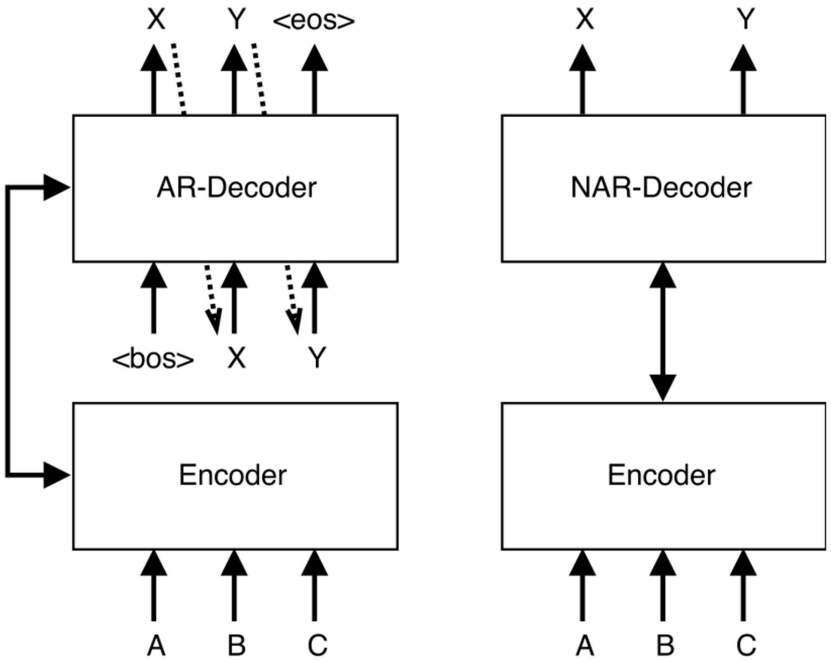
Неавторегрессивный перевод

Розаева Мария, 06.12.2024

План

- Разберемся, что такое авторегрессивный и неавторегрессивный перевод
- Вспомним необходимые мелочи: трансформеры, BLEU
- Разберем три архитектуры
  1. The Non-Autoregressive Transformer (2018)
  2. Mask-Predict (2019)
  3. Glancing Transformer (2021)
- Поговорим об области в целом

Пример: AR (autoregressive) vs NAR (non-autoregressive)



Зачем это вообще нужно?

