

TRANSFORMER И ЧТО ИЗ ЭТОГО ПОЛУЧИЛОСЬ

Сергей Николенко

Академия MADE — Mail.Ru

04 декабря 2021 г.

Random facts:

- 4 декабря 771 г. король франков Карломан умер в своём дворце Самусси, и Карл Великий стал единовластным правителем Франкского королевства
- 4 декабря 1783 г., через 9 дней после отплытия последнего корабля англичан, Джордж Вашингтон на ужине в таверне Френсиса (Fraunces Tavern) попрощался со своими офицерами, подал в отставку и удалился в своё поместье
- 4 декабря 1934 г. Леонид Николаев выстрелом из револьвера убил в Смольном Сергея Кирова; как это убийство было задумано и организовано, не ясно до сих пор
- 4 декабря 1956 г. на джем-сессию в Sun Studio в первый и последний раз собрался Million Dollar Quartet: Элвис Пресли, Джерри Ли Льюис, Карл Перкинс и Джонни Кэш
- 4 декабря 1971 г. во время концерта Фрэнка Заппы дотла сгорело многоэтажное казино в Монктре; это событие увековечено в песне Deep Purple «Smoke on the Water»

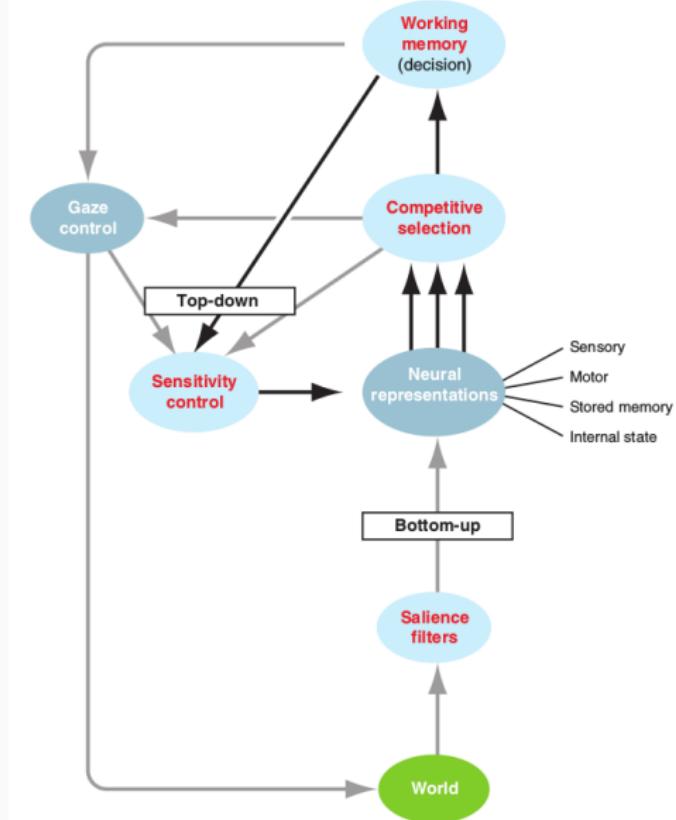
ЧТО ТАКОЕ ВНИМАНИЕ?

ВНИМАНИЕ

- Вы же сейчас внимательно меня слушаете, правильно?
- А что это значит?..
- Изображение с сетчатки тоже проходит через CNN, но потом мы часть его замечаем, а часть не особенно. Что это значит?
- Оказывается, что это довольно сложный вопрос.
- А.Р. Лурия: внимание, память и активация коры.

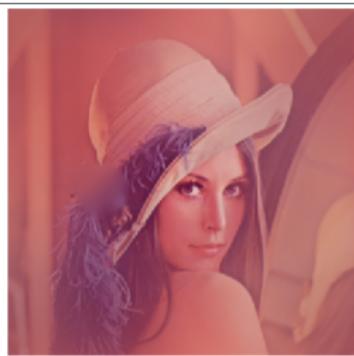
ВНИМАНИЕ

- Дело во взаимодействии с рабочей памятью (Knudsen, 2007):

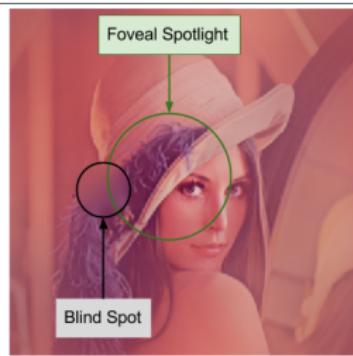


ВНИМАНИЕ

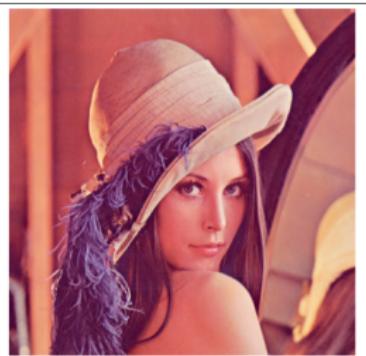
- Как это реализовать в нейронной сети? Особенно «сознательное» внимание.
- Но и «бессознательное» тоже; например, с картинками: мы же на самом деле мало чего видим в каждый момент времени.
- Центральная ямка сетчатки (fovea):



What you *actually* see



What you *actually* see (annotated)

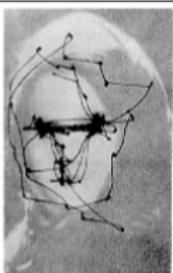


What you *think* you see

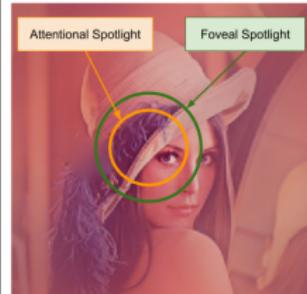
- Simons, Chabris, 1999: <https://www.youtube.com/watch?v=vJG698U2Mvo>

ВНИМАНИЕ

- Мы делаем саккады, причём это тоже не всегда помогает:



A typical saccade trace



Foveal vs Attentional Spotlights: Convergence



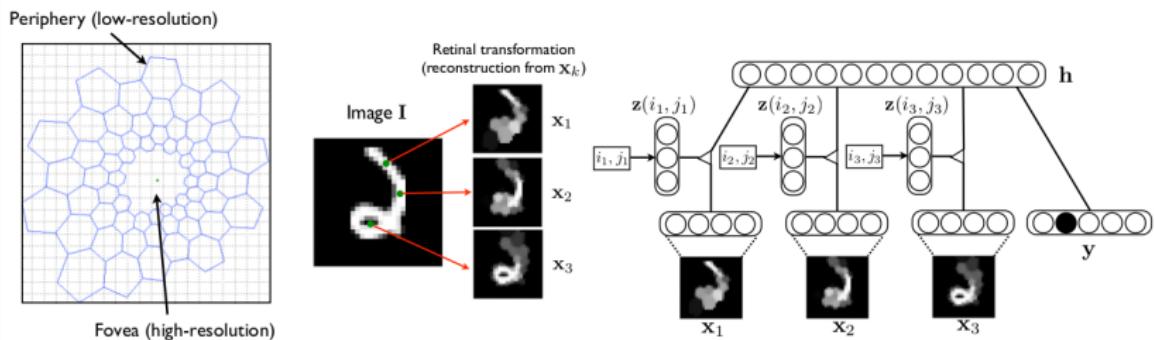
Foveal vs Attentional Spotlights: Divergence

- Или вот не совсем зрительный пример:

§ 445. Задача XIV. Лисица, преследуемая зайцем, находится от него на расстоянии 60 ея скачков; она отдаёт 9 скачков в то время, въ которое заяцъ отдаёт 6; величина же 3 скачковъ зайца равна величинѣ 7 скачковъ лисицы. Сколько скачковъ сдѣлаетъ заяцъ, чтобы догнать лисицу?

FOVEAL GLIMPSES

- В нейросети мы тоже хотим осознанно понимать, «на что» смотреть.
- Одна из первых работ (Larochelle, Hinton, 2010):

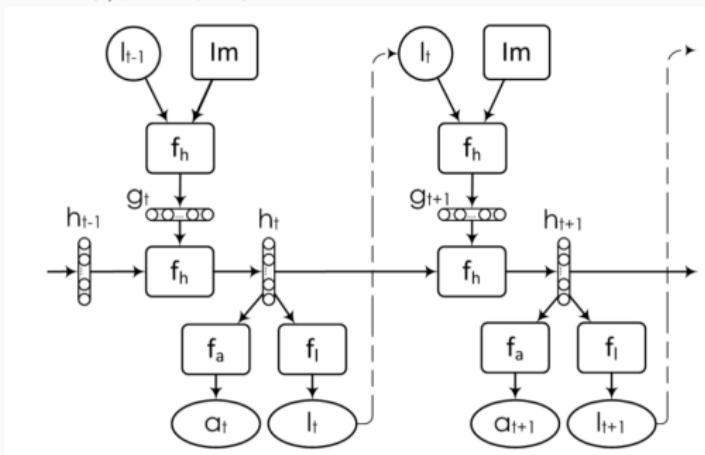


- Пытаются моделировать положения фиксаций и строить последовательность при помощи RBM.
- Последовательность – значит...

РЕКУРРЕНТНЫЕ МОДЕЛИ ЗРИТЕЛЬНОГО ВНИМАНИЯ

RECURRENT VISUAL ATTENTION

- По-настоящему всё появилось в (Mnih et al., 2014), «Recurrent Models of Visual Attention»:
 - из предыдущего \mathbf{h}_{t-1} и положения \mathcal{L}_t для нового «взгляда» f_g делает \mathbf{g}_t , вход для шага t ;
 - из \mathbf{h}_{t-1} и \mathbf{g}_t функцией f_h получается \mathbf{h}_t ;
 - из него – «действие» $a_t = f_a(\mathbf{h}_t)$ и положение следующего «взгляда» $\mathcal{L}_{t+1} = f_l(\mathbf{h}_t)$.



RECURRENT VISUAL ATTENTION

- Давайте разберёмся в модели формально:

$$\mathbf{g}_t = f_g(\mathbf{x}_t, \mathbf{l}_{t-1}; \theta_g),$$

$$\mathbf{h}_t = f_h(\mathbf{h}_{t-1}, \mathbf{g}_t; \theta_h),$$

$$\mathbf{l}_t \sim p(\cdot \mid f_l(\mathbf{h}_t; \theta_l)),$$

$$a_t \sim p(\cdot \mid f_a(\mathbf{h}_t; \theta_a)).$$

- После очередного действия получается новое наблюдение \mathbf{x}_{t+1} и награда r_t , которая будет скорее всего в конце, после всех шагов, за правильную классификацию.
- Что это напоминает?..

RECURRENT VISUAL ATTENTION

- ...о да, это reinforcement learning!
- Выучить надо стохастическую стратегию $\pi((\mathbf{l}_t, a_t) \mid \mathbf{s}_{1:t}; \theta)$, которая по истории будет выдавать следующее действие.
- У нас π задаётся через RNN, а оптимизировать надо

$$J(\theta) = \mathbb{E}_{p(\mathbf{s}_{1:T}; \theta)} [R] = \mathbb{E}_{p(\mathbf{s}_{1:T}; \theta)} \left[\sum_{t=1}^T r_t \right].$$

- Выглядит очень сложно – ожидание по последовательностям действий, т.е. по пространству большой размерности.
- Но есть методы – давайте сделаем preview тому, что потом будет в reinforcement learning.

