On Discrepancy between Density Estimation and Sequence Generation

BLEU vs. LL

Титизян Армине, 172, 11.11.2020

В чем проблема?

BLEU vs. Log-likelihood

- Многие **seq2seq** задачи формулируются как нахождение лучшей оценки распределения $p(y \mid x)$ для выхода y по входу x
- Учим модель максимизировать **лог-правдоподобие,** качество модели оцениваем по **лог-правдоподобию** на тестовой выборке
- Цель решения задачи: выдать $y*$, которая бы давала лучшие значения для метрики R (для каждой задачи своя, у перевода **BLEU**) в сравнении с *золотым стандартом* $\hat{y}$, то есть лучшие значения $R(\hat{y}, y* \mid x)$
- Нет гарантии, что оптимизация по лог-правдоподобию даст оптимизацию R?

Гарантии есть!

Но с огромными оговорками

Эксперименты

- Авторы обучили для 5 языковых пар (WMT’14 En↔De, WMT’16 En↔Ro, IWSLT’16 De→En) следующие виды моделей:

(1) Авторегрессионые модели

(2) Модели со скрытыми переменными с не-авторегрессионным декодером и гауссовским прайером (**diagonal Gaussian prior**)

(3) Модели со скрытыми переменными с не-авторегрессионным декодером и гибким прайером (**normalizing flow**)

(4) Перечисленные выше модели с дистилляцией (**Knowledge distillation**)

Модели

Авторегрессионные модели

Обучение

- Факторизуется совместное распределение как произведение условных
- $\log p_{\mathrm{AR}}(\mathbf{y} \mid \mathbf{x}) = \sum_{t=1}^{T} \log p_\theta \left(y_t \mid y_{<t}, \mathbf{x}\right)$
- Максимизируем $L_{\mathrm{AR}}(\theta) = \frac{1}{N} \sum_{n=1}^{N} \log p_{\mathrm{AR}} \left(\mathbf{y}_n \mid \mathbf{x}_n\right)$

Авторегрессионые модели

Вывод и типы используемых архитектур

- Вывод является проблемой поиска:

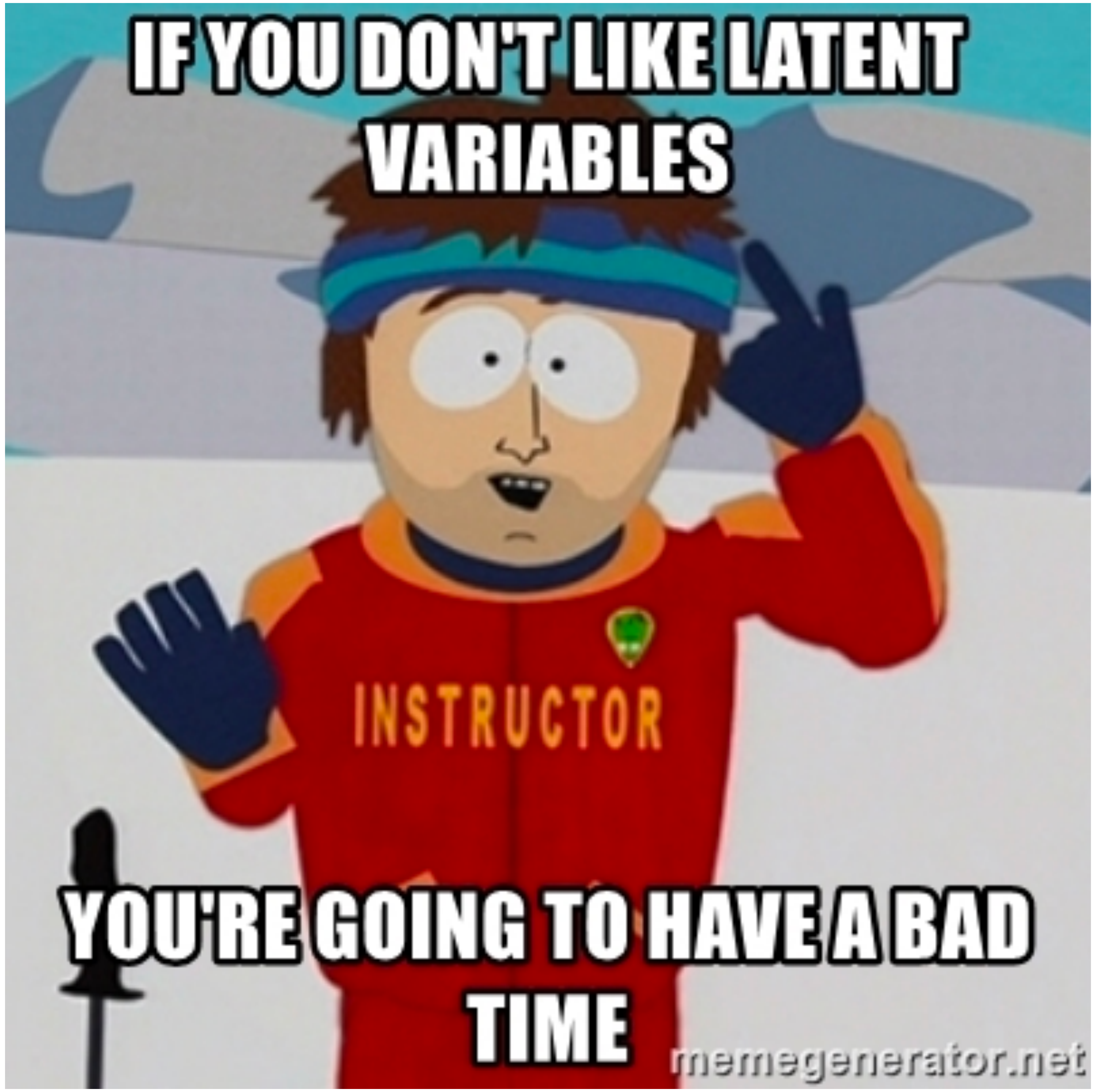
$$\operatorname{argmax}_{\mathbf{y}} \log p_\theta(\mathbf{y} \mid \mathbf{x}) = \operatorname{argmax}_{y_{1:T}} \sum_{t=1}^{T} \log p_\theta\left(y_t \mid y_{<t}, \mathbf{x}\right),$$