Авторегрессивный перевод хорошо решает поставленную задачу, но

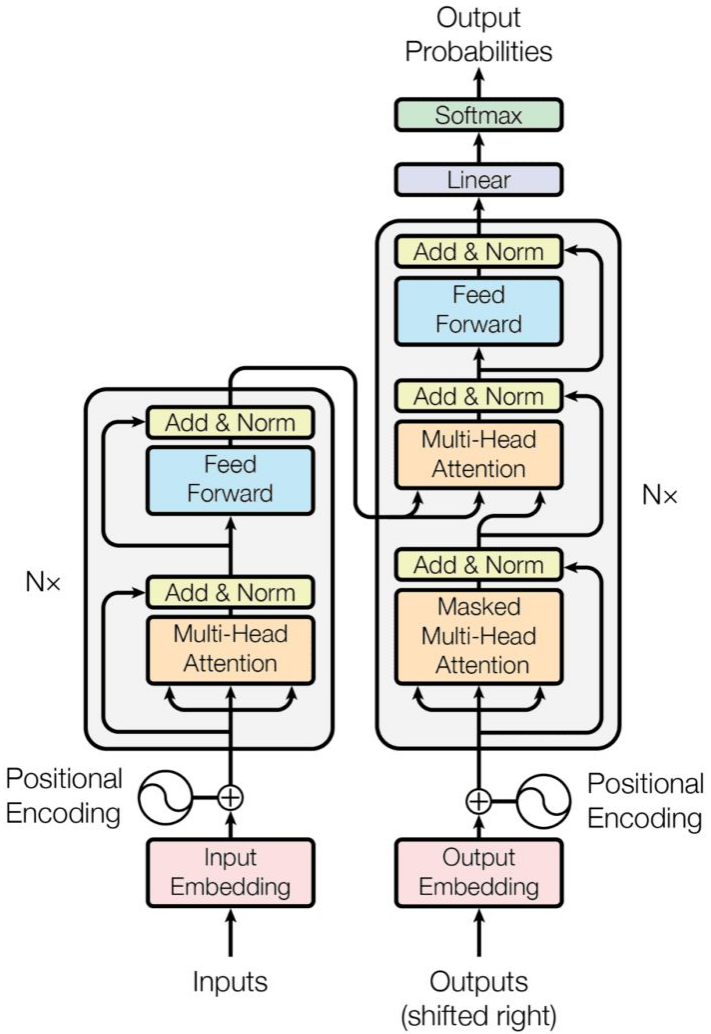
Его нельзя распараллелить

* здесь и далее говорим про декодеры

Напоминание про трансформеры и авторегрессивный перевод

Главное: есть энкодер и декодер

Тоже главное: обычно генерируем последовательность токен за токеном



BLEU – метрика качества перевода

- Разработана в 2001 году
- Основана на подсчете слов (unigrams) и словосочетаний (n-grams)
- Быстрая
- Плохая

Наивная идея о неавторегрессивном декодере

$$\text{source sentence } X = \{x_1, ..., x_{T'}\}$$
$$\text{output sentence } Y = \{y_1, ..., y_T\}$$

Авторегрессивный подход

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

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

Неавторегрессивный подход

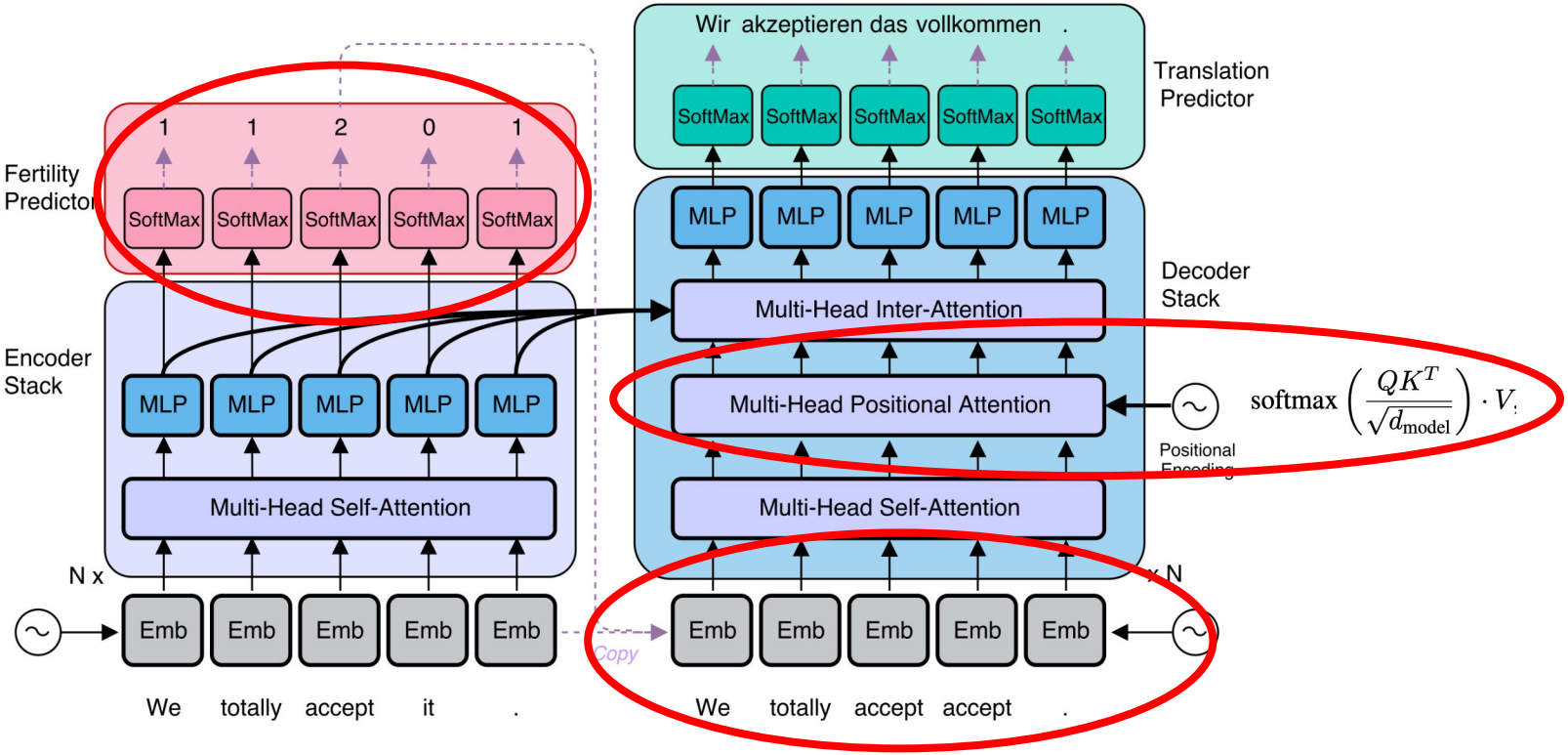
$$p_{\mathcal{NA}}(Y|X;\theta) = p_L(T|x_{1:T'};\theta) \cdot \prod_{t=1}^{T} p(y_t|x_{1:T'};\theta)$$