RECURRENT VISUAL ATTENTION

- (Williams, 1992): алгоритм REINFORCE, в котором доказывается и используется выборочная оценка этого ожидания. Давайте выведем, а потом применим к нашему случаю...
- Нам надо оптимизировать

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=1}^T r_t(s_t, a_t) \right] \approx \frac{1}{M} \sum_{i=1}^M \sum_{t=1}^T r(s_t^{(i)}, a_t^{(i)}),$$

где мы взяли M примеров траекторий τ .

- Определим $r(\tau) = \sum_t r(s_t, a_t)$. Тогда

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} [r(\tau)] = \int \pi_\theta(\tau) r(\tau) d\tau,$$

т.е. тот самый страшный интеграл по траекториям.

- Но оказывается, что можно продифференцировать по θ ...

- Продифференцируем по θ :

$$\begin{aligned}\nabla_{\theta} J(\theta) &= \int \nabla_{\theta} \pi_{\theta}(\tau) r(\tau) d\tau \\&= \int \pi_{\theta}(\tau) \frac{\nabla_{\theta} \pi_{\theta}(\tau)}{\pi_{\theta}(\tau)} r(\tau) d\tau \\&= \int \pi_{\theta}(\tau) \nabla_{\theta} \log \pi_{\theta}(\tau) r(\tau) d\tau \\&= \mathbb{E}_{\tau \sim \pi_{\theta}(\tau)} [\nabla_{\theta} \log \pi_{\theta}(\tau) r(\tau)] \\&\approx \frac{1}{M} \sum_{i=1}^M \nabla_{\theta} \log \pi_{\theta}(\tau^{(i)}) r^{(i)}(\tau),\end{aligned}$$

если приблизить выборкой; но сначала давайте ещё посмотрим на $\pi_{\theta}(\tau)$...

RECURRENT VISUAL ATTENTION

- Вероятность определяется как

$$\pi_\theta(\tau) = p(s_1) \prod_{t=1}^T \pi_\theta(a_t | s_t) p(s_{t+1} | s_t, a_t).$$

- Берём логарифм, а потом заметим, что от θ зависят только действия:

$$\begin{aligned}\nabla_\theta \log \pi_\theta(\tau) &= \\ &= \nabla_\theta \left(\log p(s_1) + \sum_{t=1}^T \log \pi_\theta(a_t | s_t) + \sum_{t=1}^T \log p(s_{t+1} | s_t, a_t) \right) = \\ &= \sum_{t=1}^T \nabla_\theta \log \pi_\theta(a_t | s_t).\end{aligned}$$

RECURRENT VISUAL ATTENTION

- Итого получается вполне tractable градиент:

$$\begin{aligned}\nabla_{\theta} J(\theta) &= \mathbb{E}_{\tau \sim \pi_{\theta}(\tau)} \left[r(\tau) \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right] \\ &\approx \frac{1}{M} \sum_{i=1}^M r(\tau^{(i)}) \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_t^{(i)} | s_t^{(i)}).\end{aligned}$$

- У нас тоже можно считать, что награда R даётся целиком:

$$\nabla_{\theta} J(\theta) \approx \frac{1}{M} \sum_{i=1}^M \sum_{t=1}^T \nabla_{\theta} \log \pi \left(l_t^{(i)}, a_t^{(i)} | s_{1:t}^{(i)}; \theta \right) R^{(i)}.$$

- Т.е. надо уметь считать $\log \pi \left(l_t^{(i)}, a_t^{(i)} | s_{1:t}^{(i)}; \theta \right)$, но в случае RNN это просто градиент сети, который можно посчитать через backpropagation.
- Ещё можно сделать частично supervised loss на последнем шаге, где мы знаем классификацию.

RECURRENT VISUAL ATTENTION

- Результаты:



(a) Translated MNIST inputs.



(b) Cluttered Translated MNIST inputs.

(a) 28x28 MNIST

Model	Error
FC, 2 layers (256 hiddens each)	1.69%
Convolutional, 2 layers	1.21%
RAM, 2 glimpses, 8×8 , 1 scale	3.79%
RAM, 3 glimpses, 8×8 , 1 scale	1.51%
RAM, 4 glimpses, 8×8 , 1 scale	1.54%
RAM, 5 glimpses, 8×8 , 1 scale	1.34%
RAM, 6 glimpses, 8×8 , 1 scale	1.12%
RAM, 7 glimpses, 8×8 , 1 scale	1.07%

(b) 60x60 Translated MNIST

Model	Error
FC, 2 layers (64 hiddens each)	6.42%
FC, 2 layers (256 hiddens each)	2.63%
Convolutional, 2 layers	1.62%
RAM, 4 glimpses, 12×12 , 3 scales	1.54%
RAM, 6 glimpses, 12×12 , 3 scales	1.22%
RAM, 8 glimpses, 12×12 , 3 scales	1.2%

RECURRENT VISUAL ATTENTION

- Результаты:

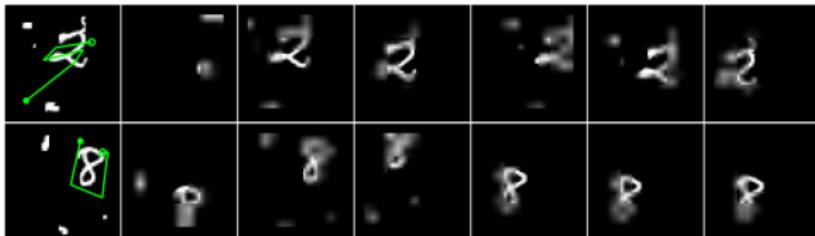
(a) 60x60 Cluttered Translated MNIST

Model	Error
FC, 2 layers (64 hiddens each)	28.58%
FC, 2 layers (256 hiddens each)	11.96%
Convolutional, 2 layers	8.09%
RAM, 4 glimpses, 12×12 , 3 scales	4.96%
RAM, 6 glimpses, 12×12 , 3 scales	4.08%
RAM, 8 glimpses, 12×12 , 3 scales	4.04%
RAM, 8 random glimpses	14.4%

(b) 100x100 Cluttered Translated MNIST

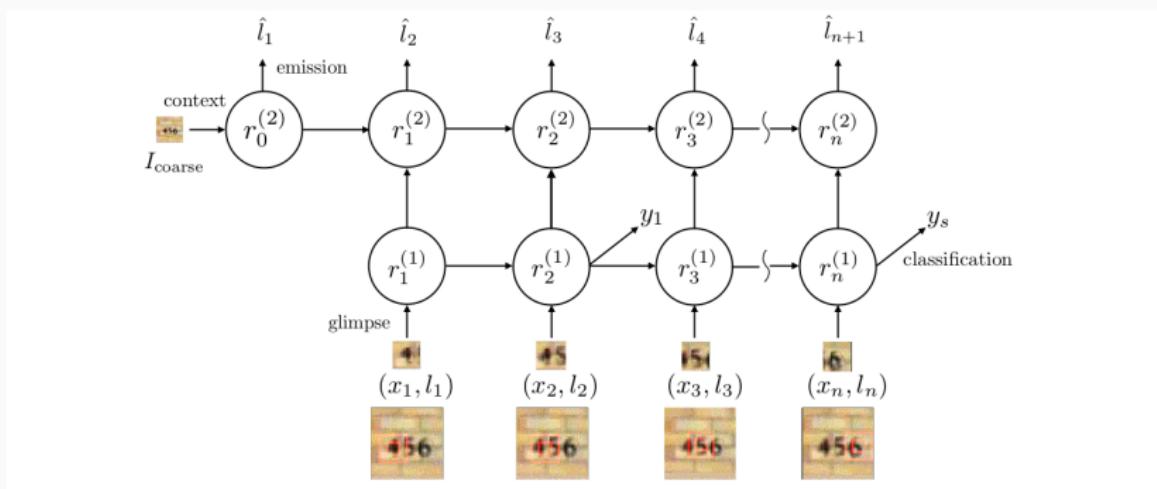
Model	Error
Convolutional, 2 layers	14.35%
RAM, 4 glimpses, 12×12 , 4 scales	9.41%
RAM, 6 glimpses, 12×12 , 4 scales	8.31%
RAM, 8 glimpses, 12×12 , 4 scales	8.11%
RAM, 8 random glimpses	28.4%

- А вот как внимание гуляет по картинке:



RECURRENT VISUAL ATTENTION

- В следующей работе (Ba et al., 2015) сделали глубокую модель:



- Кстати, обучали по-другому, вариационными методами. Как это?..

RECURRENT VISUAL ATTENTION

- Нам нужно классифицировать, т.е. $p(y | \mathbf{x}, \theta)$ максимизировать.
- Маргинализуем по положениям glimpses:

$$\log p(y | \mathbf{x}, \theta) = \log \sum_l p(l | \mathbf{x}, \theta) p(y | l, \mathbf{x}, \theta).$$

- Запишем вариационную нижнюю оценку свободной энергии (как её получить?):

$$\begin{aligned} \log \sum_l p(l | \mathbf{x}, \theta) p(y | l, \mathbf{x}, \theta) &\geq \sum_l p(l | \mathbf{x}, \theta) \log p(y, l | \mathbf{x}, \theta) + H[l] \\ &= \sum_l p(l | \mathbf{x}, \theta) \log p(y | l, \mathbf{x}, \theta). \end{aligned}$$

RECURRENT VISUAL ATTENTION

- И теперь можно брать производные:

$$\begin{aligned}\frac{\partial J}{\partial \theta} &= \sum_l p(l | \mathbf{x}, \theta) \frac{\partial \log p(y | l, \mathbf{x}, \theta)}{\partial \theta} + \sum_l \log p(y | l, \mathbf{x}, \theta) \frac{\partial p(l | \mathbf{x}, \theta)}{\partial \theta} \\ &= \sum_l p(l | \mathbf{x}, \theta) \left[\frac{\partial \log p(y | l, \mathbf{x}, \theta)}{\partial \theta} + \log p(y | l, \mathbf{x}, \theta) \frac{\partial \log p(l | \mathbf{x}, \theta)}{\partial \theta} \right].\end{aligned}$$

- А эту сумму уже будем приближать выборкой:

$$\frac{\partial J}{\partial \theta} \approx \frac{1}{M} \sum_{i=1}^M \left[\frac{\partial \log p(y | l^{(i)}, \mathbf{x}, \theta)}{\partial \theta} + \log p(y | l^{(i)}, \mathbf{x}, \theta) \frac{\partial \log p(l^{(i)} | \mathbf{x}, \theta)}{\partial \theta} \right],$$

где $l^{(i)} \sim p(l_n | \mathbf{x}, \theta) = N(l_n | \hat{l}_n, \Sigma)$.