где θ — параметры модели

- С ростом T область поиска растет экспоненциально, поэтому данную задачу решают с помощью жадного (**greedy**) или лучевого (**beam**) поиска

- Возможные архитектуры: рекуррентные нейронные сети (**RNN**), сверточные сети (**CNN**) или **Transformer** с **self-attention** (последние два возможны в силу того, что новая информация $y_{\geqslant t}$ не используется для предсказания текущего y_t)

Модели со скрытыми переменными



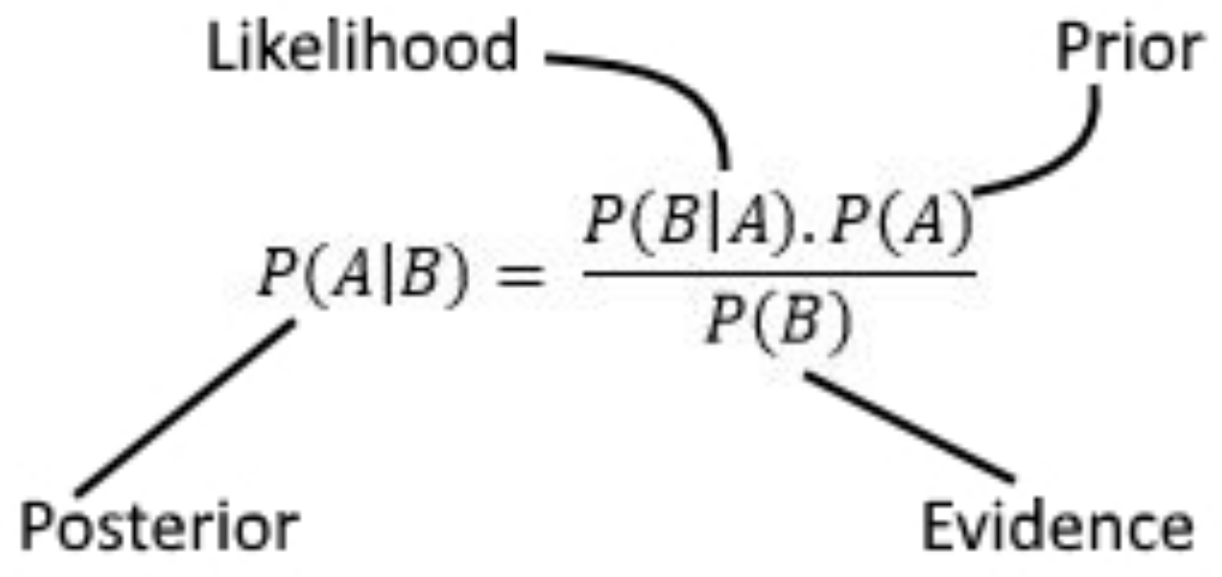
Модели со скрытыми переменными (*LVM*)