Глобальные проблемы

- Не понятно, как улавливать связь между словами
- Как предсказывать длину выходной последовательности
- Как решить две проблемы выше, добившись хорошей скорости
- И при этом хорошего качества

1/3. The Non-Autoregressive Transformer

NAT. Архитектура

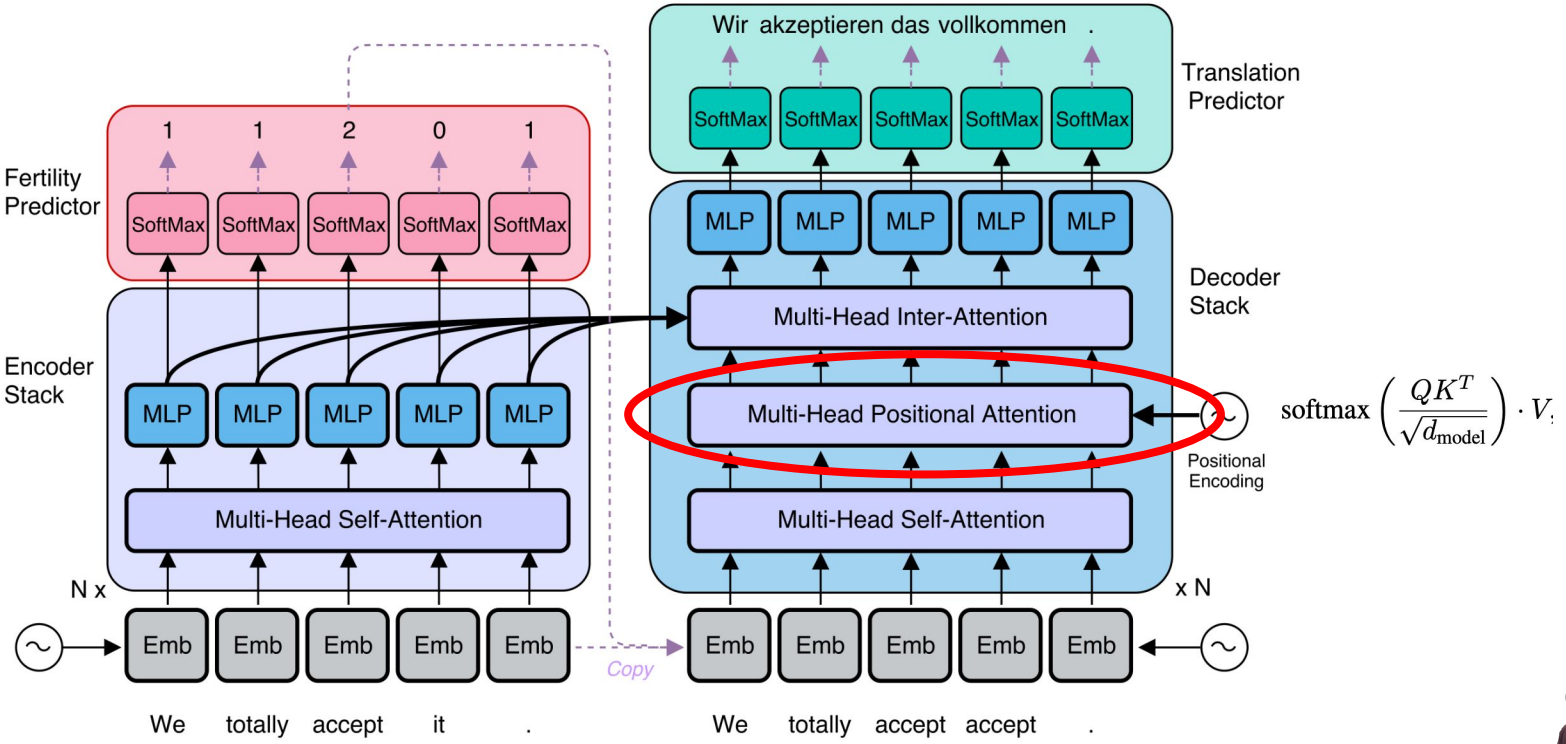


NAT. Fertilities (независимость и длина выхода)