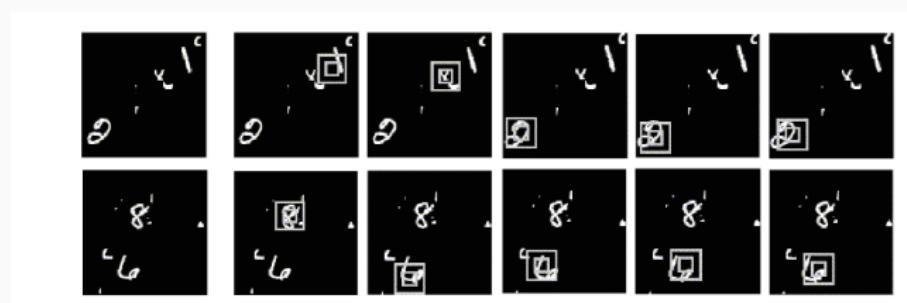
- И это уже алгоритм: сэмплируем glimpses, потом используем их в backpropagation.
- Как и в (Mnih et al., 2014), надо бы уменьшить дисперсию; для этого вычитают baseline, пока не будем углубляться.

RECURRENT VISUAL ATTENTION

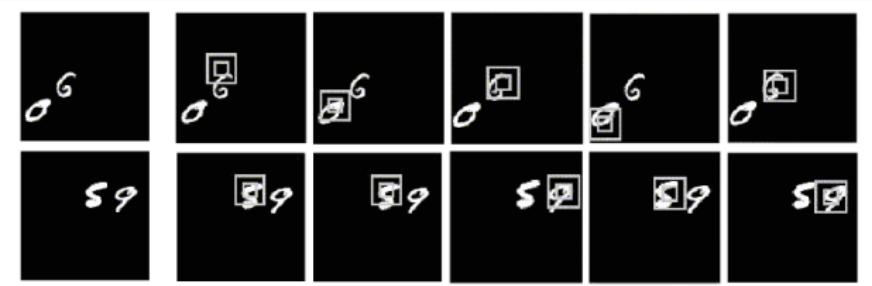
- Получился интересный результат – мы увидели, что примерно один и тот же алгоритм может получиться с двух разных сторон:
 - из обучения с подкреплением через REINFORCE;
 - из вариационной оценки собственно целевой функции.
- Важный гиперпараметр – размер *glimpse*, т.е. как переводить единицы измерений в системе координат *glimpses* в пиксели.
- То же самое легко расширить на последовательную классификацию нескольких объектов – просто фиксированное число *glimpses* на объект, потом классифицируем, плюс терминальное действие в конце всего.

RECURRENT VISUAL ATTENTION

- Вот как это работает на распознавании двух цифр:

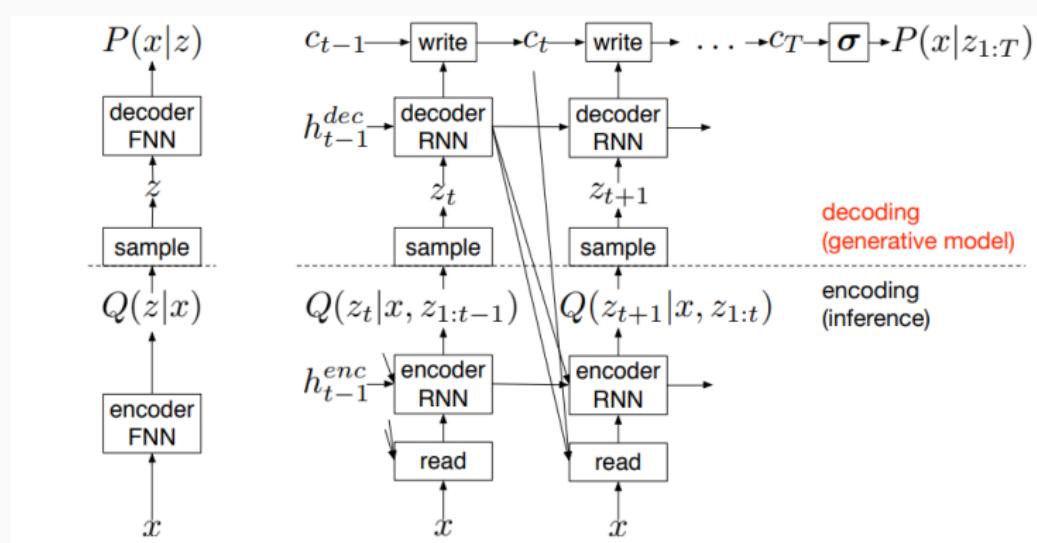


- Любопытно, что на сложении по-другому:



DRAW

- (Gregor et al., 2015): DRAW: A Recurrent Neural Network For Image Generation.
- Следующий шаг в развитии visual attention, Deep Recurrent Attentive Writer (DRAW).
- Это на самом деле вариант на тему вариационного автокодировщика



- Без особых подробностей:

$$\hat{x}_t = x - \sigma(c_{t-1})$$

$$r_t = \text{read}(x_t, \hat{x}_t, h_{t-1}^{\text{dec}})$$

$$h_t^{\text{enc}} = \text{RNN}^{\text{enc}}(h_{t-1}^{\text{enc}}, [r_t, h_{t-1}^{\text{dec}}])$$

$$z_t \sim Q(Z_t | h_t^{\text{enc}})$$

$$h_t^{\text{dec}} = \text{RNN}^{\text{dec}}(h_{t-1}^{\text{dec}}, z_t)$$

$$c_t = c_{t-1} + \text{write}(h_t^{\text{dec}})$$

- Здесь Q – гауссиан, чьи параметры выдаёт encoder, \hat{x} – error image, текущая ошибка порождения.

- Целевая функция $L = L^x + L^z$:
 - reconstruction loss $L^x = -\log p(x \mid c_T)$, где вероятность из распределений Бернулли;
 - latent loss $L^z = \sum_{t=1}^T \text{KL}(Q(Z_t \mid h_t^{\text{enc}}) \| p(Z_t))$, где мы выбираем $p(Z_t)$ сами; например, для стандартного гауссиана
- И можно сделать новую картинку из сети, выбрав \mathbf{z}_t из $p(Z_t)$, а потом картинку из $p(X \mid c_T)$.

- Два варианта read и write.
- Без механизма внимания – это baseline:

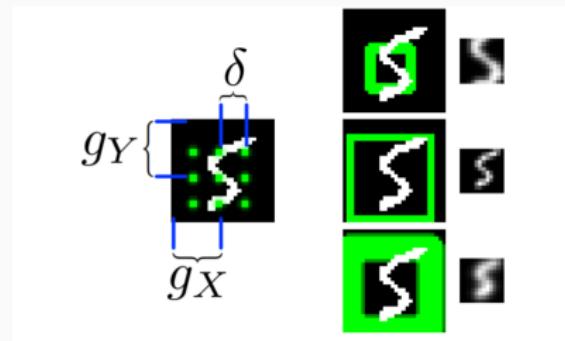
$$\text{read}(x, \hat{x}_t, h_{t-1}^{\text{dec}}) = [x, \hat{x}_t]$$

$$\text{write}(h_t^{\text{dec}}) = W(h_t^{\text{dec}})$$

- С вниманием... тут самое главное, надо сделать внимание так, чтобы сохранить дифференцируемую функцию ошибки и не надо было никакого REINFORCE.

- Давайте покроем картинку $N \times N$ прямоугольной сеткой гауссовских фильтров, т.е. выберем центр (g_X, g_Y) и шаг δ :

$$\mu_X^i = g_X + (i - N/2 - 1/2)\delta,$$
$$\mu_Y^i = g_Y + (j - N/2 - 1/2)\delta.$$



- Ещё надо определить дисперсию σ^2 и интенсивность γ для фильтров.

- Все эти параметры будут из выхода декодера определяться; для картинки $A \times B$:

$$(\tilde{g}_X, \tilde{g}_Y, \log \sigma^2, \log \tilde{\delta}, \log \gamma) = W(\mathbf{h}^{\text{dec}}),$$

$$g_X = \frac{A+1}{2}(\tilde{g}_X + 1), \quad g_Y = \frac{B+1}{2}(\tilde{g}_Y + 1), \quad \delta = \frac{\max(A, B) - 1}{N - 1}\tilde{\delta},$$

и масштаб такой, чтобы первый участок примерно покрывал всю картинку.

- Теперь можно построить вертикальные и горизонтальные матрицы банка фильтров:

$$F_X[i, a] = \frac{1}{Z_X} e^{-\frac{(a - \mu_X^i)^2}{2\sigma^2}}, \quad F_Y[i, a] = \frac{1}{Z_Y} e^{-\frac{(b - \mu_Y^i)^2}{2\sigma^2}},$$

где (i, j) – точка в участке внимания, (a, b) – точка на картинке.



- Ну и теперь переопределим read и write:

$$\text{read}(\mathbf{x}, \hat{\mathbf{x}}_t, \mathbf{h}_{t-1}^{\text{dec}}) = \gamma [F_Y \mathbf{x} F_X^\top, F_Y \hat{\mathbf{x}} F_X^\top],$$

$$\text{write}(\mathbf{h}_t^{\text{dec}}) = \frac{1}{\hat{\gamma}} \hat{F}_Y^\top \mathbf{w}_t \hat{F}_X,$$

где $\mathbf{w}_t = W(\mathbf{h}_t^{\text{dec}})$ – writing patch, выданный $\mathbf{h}_t^{\text{dec}}$.

- И можно рисовать:

<https://www.youtube.com/watch?v=Zt-7MI9eKEo>

DRAW

- А вот как DRAW классифицирует:

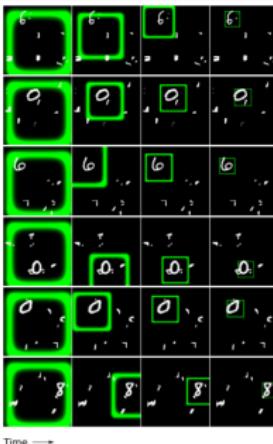


Table 1. Classification test error on 100×100 Cluttered Translated MNIST.

