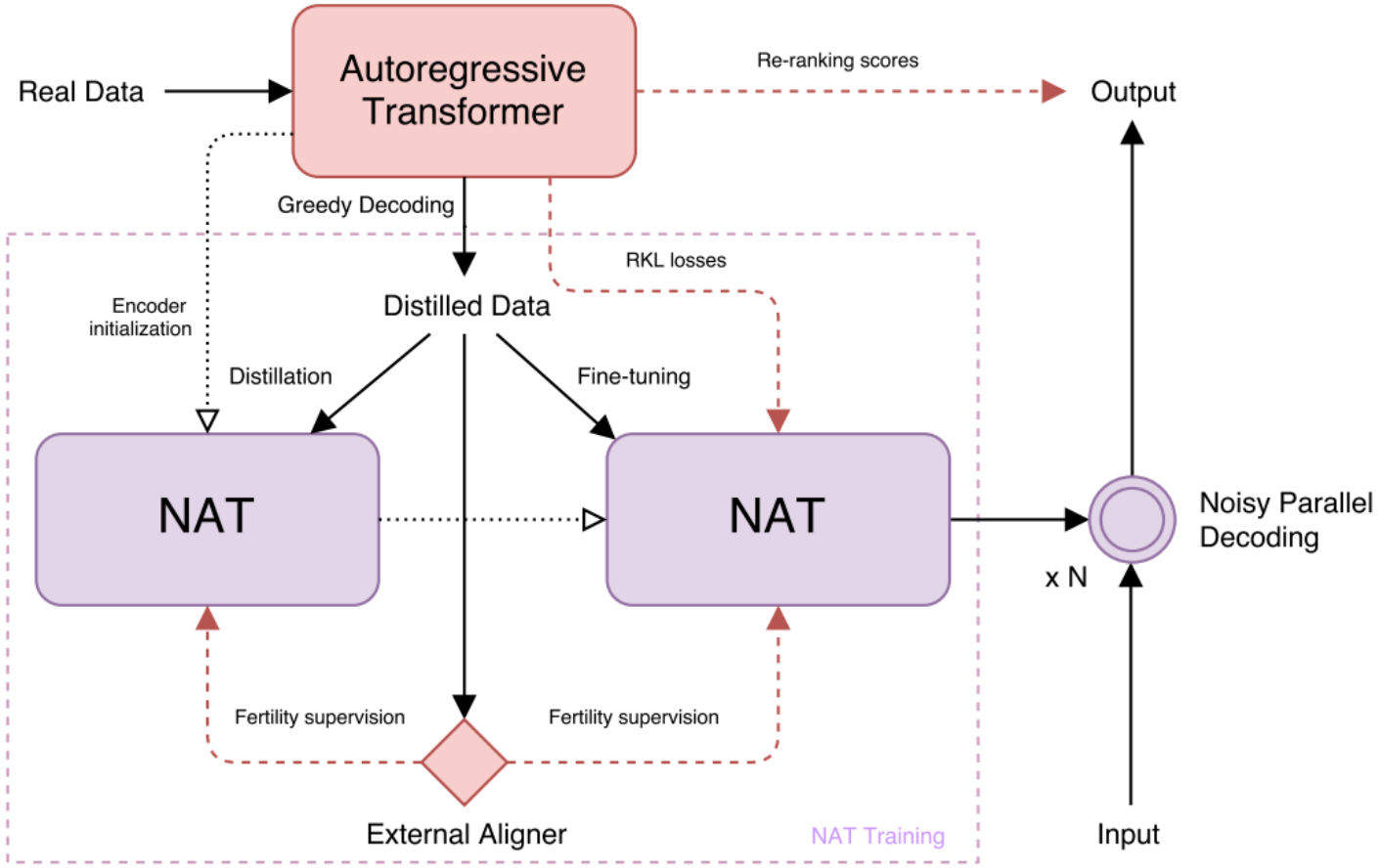
Part 8.