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

where $\mathcal{F} = \{f_1, ..., f_{T'} | \sum_{t'=1}^{T'} f_{t'} = T, f_{t'} \in \mathbb{Z}^*\}$ is the set of all fertility sequences

NAT. Positional Attention



$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_{\text{model}}}}\right)\cdot V$$

NAT. Обучение и инференс

NAT. Метрики (BLEU)

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; decoding is implemented in PyTorch on a single NVIDIA Tesla P100

При обучении использовалось Knowledge Distilation

FT – Fine Tuning

NPD (Noisy parallel decoding) – механизм выбора лучшего перевода с использованием авторегрессивного учителя

s – количество семплов, из которых выбираем

NAT. Проблема

Делаем предположение, что токены в выходной последовательности независимы друг от друга

Из-за этого не можем учитывать связи между словами в выходной последовательности

2/3. Mask-Predict

Mask-Predict. Идея

	[the] вывод [the] французских войск был [on] 20(ого) ноября завершен .
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 .

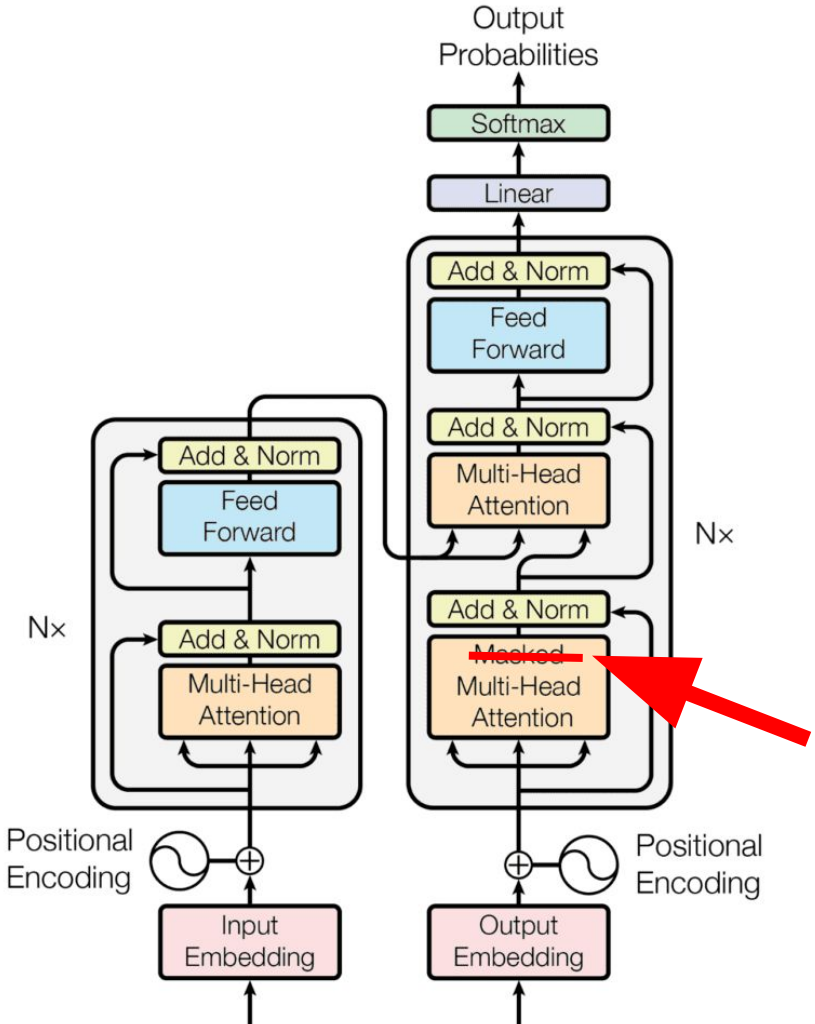
Y_masked

Y_observed

Mask-Predict. Архитектура

Параметры те же самые, что и у обычного трансформера

Отличие: не используем маску в декодере



Mask-Predict. Conditional Masked Language Model

source text: *X*
target text: *Y*

пусть длину выходной последовательности N нам пока что дали свыше

Y_observed – доступные токены
Y_masked – закрытые токены, которые нужно предсказать

Предполагаем, что все *y* в *Y_masked* независимы друг от друга

Предсказываем: *P(y | X, Y_observed)* для каждого *y* из *Y_masked*

Неявно добавляем условие на длину последовательности *N*, так как знаем, что *N* = |*Y_observed*| + |*Y_masked*|

Mask-Predict. Conditional Masked Language Model

[the] вывод [the] французских войск был [on] 20(ого) ноября завершен .