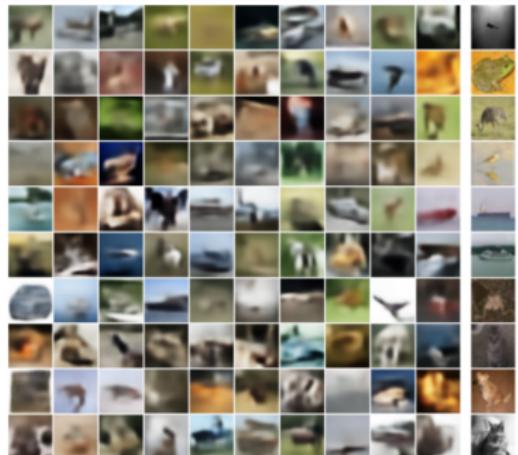
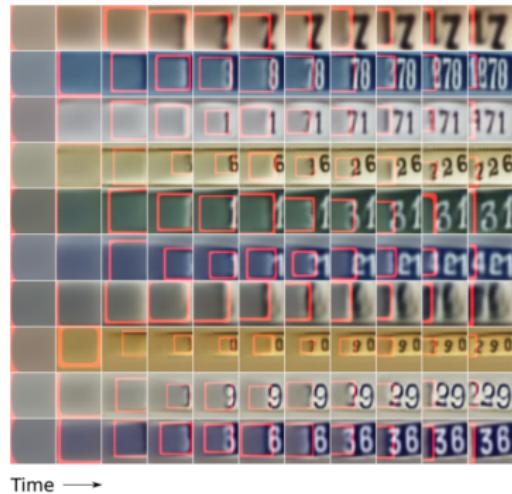
Model	Error
Convolutional, 2 layers	14.35%
RAM, 4 glimpses, 12×12 , 4 scales	9.41%
RAM, 8 glimpses, 12×12 , 4 scales	8.11%
Differentiable RAM, 4 glimpses, 12×12	4.18%
Differentiable RAM, 8 glimpses, 12×12	3.36%

- Примеры порождения, SVHN:



DRAW

- Примеры порождения, SVHN и CIFAR:



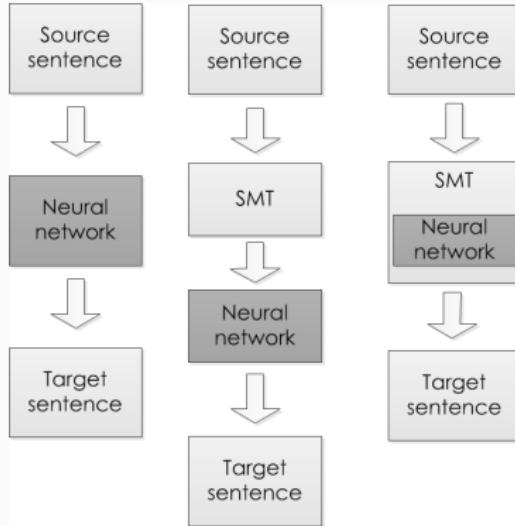
МАШИННЫЙ ПЕРЕВОД: ENCODER-DECODER И ВНИМАНИЕ

МАШИННЫЙ ПЕРЕВОД

- Перевод – очень хорошая задача:
 - очевидно очень практическая;
 - очевидно очень высокоуровневая, требует понимания;
 - считается довольно неплохо квантифицируемой (BLEU, TER – хотя см. выше);
 - имеет большие доступные датасеты параллельных переводов.

МАШИННЫЙ ПЕРЕВОД

- Статистический машинный перевод (statistical machine translation, SMT): моделируем условную вероятность $p(y | x)$ перевода y при условии исходного текста x .
- Классический SMT: моделируем $\log p(y | x)$ линейной комбинацией признаков, строим признаки.



МАШИННЫЙ ПЕРЕВОД

- Нам больше интересно моделирование sequence-to-sequence:
 - RNN естественным образом моделирует последовательность $X = (x_1, x_2, \dots, x_T)$ как $p(x_1), p(x_2 | x_1), \dots, p(x_T | x_{<T}) = p(x_T | x_{T-1}, \dots, x_1)$, и теперь $p(X)$ – это просто
$$p(X) = p(x_1)p(x_2 | x_1) \dots p(x_k | x_{<k}) \dots p(x_T | x_{<T});$$
 - так RNN и в языковых моделях используются;
 - предсказываем следующее слово на основе скрытого состояния и предыдущего слова;
- Как применить эту идею к переводу?

МЕТРИКИ КАЧЕСТВА ДЛЯ SEQUENCE-TO-SEQUENCE МОДЕЛЕЙ

- Дальше будет самое интересное: машинный перевод, диалоговые модели, ответ на вопросы.
- Но как мы будем оценивать NLP-модели, которые генерируют текст?
- Есть метрики качества, которые сравнивают результат с правильными ответами:
 - BLEU (Bilingual Evaluation Understudy): перевзвешенная precision (в т.ч. для нескольких правильных ответов);
 - METEOR: гармоническое среднее precision и recall по униграммам;
 - TER (Translation Edit Rate): число исправлений между выходом и правильным ответом, делённое на среднее число слов;
 - LEPOR: комбинируем базовые факторы и метрики с настраиваемыми параметрами.
- Есть ещё куча метрик, связанных с представлениями слов и предложений (хотим поближе кциальному ответу).
- Одна только проблема...

МЕТРИКИ КАЧЕСТВА ДЛЯ SEQUENCE-TO-SEQUENCE МОДЕЛЕЙ

- ...всё это вообще не работает.

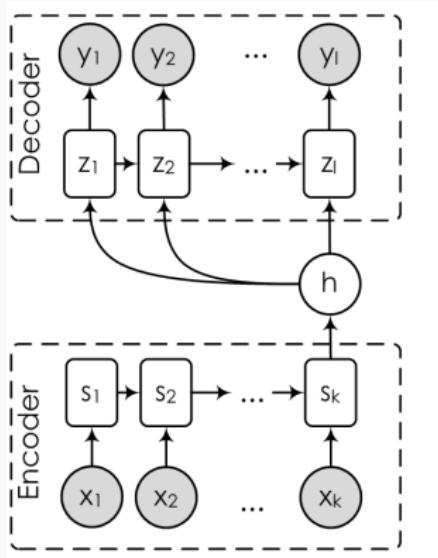
Metric	Twitter				Ubuntu			
	Spearman	p-value	Pearson	p-value	Spearman	p-value	Pearson	p-value
Greedy	0.2119	0.034	0.1994	0.047	0.05276	0.6	0.02049	0.84
Average	0.2259	0.024	0.1971	0.049	-0.1387	0.17	-0.1631	0.10
Extrema	0.2103	0.036	0.1842	0.067	0.09243	0.36	-0.002903	0.98
METEOR	0.1887	0.06	0.1927	0.055	0.06314	0.53	0.1419	0.16
BLEU-1	0.1665	0.098	0.1288	0.2	-0.02552	0.8	0.01929	0.85
BLEU-2	0.3576	< 0.01	0.3874	< 0.01	0.03819	0.71	0.0586	0.56
BLEU-3	0.3423	< 0.01	0.1443	0.15	0.0878	0.38	0.1116	0.27
BLEU-4	0.3417	< 0.01	0.1392	0.17	0.1218	0.23	0.1132	0.26
ROUGE	0.1235	0.22	0.09714	0.34	0.05405	0.5933	0.06401	0.53
Human	0.9476	< 0.01	1.0	0.0	0.9550	< 0.01	1.0	0.0

Table 3: Correlation between each metric and human judgements for each response. Correlations shown in the human row result from randomly dividing human judges into two groups.

- Тут нужно что-то новое. И пока не совсем ясно, что именно.

ENCODER-DECODER АРХИТЕКТУРЫ

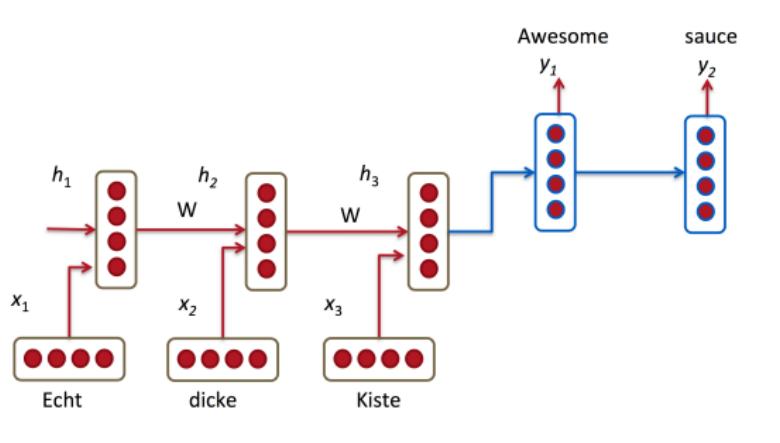
- Encoder-decoder архитектуры (Sutskever et al., 2014; Cho et al., 2014):



- Сначала кодируем, потом декодируем обратно.

ENCODER-DECODER АРХИТЕКТУРЫ

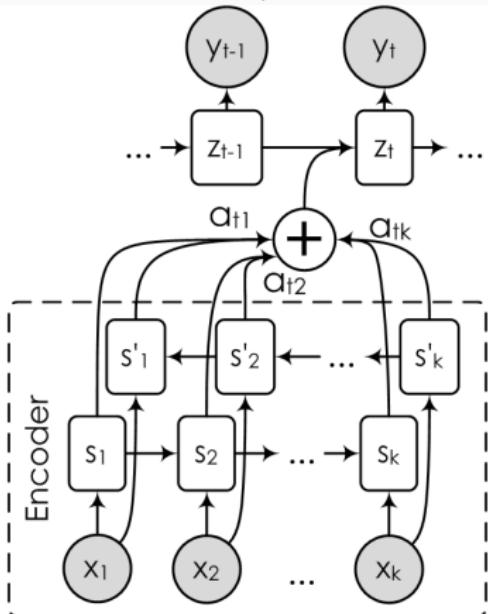
- Так же может работать и с переводом.



- Проблема: надо сжимать всё предложение в один вектор.
- С длинными участками текста это вообще перестаёт работать.

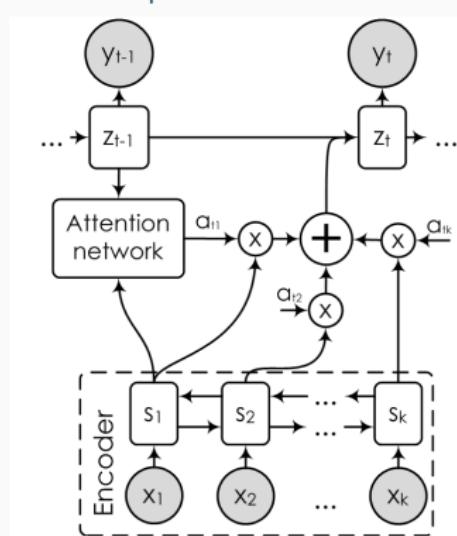
ВНИМАНИЕ В НЕЙРОННЫХ СЕТЯХ

- Решение: давайте обучим специальные веса, показывающие, насколько та или иная часть входа важна для текущего выхода.
- Прямое применение – двунаправленный LSTM плюс внимание (Bahdanau et al. 2014):



ВНИМАНИЕ В НЕЙРОННЫХ СЕТЯХ

- Мягкое внимание (soft attention) (Luong et al. 2015a; 2015b; Jean et al. 2015):
 - encoder – двунаправленная RNN, есть оба контекста;
 - сеть внимания выдаёт оценку релевантности – надо ли переводить это слово прямо сейчас?