Обучение

- Маргинальное правдоподобие по $\mathbf{z}$

$$\log p_{\mathrm{LVM}}(\mathbf{y} \mid \mathbf{x}) = \log \int_{\mathbf{z}} p_\theta(\mathbf{y} \mid \mathbf{z}, \mathbf{x}) p_\theta(\mathbf{z} \mid \mathbf{x}) dz$$

- $p(\mathbf{z} \mid \mathbf{x}, \mathbf{y}) = \dfrac{p(\mathbf{y} \mid \mathbf{z}, \mathbf{x}) p(\mathbf{z} \mid \mathbf{x})}{\int_{\mathbf{z}} p(\mathbf{y} \mid \mathbf{z}, \mathbf{x}) p(\mathbf{z} \mid \mathbf{x}) dz}$



Напоминалка: Теорема Байеса

Модели со скрытыми переменными (*LVM*)

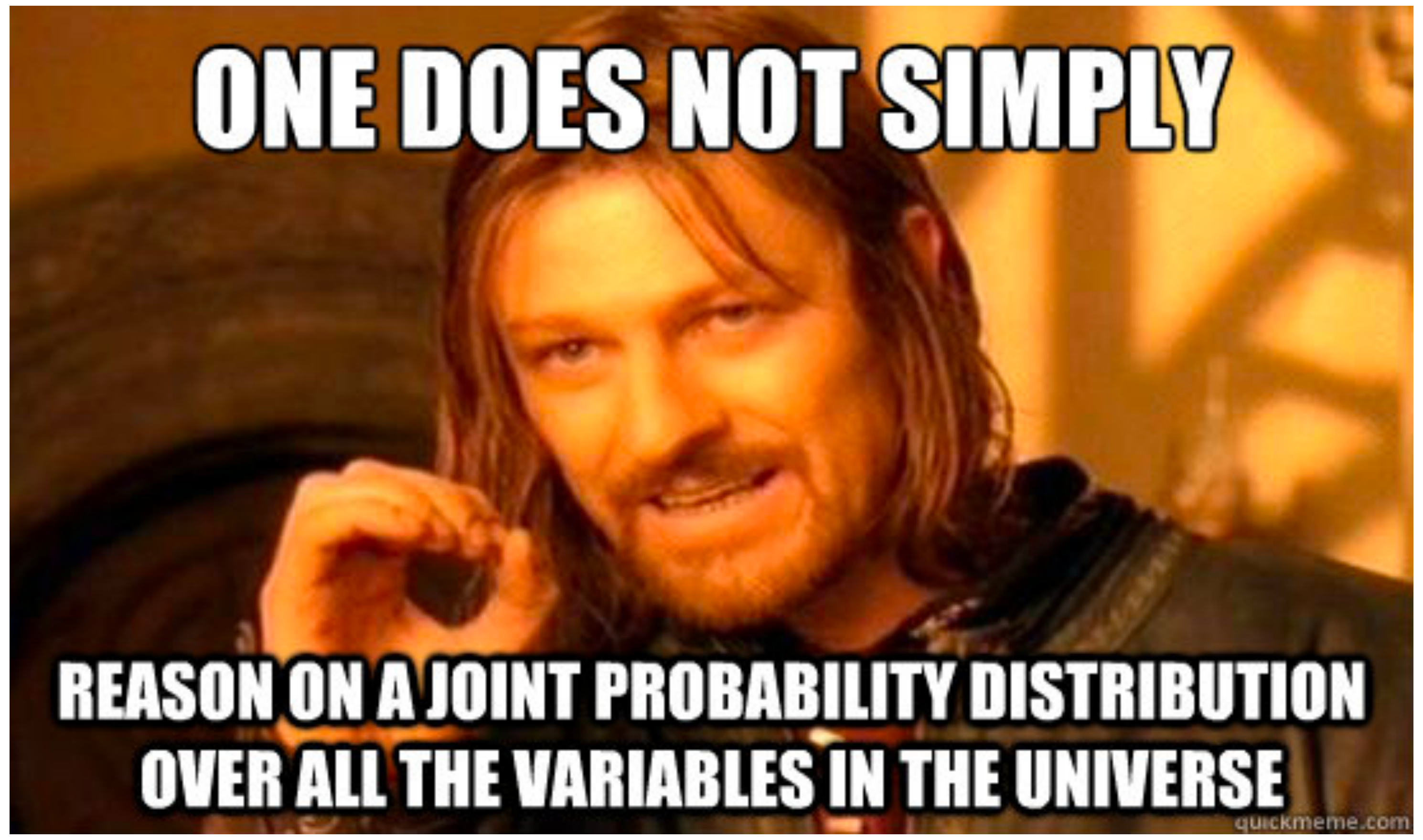
Обучение

- Т.к. аналитически посчитать интеграл и сделать полный байесовский вывод абсолютно не возможно, воспользуемся вариационным байесовским выводом.
- Будем использовать параметризированное распределение $q_\varphi(\mathbf{z} \mid \mathbf{x}, \mathbf{y})$
- Максимизируем **E**vidence **L**ower **BO**und (**ELBO**):
$L_{\mathrm{LVM}}(\theta, \varphi) = \frac{1}{N} \sum_{n=1}^{N} \mathrm{ELBO}(\mathbf{y}, \mathbf{x}; \theta, \varphi)$

$$\log p_{\mathrm{LVM}}(\mathbf{y} \mid \mathbf{x}) \geq \mathrm{ELBO}(\mathbf{y}, \mathbf{x}; \theta, \phi) = \mathop{\mathbb{E}}_{\mathbf{z} \sim q_\phi} \left[\log p_\theta(\mathbf{y} \mid \mathbf{z}, \mathbf{x})\right] - \mathrm{KL}\left[q_\phi(\mathbf{z} \mid \mathbf{y}, \mathbf{x}) \| p_\theta(\mathbf{z} \mid \mathbf{x})\right]$$

Декодер

Prior



Модели со скрытыми переменными (*LVM*)