X	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 .

Y_masked

Y_observed

$$y_i^{(t)} = \arg\max_w P(y_i = w | X, Y_{obs}^{(t)})$$

$$p_i^{(t)} = \max_w P(y_i = w | X, Y_{obs}^{(t)})$$

$$y_i^{(t)} = y_i^{(t-1)}$$

$$p_i^{(t)} = p_i^{(t-1)}$$

Mask-Predict. Длина последовательности

Добавляем специальный токен <LENGTH> к исходному предложению, и предсказываем по нему длину

Лайфхак

Выбираем топ ℓ наиболее вероятных длин последовательности, и одновременно считаем несколько переводов

Выбираем перевод с лучшим средним значением логарифмов вероятностей

Mask-Predict. Метрики (BLEU)

Model	Dimensions (Model/Hidden)	Iterations	WMT'14 EN-DE	WMT'14 DE-EN	WMT'16 EN-RO	WMT'16 RO-EN
NAT w/ Fertility (Gu et al., 2018)	512/512	1	19.17	23.20	29.79	31.44
CTC Loss (Libovický and Helcl, 2018)	512/4096	1	17.68	19.80	19.93	24.71
Iterative Refinement (Lee et al., 2018)	512/512	1	13.91	16.77	24.45	25.73
	512/512	10	21.61	25.48	29.32	30.19
(Dynamic #Iterations)	512/512	?	21.54	25.43	29.66	30.30
Small CMLM with Mask-Predict	512/512	1	15.06	19.26	20.12	20.36
	512/512	4	**24.17**	**28.55**	**30.00**	30.43
	512/512	10	**25.51**	**29.47**	**31.65**	**32.27**
Base CMLM with Mask-Predict	512/2048	1	18.05	21.83	27.32	28.20
	512/2048	4	**25.94**	**29.90**	**32.53**	**33.23**
	512/2048	10	**27.03**	**30.53**	**33.08**	**33.31**
Base Transformer (Vaswani et al., 2017)	512/2048	N	27.30	— —	— —	— —
Base Transformer (Our Implementation)	512/2048	N	27.74	31.09	34.28	33.99
Base Transformer (+Distillation)	512/2048	N	27.86	31.07	— —	— —
Large Transformer (Vaswani et al., 2017)	1024/4096	N	28.40	— —	— —	— —
Large Transformer (Our Implementation)	1024/4096	N	28.60	31.71	— —	— —

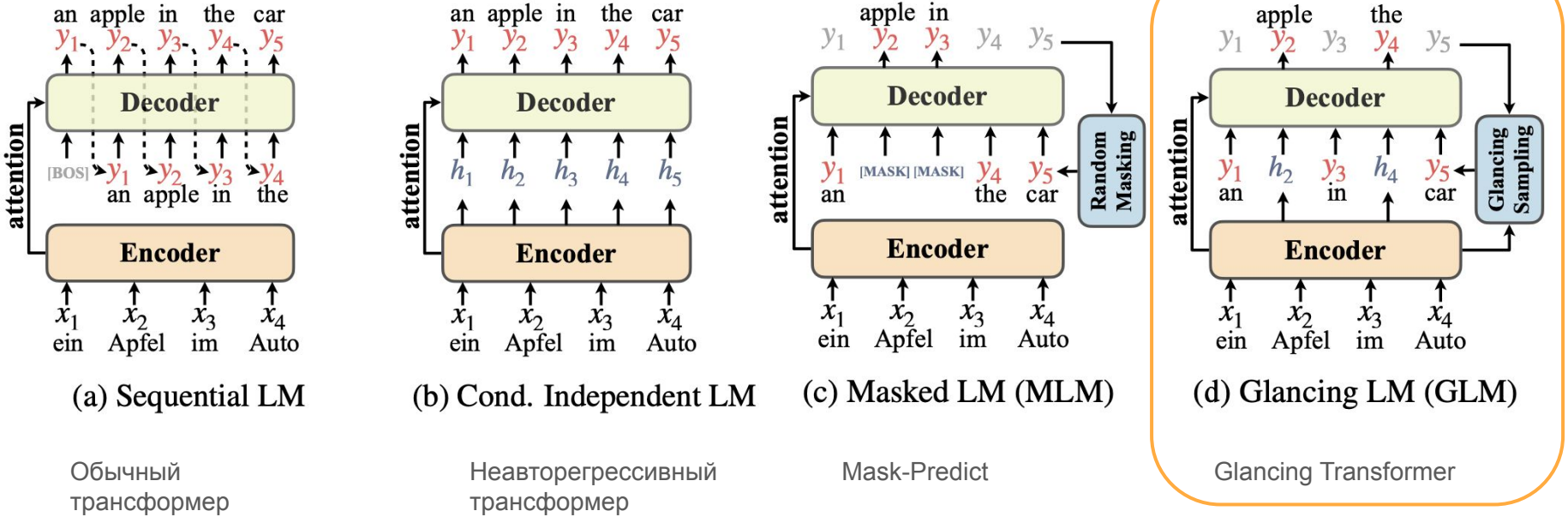
* здесь также использовали Knowledge distillation