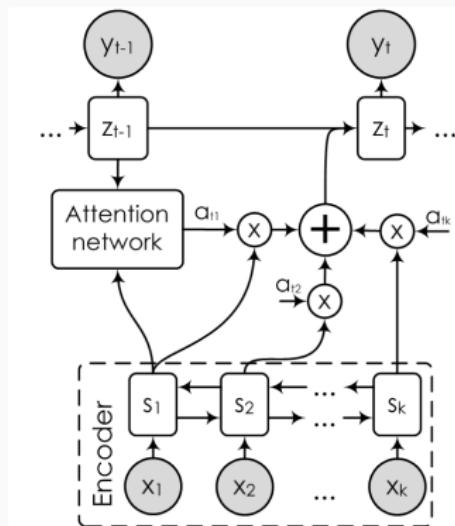


ВНИМАНИЕ В НЕЙРОННЫХ СЕТЯХ

- Формально очень просто: считаем веса внимания α_{tj} и перевзвешиваем векторы контекстов:

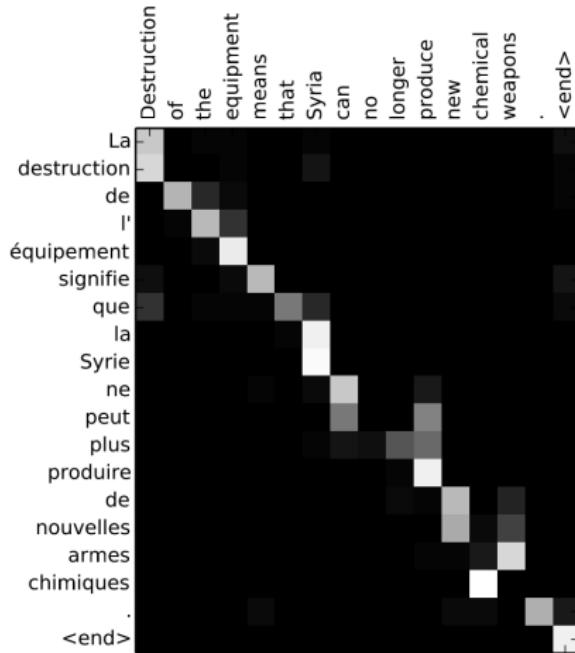
$$e_{tj} = a(z_{t-1}, j), \quad \alpha_{tj} = \text{softmax}(e_{tj}; e_{t*}),$$

$$c_t = \sum_j \alpha_{tj} h_j, \text{ и теперь } z_t = f(s_{t-1}, y_{t-1}, c_t).$$



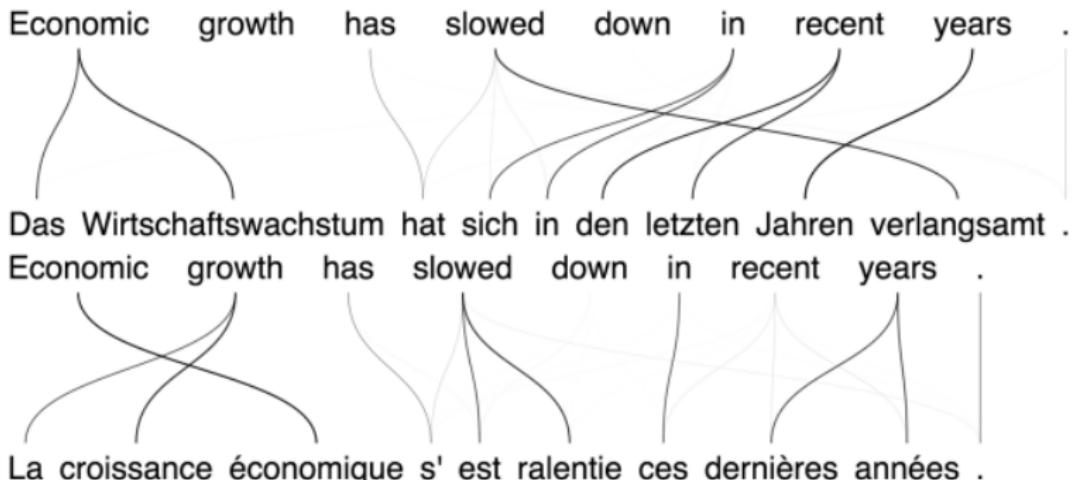
ВНИМАНИЕ В НЕЙРОННЫХ СЕТЯХ

- В результате можно визуализировать, на что смотрит сеть:



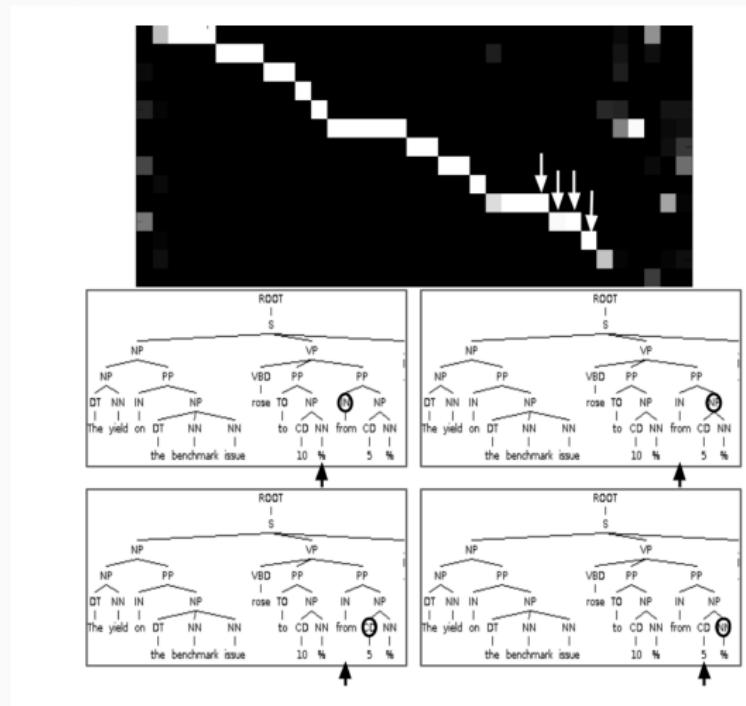
ВНИМАНИЕ В НЕЙРОННЫХ СЕТЯХ

- Получается гораздо лучше порядок слов:



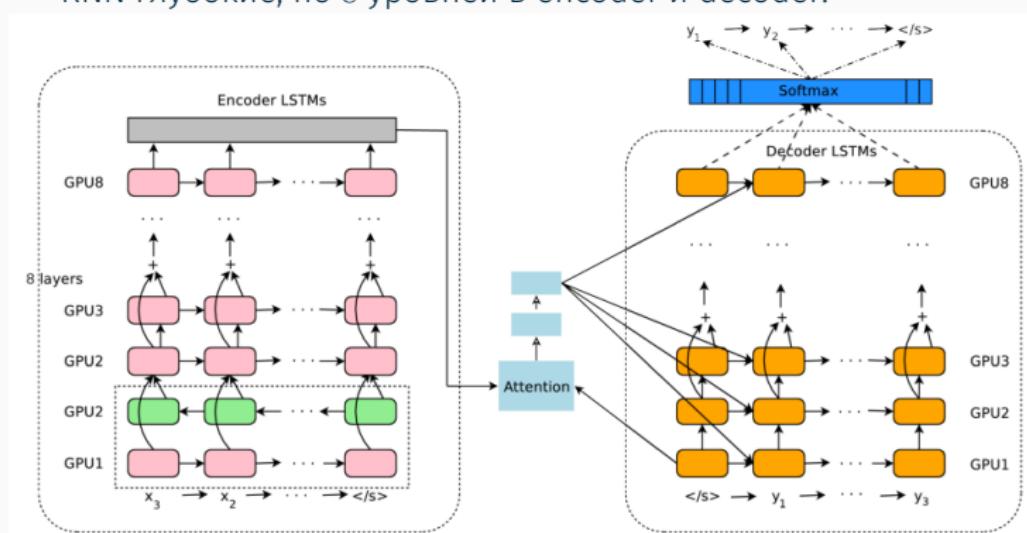
ВНИМАНИЕ В НЕЙРОННЫХ СЕТЯХ

- Другая необычная работа – «Grammar as a Foreign Language» (Vinyals et al., 2015)



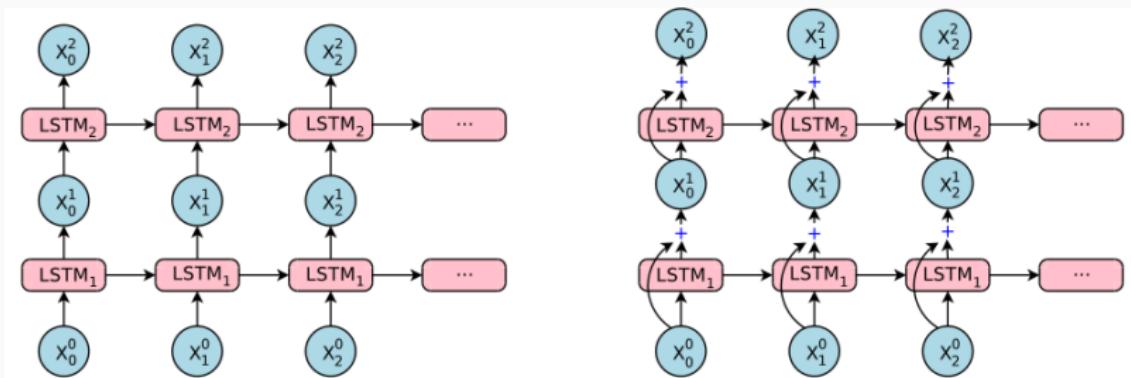
GOOGLE TRANSLATE

- Сентябрь 2016: Wu et al., *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*:
 - как на самом деле работает Google Translate;
 - базовая архитектура та же самая: encoder, decoder, attention;
 - RNN глубокие, по 8 уровней в encoder и decoder:



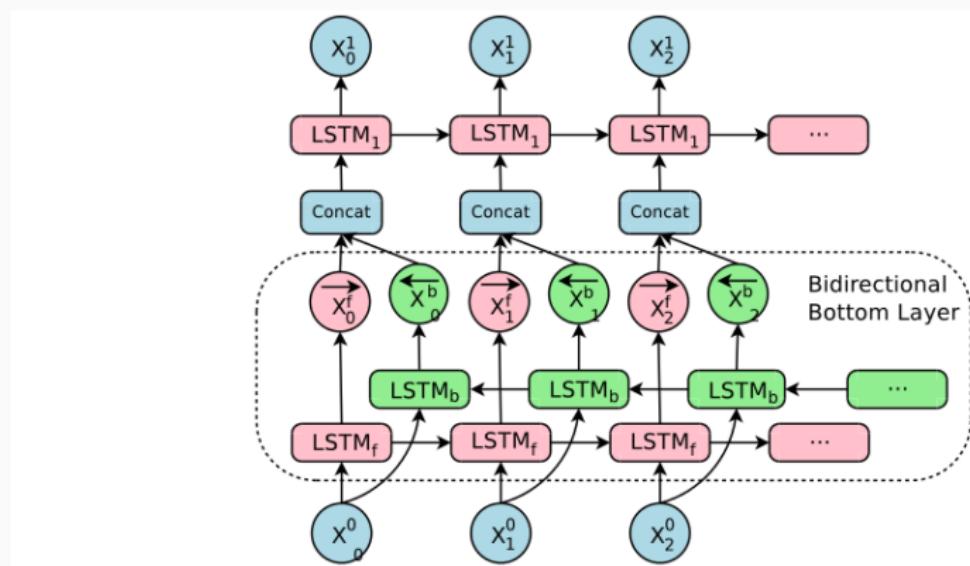
GOOGLE TRANSLATE

- Сентябрь 2016: Wu et al., *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*:
 - просто stacked LSTM перестают работать далее 4-5 уровней;
 - поэтому добавляют остаточные связи, как в ResNet:



GOOGLE TRANSLATE

- Сентябрь 2016: Wu et al., *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*:
 - нижний уровень, естественно, двунаправленный:



- Сентябрь 2016: Wu et al., *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*:
- в GNMT есть две идеи о сегментации слов:
 - *wordpiece model*: разбить слова на кусочки (отдельной моделью); пример из статьи:

Jet makers feud over seat width with big orders at stake

превращается в

_J et _makers _fe ud _over _seat _width _with _big _orders _at _stake

- *mixed word/character model*: конвертировать слова, не попадающие в словарь, в последовательность букв-токенов; пример из статьи:

Miki превращается в M <M>i <M>k <E>i

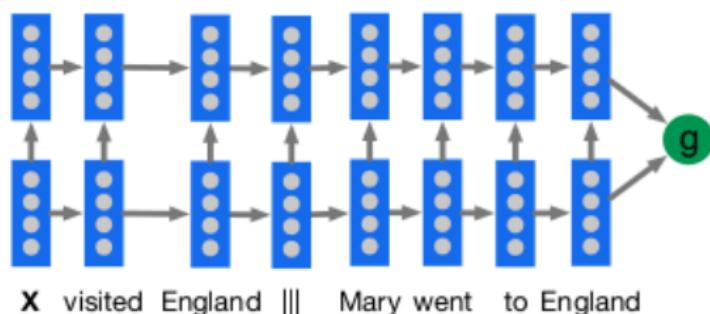
TEACHING MACHINES TO READ

- (Hermann et al., 2015): «Teaching machines to read and comprehend» (Google DeepMind)
- Предлагают новый способ построить датасет для понимания, автоматически создавая тройки (context, query, answer) из текстов новостей и т.п.

Original Version	Anonymised Version
Context <p>The BBC producer allegedly struck by Jeremy Clarkson will not press charges against the “Top Gear” host, his lawyer said Friday. Clarkson, who hosted one of the most-watched television shows in the world, was dropped by the BBC Wednesday after an internal investigation by the British broadcaster found he had subjected producer Oisin Tymon “to an unprovoked physical and verbal attack.” ...</p>	<p>the <i>ent381</i> producer allegedly struck by <i>ent212</i> will not press charges against the “ <i>ent153</i> ” host , his lawyer said friday . <i>ent212</i> , who hosted one of the most - watched television shows in the world , was dropped by the <i>ent381</i> wednesday after an internal investigation by the <i>ent180</i> broadcaster found he had subjected producer <i>ent193</i> “ to an unprovoked physical and verbal attack . ” ...</p>
Query <p>Producer X will not press charges against Jeremy Clarkson, his lawyer says.</p>	<p>producer X will not press charges against <i>ent212</i> , his lawyer says .</p>
Answer <p>Oisin Tymon</p>	<p><i>ent193</i></p>

TEACHING MACHINES TO READ

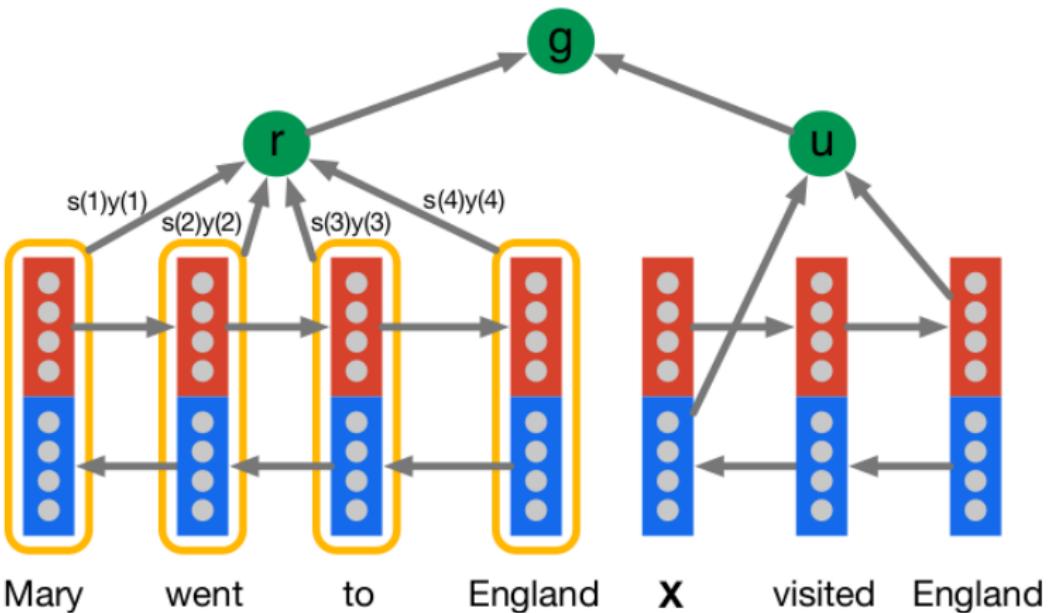
- Базовая модель – глубокий LSTM:



- Но так, конечно, совсем плохо работает.

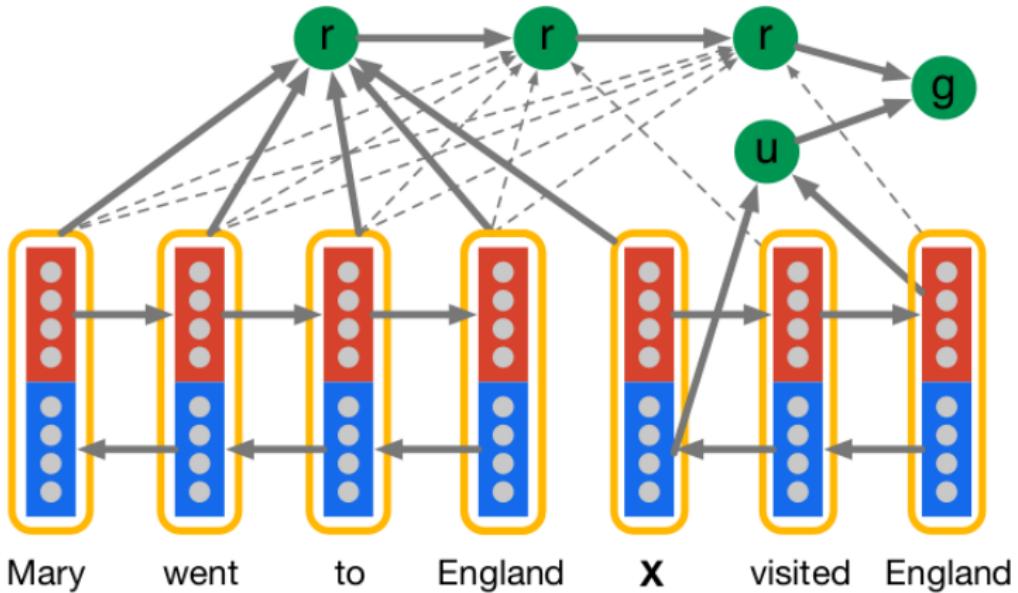
TEACHING MACHINES TO READ

- Attentive Reader: обучаемся, на какую часть документа смотреть



TEACHING MACHINES TO READ

- Impatient Reader: можем перечитывать нужные части документа по мере прочтения запроса



TEACHING MACHINES TO READ

- Получаются разумные карты внимания:

by ent423 ,ent261 correspondent updated 9:49 pm et ,thu march 19 ,2015 (ent261) a ent114 was killed in a parachute accident in ent45 ,ent85 ,near ent312 ,a ent119 official told ent261 on wednesday .he was identified thursday as special warfare operator 3rd class ent23 ,29 ,of ent187 ,ent265 .`` ent23 distinguished himself consistently throughout his career .he was the epitome of the quiet professional in all facets of his life ,and he leaves an inspiring legacy of natural tenacity and focused

...

ent119 identifies deceased sailor as X ,who leaves behind a wife

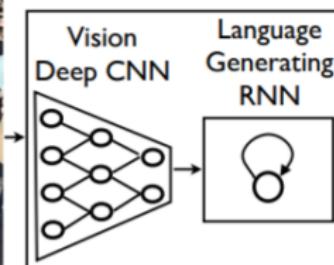
by ent270 ,ent223 updated 9:35 am et ,mon march 2 ,2015 (ent223) ent63 went familial for fall at its fashion show in ent231 on sunday ,dedicating its collection to `` mamma '' with nary a pair of `` mom jeans "in sight .ent164 and ent21 ,who are behind the ent196 brand ,sent models down the runway in decidedly feminine dresses and skirts adorned with roses ,lace and even embroidered doodles by the designers 'own nieces and nephews .many of the looks featured saccharine needlework phrases like `` i love you ,

...

X dedicated their fall fashion show to moms

Show, ATTEND, AND TELL

- Теперь давайте про подписи к картинкам.
- Сначала было «Show and Tell» (Vinyals et al., 2015):

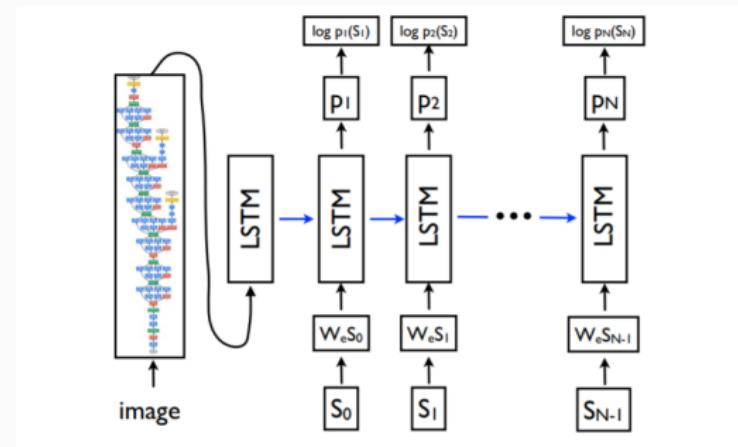


A group of people shopping at an outdoor market.

There are many vegetables at the fruit stand.

Show, ATTEND, AND TELL

- Довольно прямолинейная архитектура:
 - целевая функция – это просто $\sum_{(I,S)} \log p(S | I; \theta)$, где I – картинка, S – описание;
 - раскладываем и моделируем $p(S_t | I, S_0, \dots, S_{t-1})$ рекуррентной сетью с LSTM;
 - а CNN используем, чтобы извлечь признаки.



SHOW, ATTEND, AND TELL

- Получалось хорошо, но можно лучше:

A person riding a motorcycle on a dirt road.



Two dogs play in the grass.



A skateboarder does a trick on a ramp.



A dog is jumping to catch a frisbee.



A group of young people playing a game of frisbee.



Two hockey players are fighting over the puck.