Conclusion

- **A latent variable model for non-autoregressive machine translation**
  - Enables a decoder based on transformer architecture
    - To take full advantage of its exceptional degree of internal parallelism even at inference time
  - Latency
    - Measure translation latencies of one-tenth that of an equal-sized autoregressive model
  - Performance
    - Maintain competitive BLEU scores

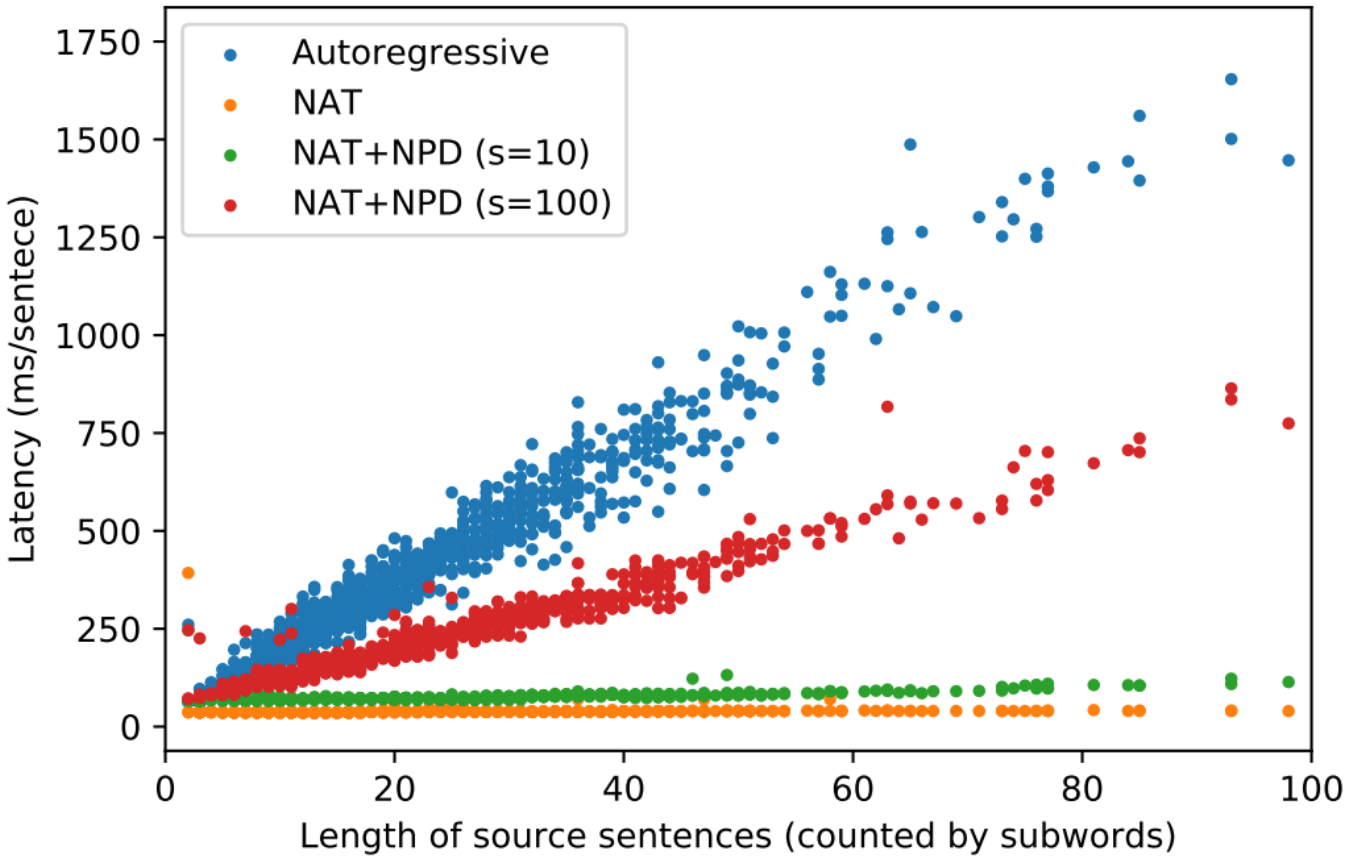
Appendix

- **The schematic structure of training and inference for the NAT**
  - The "distilled data" contains target sentences decoded by the autoregressive model and ground-truth source sentences



Appendix

- **The translation latency for a single sentence**
  - When using NPD with sample size 100, the level of parallelism is enough to more than saturate the GPU, leading again to linear latencies



Appendix

- **Learning curves**
  - Learning curves for training and fine-tuning of the NAT on IWSLT. BLEU scores are on the development set