Mask-Predict. Проблема

Для получения хорошего качества может потребоваться много итераций

3/3. Glancing Transformer

GLAT. Отличие от рассмотренных архитектур



Обычный трансформер

Неавторегрессивный трансформер

Mask-Predict

Glancing Transformer

GLAT. Glancing Language Model (GLM)

$$X = \{x_1, x_2, ..., x_N\}$$

$$Y = \{y_1, y_2, ..., y_T\}$$

изначально предсказанные токены

Хотим максимизировать

$$\mathcal{L}_{\text{GLM}} = \sum_{y_t \in \mathbb{GS}(Y, \hat{Y})} \log p(y_t | \mathbb{GS}(Y, \hat{Y}), X; \theta)$$

подмножество токенов, выбранных с помощью “glancing sampling”

$$\mathbb{GS}(Y, \hat{Y}) = \text{Random}(Y, S(Y, \hat{Y}))$$

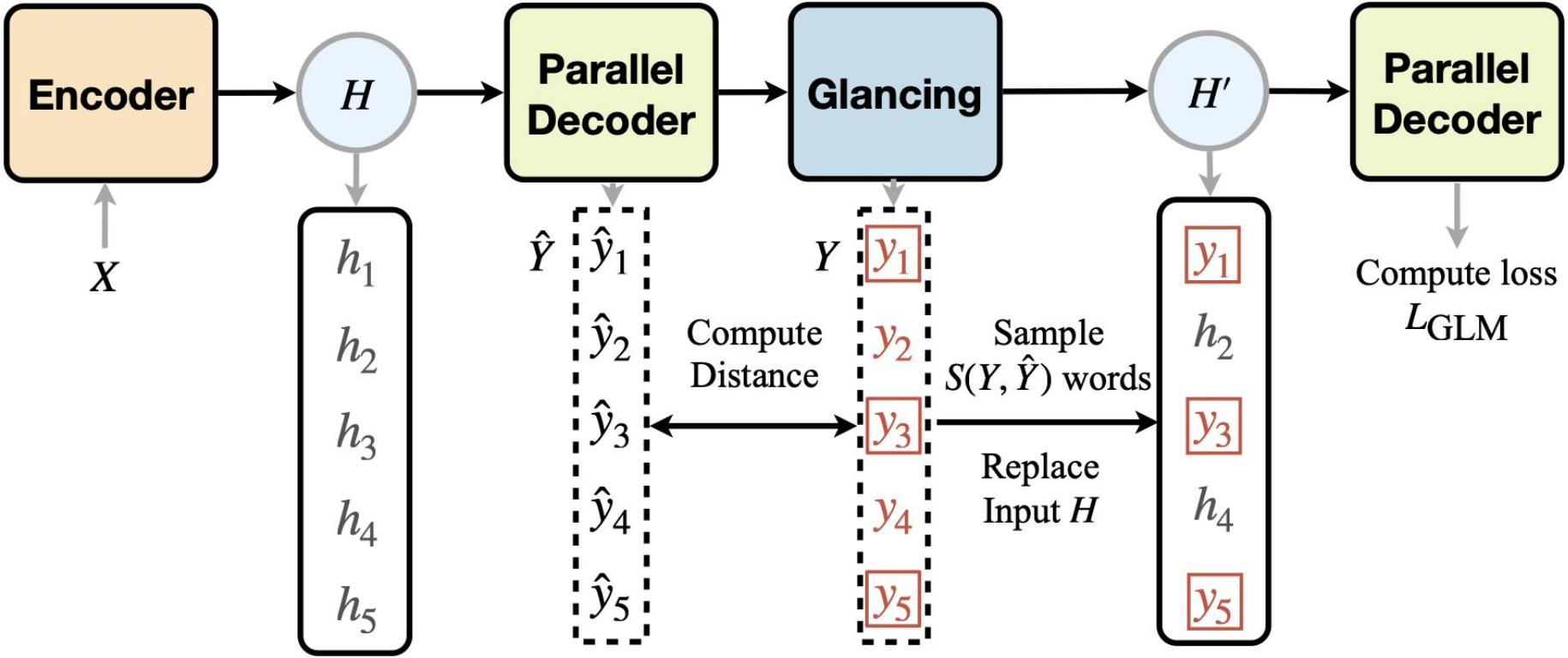
$$S(Y, \hat{Y}) = \lambda \cdot d(Y, \hat{Y})$$