A little girl in a pink hat is blowing bubbles.



A refrigerator filled with lots of food and drinks.



A herd of elephants walking across a dry grass field.



A close up of a cat laying on a couch.



A red motorcycle parked on the side of the road.



A yellow school bus parked in a parking lot.



Describes without errors

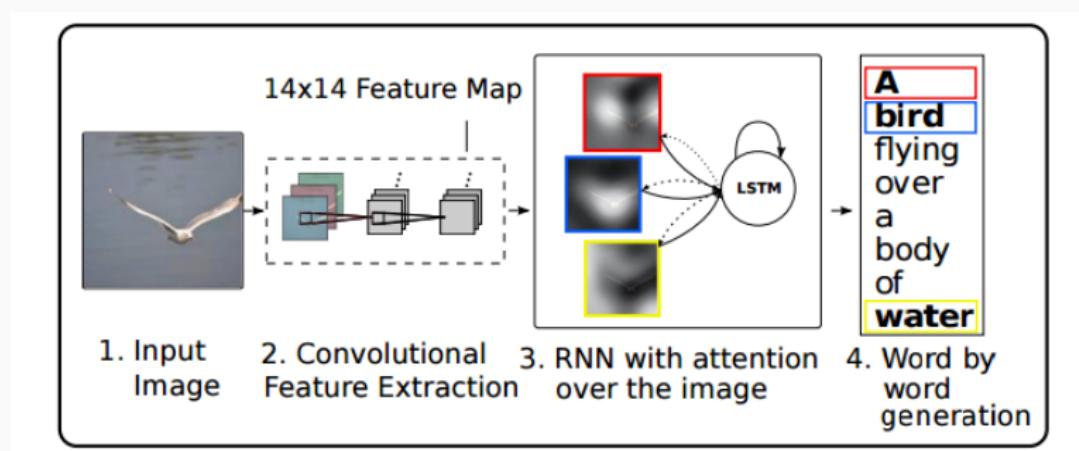
Describes with minor errors

Somewhat related to the image

Unrelated to the image

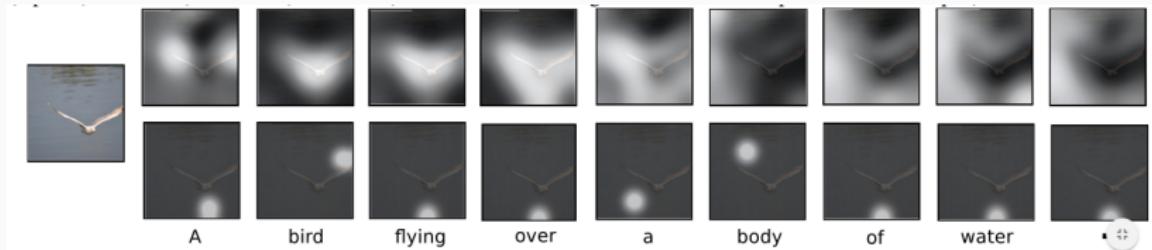
SHOW, ATTEND, AND TELL

- Из этого появилось «Show, Attend, and Tell» (Xu et al., 2015)



Show, ATTEND, AND TELL

- Soft attention vs. hard attention (стохастически выбираем однозначный кусок картинки).



- Soft attention – строим аннотацию с весами

$$\phi(\{\mathbf{a}\}_i, \{\alpha_i\}) = \sum_{i=1}^L \alpha_{t,i} \mathbf{a}_i.$$

- Hard attention обучается максимизацией вариационной нижней оценки

$$L_s = \sum_s p(s | \mathbf{a}) \log p(\mathbf{y} | s, \mathbf{a}) \leq \log \sum_s p(s | \mathbf{a}) p(\mathbf{y} | s, \mathbf{a}) = \log p(\mathbf{y} | \mathbf{a}).$$

- От L_s можно брать производные:

$$\frac{\partial L_s}{\partial W} = \sum_s p(s | \mathbf{a}) \left[\frac{\partial \log p(\mathbf{y} | s, \mathbf{a})}{\partial W} + \log p(\mathbf{y} | s, \mathbf{a}) \frac{\partial \log p(s | \mathbf{a})}{\partial W} \right].$$

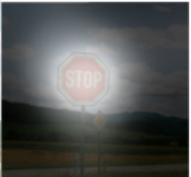
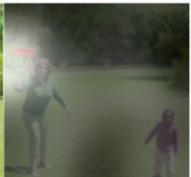
- И дальше сэмплируем s_t с вероятностями α_i и приближаем ожидание выборкой.
- Опять те же трюки, вычитаем baseline, всё такое.

SHOW, ATTEND, AND TELL

- Часто получаются очень хорошие результаты:



A woman is throwing a frisbee in a park.



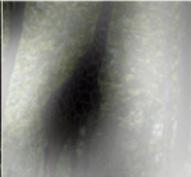
A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.



A giraffe standing in a forest with trees in the background.

- А когда плохие, можно посмотреть почему.

SHOW, ATTEND, AND TELL

- Примеры – hard attention:



SHOW, ATTEND, AND TELL

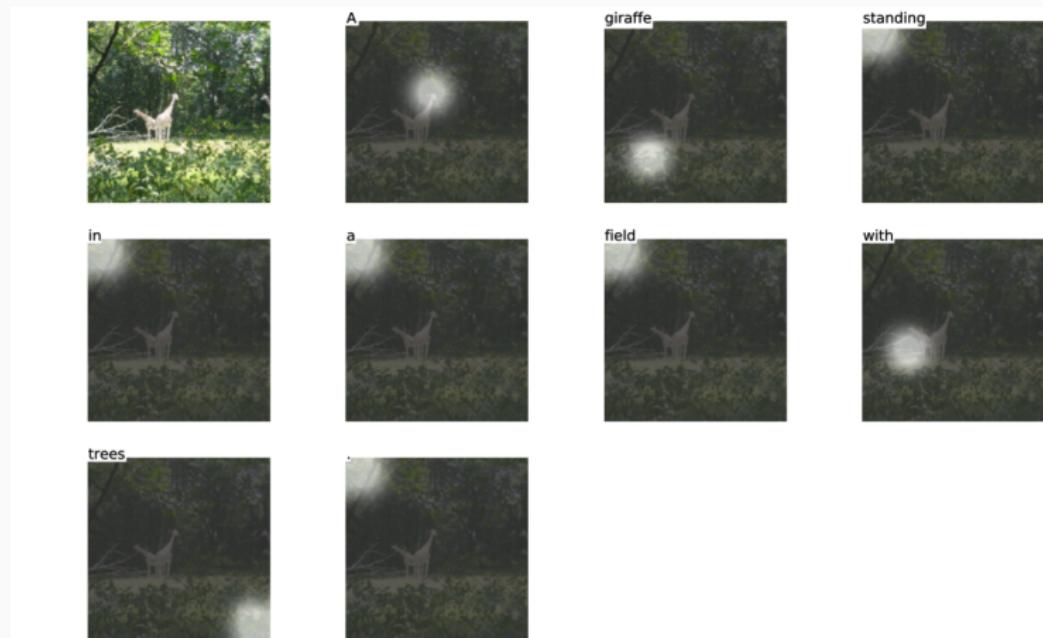
- Примеры – soft attention:



(b) A woman is throwing a frisbee in a park.

SHOW, ATTEND, AND TELL

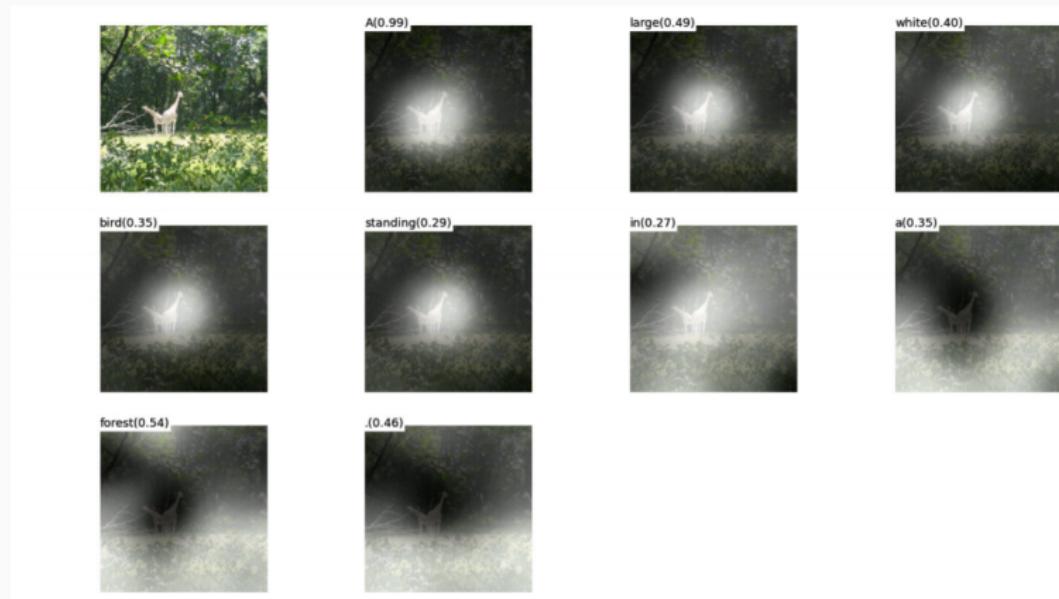
- Примеры – hard attention:



(a) A giraffe standing in the field with trees.

SHOW, ATTEND, AND TELL

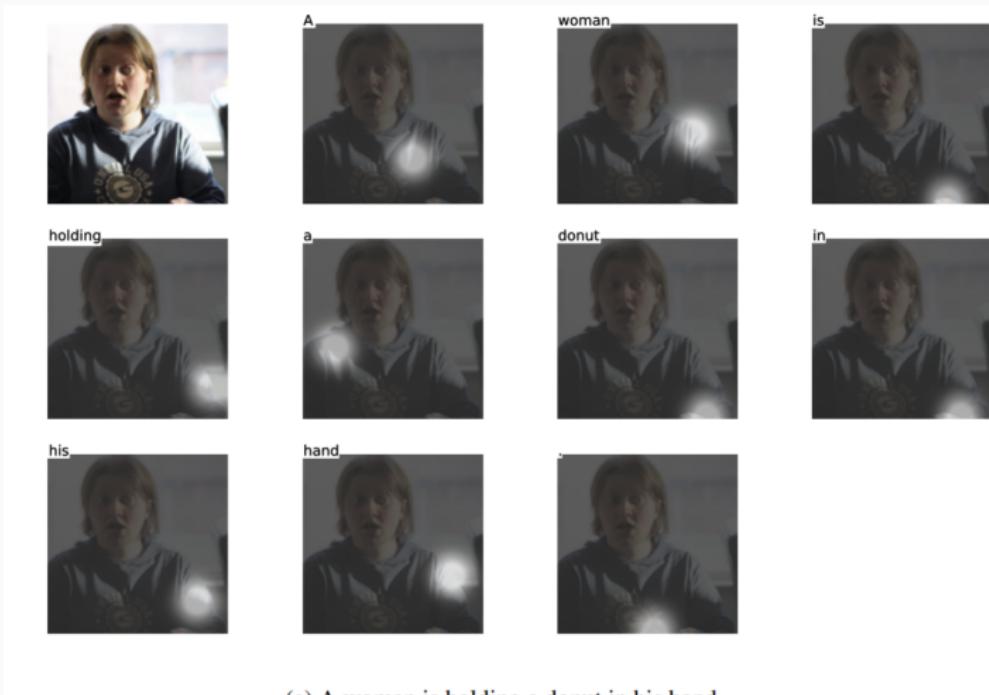
- Примеры – soft attention:



(b) A large white bird standing in a forest.