- Т.к LVM умеет выявлять зависимости между переменными, то декодирующее распределение можно факторизовать

$$p_\theta(\mathbf{y} \mid \mathbf{z}, \mathbf{x}) = \prod_{t=1}^{T} p_\theta(y_t \mid \mathbf{z}, \mathbf{x})$$

- Так же можно факторизовать

$$q_\phi(\mathbf{z}_{1:T} \mid \mathbf{y}, \mathbf{x}) = \prod_{t=1}^{T} \mathcal{N}\left(z_t \mid \mu_{\phi,t}(\mathbf{y}, \mathbf{x}), \sigma_{\phi,t}(\mathbf{y}, \mathbf{x})\right)$$

Модели со скрытыми переменными (*LVM*)

Вывод

- Оптимизируем **ELBO** относительно выхода $\operatorname{argmax}_{\mathbf{y}} \mathrm{ELBO}(\mathbf{y}, \mathbf{x}; \theta, \varphi)$.
- Посчитать опять невозможно
- Используем Delta Posterior (https://arxiv.org/pdf/1908.07181.pdf)
- $\delta(\mathbf{z} \mid \mu) = \begin{cases} 1, & \text{if } \mathbf{z} = \mu \\ 0, \text{ otherwise} & \end{cases}$
- $\mathop{\mathbb{E}}_{\mathbf{z} \sim \delta(\mathbf{z} \mid \boldsymbol{\mu})} \left[p_\theta(\mathbf{y} \mid \mathbf{z}, \mathbf{x}) + p_\theta(\mathbf{z} \mid \mathbf{x}) \right] + \overbrace{\mathscr{H}(\delta)}^{=0} = \log p_\theta(\mathbf{y} \mid \boldsymbol{\mu}, \mathbf{x}) + \log p_\theta(\boldsymbol{\mu} \mid \mathbf{x})$
- В бой вступает EM-алгоритм:
    1. $\mu = \mathbb{E}_{q_\phi}(\mathbf{z})$
    2. Максимизируем proxy ELBO: $\hat{\mathbf{y}} = \operatorname{argmax}_{\mathbf{y}}(\log p_\theta(\mathbf{y} \mid \boldsymbol{\mu}, \mathbf{x}))$
- Таким образом вывод абсолютно детерминированный, что очень желаемое свойство для задач генерации