lambda – гиперпараметр, d – расстояние

$$d(Y, \hat{Y}) = \sum_{t=1}^{T} (y_t \neq \hat{y}_t)$$

в случае неравенства длин используются другие формулы

GLAT. Glancing sampling (обучение)



GLAT. Длина последовательности

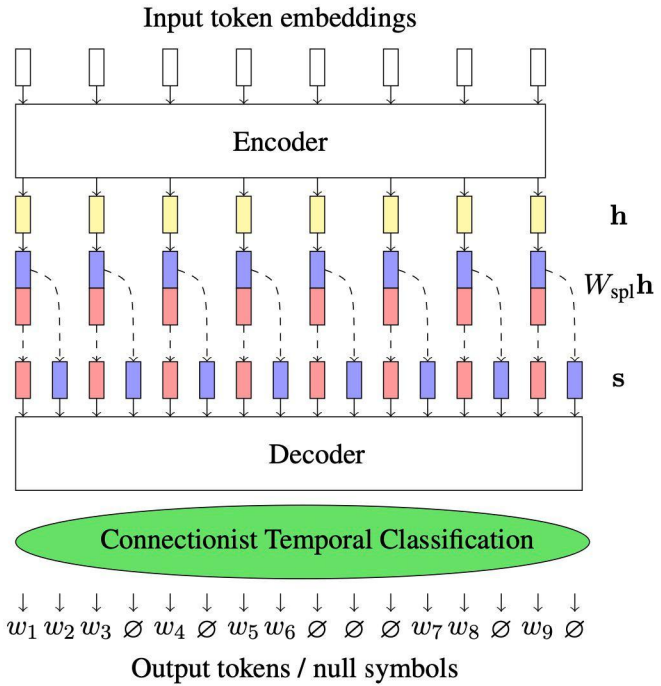
- Специальный <LENGTH> токен

или

- CTC (на картинке)

или

- 1. Выбираем m кандидатов в длины
  2. Генерируем по ним лучше переводы
  3. Отдельным трансформером выбираем лучший перевод



GLAT. Метрики (BLEU)

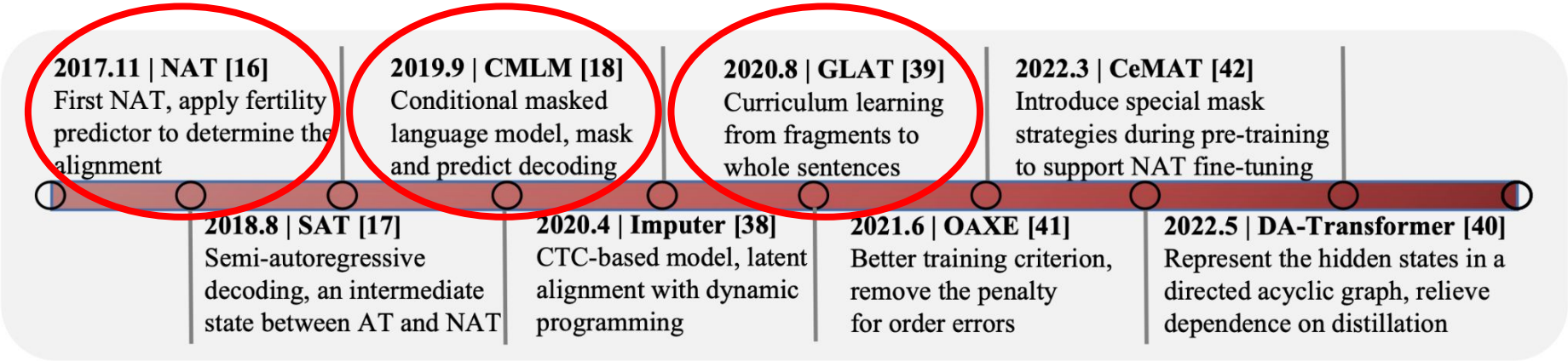
		Models	I_{dec}	WMT14 EN-DE	WMT14 DE-EN	WMT16 EN-RO	WMT16 RO-EN	Speed Up
AT Models		Transformer (Vaswani et al., 2017)	T	27.30	/	/	/	/
		Transformer (ours)	T	27.48	31.27	33.70	34.05	1.0×
Fully NAT		NAT-FT (Gu et al., 2018)	1	17.69	21.47	27.29	29.06	15.6×
		Mask-Predict (Ghazvininejad et al., 2019)	1	18.05	21.83	27.32	28.20	/
		imit-NAT (Wei et al., 2019)	1	22.44	25.67	28.61	28.90	18.6×
		NAT-HINT (Li et al., 2019)	1	21.11	25.24	/	/	/
		Flowseq (Ma et al., 2019)	1	23.72	28.39	29.73	30.72	1.1×
		NAT-DCRF (Sun et al., 2019)	1	23.44	27.22	/	/	10.4 ×
	w/ CTC	NAT-CTC (Libovický and Helcl, 2018)	1	16.56	18.64	19.54	24.67	/
	w/ CTC	Imputer (Saharia et al., 2020)	1	25.80	28.40	32.30	31.70	18.6×
	w/ NPD	NAT-FT + NPD (m=100)	1	19.17	23.20	29.79	31.44	2.4×
	w/ NPD	imit-NAT + NPD (m=7)	1	24.15	27.28	31.45	31.81	9.7×
	w/ NPD	NAT-HINT + NPD (m=9)	1	25.20	29.52	/	/	/
	w/ NPD	Flowseq + NPD (m=30)	1	25.31	30.68	32.20	32.84	/
	w/ NPD	NAT-DCRF + NPD (m=9)	1	26.07	29.68	/	/	6.1×
	Ours	NAT-base*	1	20.36	24.81	28.47	29.43	15.3×
	Ours	CTC*	1	25.52	28.73	32.60	33.46	14.6 ×
	Ours	GLAT	1	25.21	29.84	31.19	32.04	15.3×
	Ours	GLAT + CTC	1	26.39	29.54	32.79	**33.84**	14.6 ×
	Ours	GLAT + NPD (m=7)	1	**26.55**	**31.02**	**32.87**	33.51	7.9×

* здесь также использовали Knowledge distillation

GLAT. Проблема

Авторегрессивные трансформеры все равно переводят лучше :(

Краткая история неавторегрессивного перевода



Проблемы

- **Очень** зависят от объема данных, их качества и выбранной стратегии обучения
- Часто необходимо использовать множество манипуляции с данными и/или предобученные модели (например knowledge distilation)
- Есть подозрения, что различия в соотношении скорость/качество могут быть не такими значительными
- BLEU – не лучшая метрика для оценки неавторегрессивного (да и в целом) перевода из-за его необычных ошибок
- …

Приложения в других областях

Неавторегрессивные подходы иногда используются в:

- Text Generation
- Semantic Parsing (“извлечение смысла”)
- Text to Speech
- Diffusion Models
- …

Итог

Пока авторегрессивные трансформеры лучше справляются с задачей перевода

Но неавторегрессивные:

- продолжают развиваться
- полезны, когда скорость важнее качества
- идеи используются в приложениях, не связанных с переводом

Источники

- Non-Autoregressive Transformer
- Mask-Predict
- Glancing Transformer
- Неплохой обзор направления: A Survey on Non-Autoregressive Generation for Neural Machine Translation and Beyond
- Почему BLEU так себе метрика: To Ship or Not to Ship: An Extensive Evaluation of Automatic Metrics for Machine Translation
- Про уменьшение скорости при увеличении батчей: Non-Autoregressive Machine Translation: It's Not as Fast as it Seems

Для интересующихся: страница с множеством статей на тему неавторегрессивных приложений в разных областях