Priors

Diagonal Gaussian vs. Normalizing Flow

$$\log p_\theta(z_{1:T} \mid x) = \sum_{t=1}^{T} \log \mathcal{N}\left(z_t \mid \mu_{\theta,t}(x), \sigma_{\theta,t}(x)\right),$$

где каждая скрытая переменная z_t взята из нормального распределения со средним и дисперсией, полученных во время обучения модели.

Нам понадобятся:

1. Базовое распределение $p_b(\epsilon)$ (чаще всего стандартное нормальное)
2. Обратимое преобразование $f : f(\mathbf{z}) = \epsilon$ и его обратное $f^{-1} : f^{-1}(\epsilon) = \mathbf{z}$. Так как prior — условное распределение с условием x, то $f(\mathbf{z}, x) = \epsilon$ и $f^{-1}(\epsilon, x) = \mathbf{z}$

Тогда имеем:

$$\log p_\theta(\mathbf{z} \mid \mathbf{x}) = \log p_b(f(\mathbf{z}; \mathbf{x})) + \log \left| \det \frac{\partial f(\mathbf{z}; \mathbf{x})}{\partial \mathbf{z}} \right|$$

Priors

Оптимизация Normalizing flow

- Каждое преобразование будем строить последовательно так, чтобы только на некотором подмножестве было использовано аффинное преобразование, что делает вычисление якобиана более эффективным:

$$
\begin{aligned}
\mathbf{z}_{\mathrm{id}}, \mathbf{z}_{\mathrm{tr}} &= \operatorname{split}(\mathbf{z}) \\
\mathbf{s}, \mathbf{b} &= g_{\mathrm{param}}(\mathbf{z}_{\mathrm{id}}) \\
f(\mathbf{z}) &= \operatorname{concat}(\mathbf{z}_{\mathrm{id}};\ \ \mathbf{s} \cdot \mathbf{z}_{\mathrm{tr}} + \mathbf{b}),
\end{aligned}
$$

g_{param} — может быть сколь угодно сложным, так как нет ограничения на обратимость

Используемые архитектуры

- Авторегрессионые: Transformer-big (**Tr-L**), Transformer-base (**Tr-B**) (оригинальные гиперпараметры статьи), Transformer-small (**Tr-S**): 2 головы attention, по 5 слоев encoder и decoder
- Скрытые переменные:
  - sentence encoder (Transformer encoder),
  - length predictor (2 слойный перцептрон),
  - prior (Transformer)
    - ✳ Gauss: на вход — последовательность позиционных кодировок длины T, выход — значения среднего и стандартного отклонения для каждого токена. Gauss-base -> **Ga-B**, Gauss-large -> **Ga-L**
    - ✳ Flow: g_{param} — Transformer decoder + Linear with weight-normalization, сама модель трехуровневая, на каждом уровне половина переменных моделируется стандартным нормальным распределением. Сплит с помощью 1x1 multi-head conv.(https://arxiv.org/pdf/1909.02480.pdf). Flow-base -> **Fl-B,** Flow-small -> **Fl-S**
  - decoder(Transformer decoder, на вход подаются исходные скрытые состояния из encoder)
  - posterior (Transformer decoder + Linear with weight-normalization, на вход подаются исходные скрытые состояния из encoder);
- ***KD***: на основе **Tr-B** с beam == 4

Результаты

Корреляция

Среди моделей одного семейства

- Среди авторегрессионных моделей наблюдается идеальная корреляция лог-правдоподобия и BLEU

- Среди неавторегрессионных моделей с одинаковым prior есть сильная, но не идеальная корреляция

		BLEU (↑)		LL (↑)	
		Raw	Dist.	Raw	Dist.
WMT'14 En→De	Tr-S	24.54	24.94	-1.77	-2.36
	Tr-B	28.18	27.86	-1.44	-2.19
	Tr-L	29.39	28.29	-1.35	-2.23
	Ga-B	15.74	24.54	-1.51	-2.44
	Ga-L	17.33	**25.53**	-1.47	-2.24
	Fl-S	18.17	21.98	-1.41	-2.13
	Fl-B	18.57	21.82	**-1.23**	-2.05
	Fl-B$^{(*)}$	18.55	21.45		
	Fl-L$^{(*)}$	20.85	23.72		
WMT'14 De→En	Tr-S	29.15	28.40	-1.66	-2.24
	Tr-B	32.21	32.24	-1.42	-2.12
	Tr-L	33.16	32.24	-1.35	-2.05
	Ga-B	21.64	29.29	-1.41	-2.17
	Ga-L	23.03	**30.30**	-1.31	-2.04
	Fl-S	23.17	27.14	-1.28	-1.73
	Fl-B	23.12	26.72	**-1.20**	-1.71
	Fl-B$^{(*)}$	23.36	26.16		
	Fl-L$^{(*)}$	25.40	28.39		
WMT'16 En→Ro	Tr-S	30.12	29.57	-1.72	-1.95
	Tr-B	33.46	33.28	-1.63	-2.52
	Ga-B	28.03	29.71	-2.38	-3.48
	Ga-L	28.16	**30.91**	-2.44	-3.54
	Fl-S	26.85	28.63	-1.53	-2.42
	Fl-B	27.49	29.09	**-1.50**	-2.31
	Fl-B$^{(*)}$	29.26	29.34		
	Fl-L$^{(*)}$	29.86	29.73		
WMT'16 Ro→En	Tr-S	29.33	28.87	-1.84	-1.93
	Tr-B	32.19	31.15	-1.79	-2.28
	Ga-B	26.48	27.81	-2.41	-2.92
	Ga-L	27.35	**28.02**	-2.32	-3.01
	Fl-S	26.03	26.12	-1.65	-2.05
	Fl-B	27.14	27.33	**-1.64**	-2.01
	Fl-B$^{(*)}$	30.16	30.44		
	Fl-L$^{(*)}$	30.69	30.72		
IWSLT	Tr-S	31.54	31.72	-1.84	-2.56
	Ga-B	24.36	26.80	-1.98	-2.70
	Fl-S	23.64	26.69	-1.66	-2.28
	Fl-B	24.89	**27.00**	**-1.57**	-2.46
	Fl-B$^{(*)}$	24.75	27.75		

Корреляция

Среди моделей разных семейств

- Корреляции нет!
- Fl-B дает лучшее моделирование распределения, но при этом очень плохое качество перевода (без дистилляция)
- Ga-L с дистилляцией —сильно лучшее качество перевода перед Fl-B, хотя он дает лучшую аппроксимацию распределения

		BLEU (↑)		LL (↑)	
		RAW	DIST.	RAW	DIST.
WMT'14 EN→DE	TR-S	24.54	24.94	-1.77	-2.36
	TR-B	28.18	27.86	-1.44	-2.19
	TR-L	29.39	28.29	-1.35	-2.23
	GA-B	15.74	24.54	-1.51	-2.44
	GA-L	17.33	**25.53**	-1.47	-2.24
	FL-S	18.17	21.98	-1.41	-2.13
	FL-B	18.57	21.82	**-1.23**	-2.05
	FL-B(*)	18.55	21.45		
	FL-L(*)	20.85	23.72		
WMT'14 DE→EN	TR-S	29.15	28.40	-1.66	-2.24
	TR-B	32.21	32.24	-1.42	-2.12
	TR-L	33.16	32.24	-1.35	-2.05
	GA-B	21.64	29.29	-1.41	-2.17
	GA-L	23.03	**30.30**	-1.31	-2.04
	FL-S	23.17	27.14	-1.28	-1.73
	FL-B	23.12	26.72	**-1.20**	-1.71
	FL-B(*)	23.36	26.16		
	FL-L(*)	25.40	28.39		
WMT'16 EN→RO	TR-S	30.12	29.57	-1.72	-1.95
	TR-B	33.46	33.28	-1.63	-2.52
	GA-B	28.03	29.71	-2.38	-3.48
	GA-L	28.16	**30.91**	-2.44	-3.54
	FL-S	26.85	28.63	-1.53	-2.42
	FL-B	27.49	29.09	**-1.50**	-2.31
	FL-B(*)	29.26	29.34		
	FL-L(*)	29.86	29.73		
WMT'16 RO→EN	TR-S	29.33	28.87	-1.84	-1.93
	TR-B	32.19	31.15	-1.79	-2.28
	GA-B	26.48	27.81	-2.41	-2.92
	GA-L	27.35	**28.02**	-2.32	-3.01
	FL-S	26.03	26.12	-1.65	-2.05
	FL-B	27.14	27.33	**-1.64**	-2.01
	FL-B(*)	30.16	30.44		
	FL-L(*)	30.69	30.72		
IWSLT	TR-S	31.54	31.72	-1.84	-2.56
	GA-B	24.36	26.80	-1.98	-2.70
	FL-S	23.64	26.69	-1.66	-2.28
	FL-B	24.89	**27.00**	**-1.57**	-2.46
	FL-B(*)	24.75	27.75		

Корреляция

Пирсон

	TR-B	GA-B	FL-B
RAW	0.926	0.831	0.678
DIST.	-0.758	-0.897	-0.873

Наблюдаем отрицательную корреляцию для дистилляции

Корреляция

Out-of-Distribution

- Видим, что результаты соотносятся с in-distribution

- Flow prior модели лучше всех (даже авторегрессионных) оценивают распределение, но их качество перевода худшее

		BLEU (↑)		LL (↑)	
		RAW	DIST.	RAW	DIST.
WMT'14 →IWSLT	TR-S	29.15	28.40	-1.65	-2.25
	TR-B	32.29	31.75	-1.42	-2.12
	TR-L	<u>33.16</u>	32.24	-1.35	-2.06
	GA-B	24.26	28.77	-1.37	-2.10
	GA-L	25.46	**29.60**	-1.28	-2.01
	FL-S	24.35	26.79	-1.26	-1.76
	FL-B	24.25	27.12	**-1.19**	-1.73
IWSLT→ WMT'14	TR-S	18.50	<u>18.94</u>	-2.79	-3.41
	GA-B	12.12	13.78	-3.10	-3.83
	FL-S	11.78	**14.35**	-2.81	-3.22
	FL-B	12.56	14.30	**-2.62**	-3.43

Влияние дистилляции

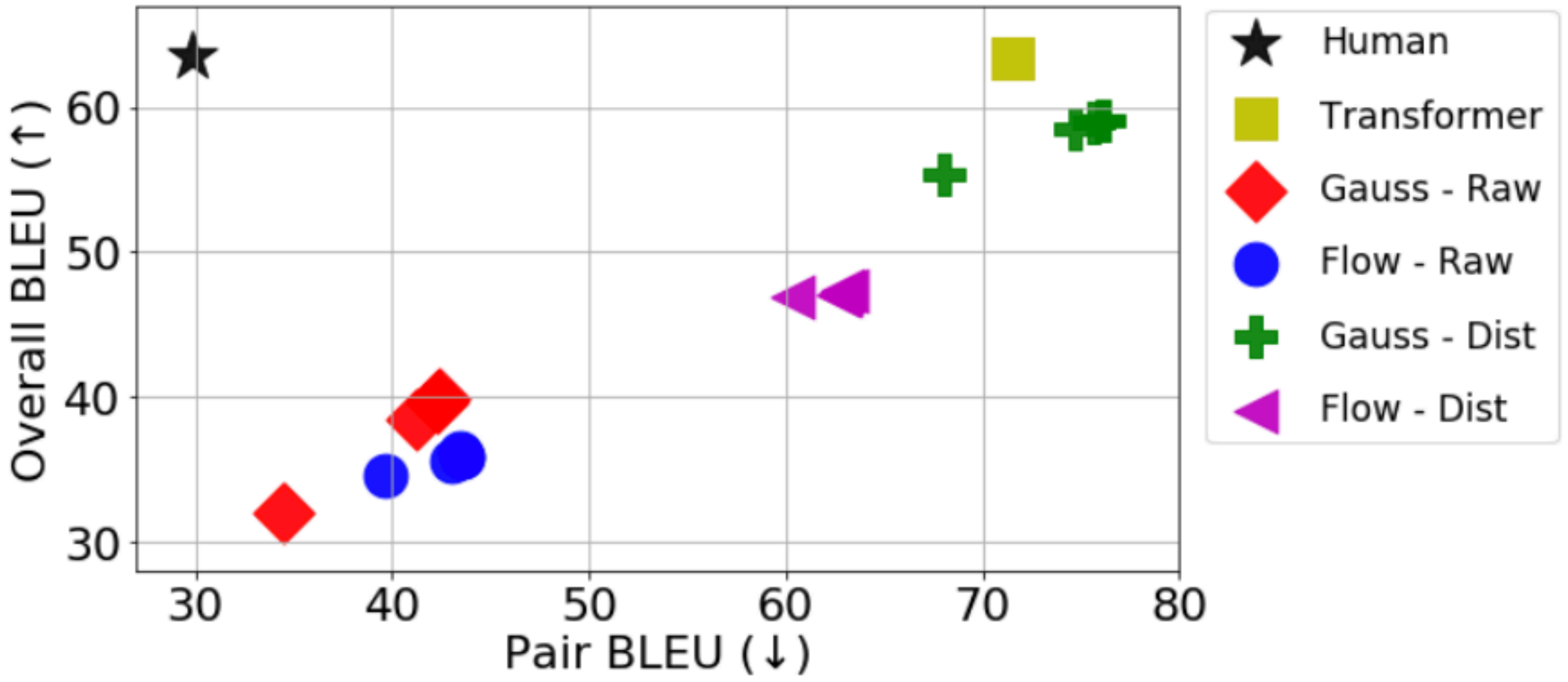
- Строго вредит качеству оценки распределения на всех датасетах
- В рамках улучшения качества перевода, стабильно улучшается качество для неавторегрессионных моделей

Сравнение моделей со скрытыми переменными

Итеративный вывод

- Улучшает для обоих моделей как качество перевода, так и качество оценки распределения, но для flow prior прирост меньше.
- Улучшение получаем взамен на ухудшение вариативности вариантов в beam
- Дистилляция улучшает BLEU (больше для Gauss), но разнообразие страдает (для Flow очень сильное ухудшение за счет очень маленького улучшения качества)

		Number of refinement steps				
		0	1	2	4	8
BLEU	GA-B	22.88	24.36	24.60	24.69	24.69
	FL-B	24.57	24.89	24.81	24.92	24.77
ELBO	GA-B	-1.11	-0.93	-0.90	-0.89	-0.89
	FL-B	-1.22	-1.17	-1.16	-1.15	-1.15



BLEU vs. Speed

- Авторегрессионные модели — медленные, хоть и дают всегда лучшее качество
- Gauss с одной итерацией вывода в 6 раз быстрее в цену 2.6 BLEU
- Декодинг для неавторегрессионных моделей производится за константное время к длине выхода за счет параллельных вычислений
- Flow prior модели хоть и огромные показывают **худшие** результаты по всем параметрам, кроме моделирования распределения

	BLEU					SPEED					SIZE
$k =$	0	1	2	4	8	0	1	2	4	8	
TR-S	24.54					2.69					17M
TR-B	28.18					2.58					60M
TR-L	29.39					1.93					208M
GA-B	23.15	24.54	24.87	24.94	24.92	28.77	20.52	16.51	12.00	8.11	75M
GA-L	24.31	25.53	25.69	25.68	25.68	19.83	14.72	10.25	7.88	4.91	95M
FL-B	21.57	21.82	21.79	21.81	21.80	5.82	5.60	4.84	3.60	3.37	75M
FL-L$^{(*)}$	23.72										258M

Вопросы

- Есть ли корреляция между log правдоподобием и BLEU? Для каких моделей?

- Выпишите формулу log правдоподобия для авторегрессионой модели.

- Какие модели дают лучшее качество перевода: авторегрессионые или не-авторегрессионые (модели со скрытыми переменными)? А какие дают лучшее моделирование распределения?