SHOW, ATTEND, AND TELL

- Примеры – hard attention:



SHOW, ATTEND, AND TELL

- Примеры – soft attention:

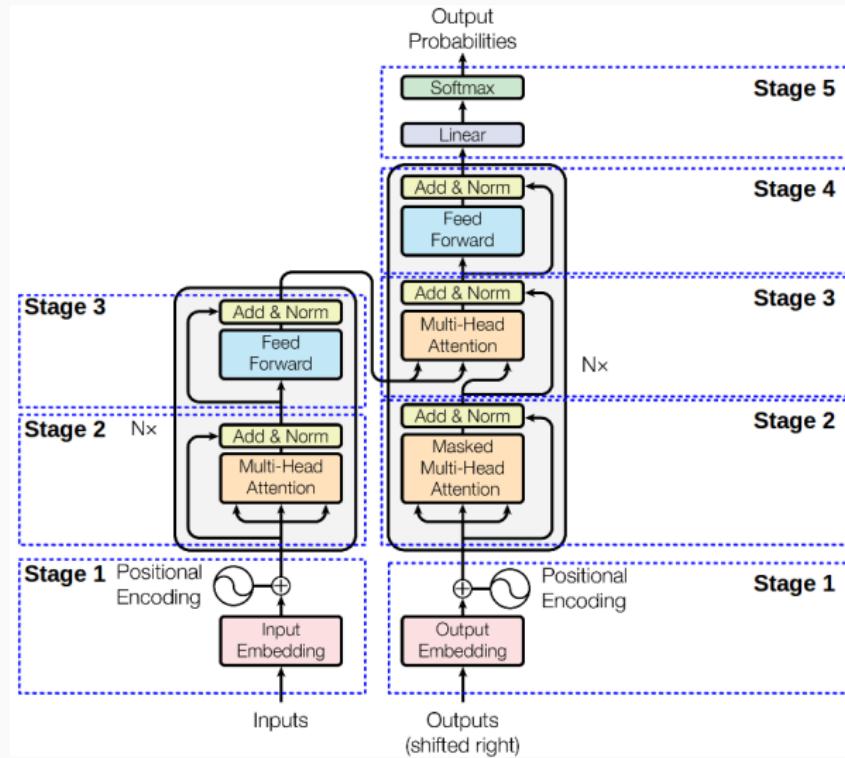


TRANSFORMER И ЧТО ИЗ НЕГО ПОЛУЧИЛОСЬ

- Мы изучали перевод на рекуррентных сетях и дошли до архитектуры Google NMT
- Но в 2017 году оказалось, что всё может быть ещё проще и интереснее
- Google: «Attention is all you need» (Vaswani et al., 2017)
- Основная идея – self-attention; оказывается очень плодотворной для всевозможных seq2seq задач
- Главная мотивация – попробовать всё-таки уйти от кодирования вектором постоянной длины

TRANSFORMER

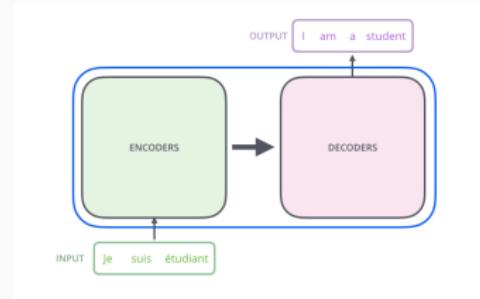
- Общая схема:



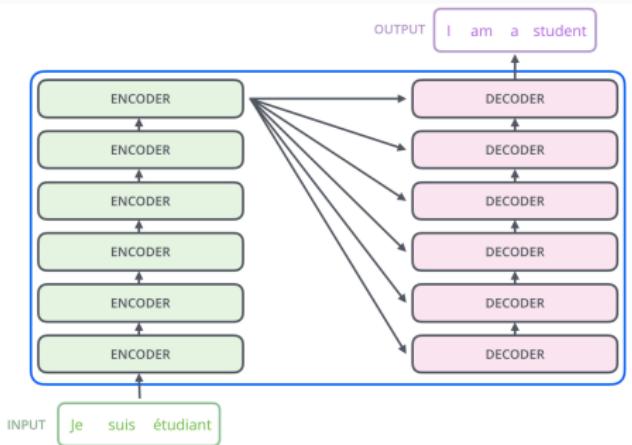
- Теперь подробнее...

TRANSFORMER

- Суть, как и раньше, – encoder-decoder:

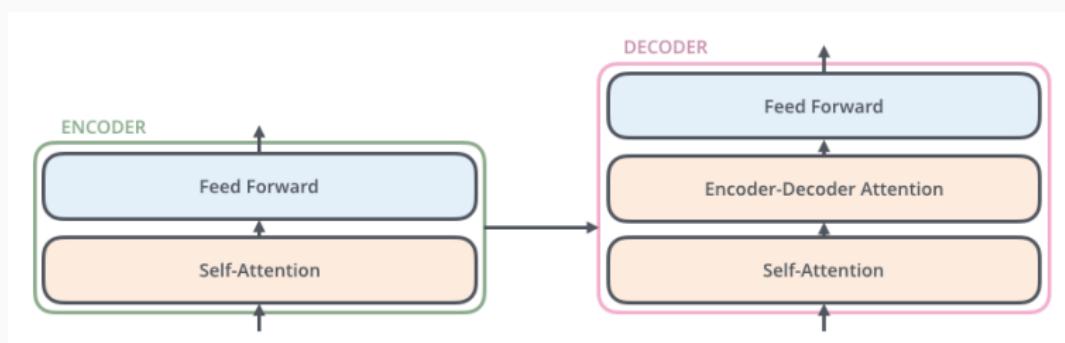


- 6 слоёв encoder'а, результат потом дают 6 слоям декодера:



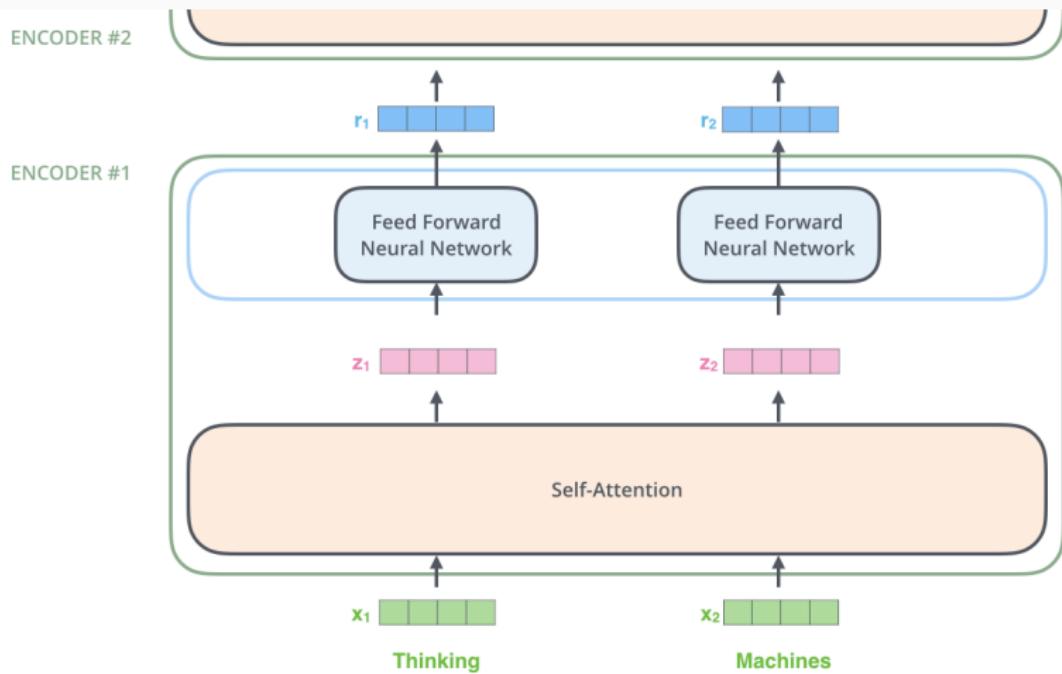
TRANSFORMER

- В каждом слое – слой self-attention, а потом feedforward layer, который независимо применяется к каждой позиции входа. У декодера ещё есть attention между ними:



TRANSFORMER

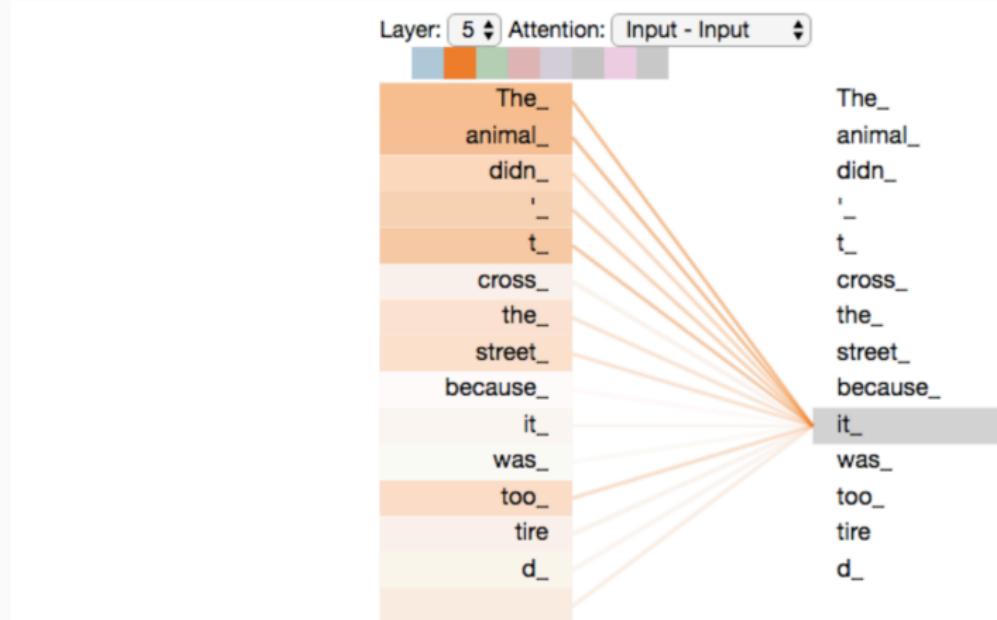
- Слова, естественно, представляются векторами, в feedforward слое всё параллельно:



- Но что же это такое – self-attention?

TRANSFORMER

- Идея в том, чтобы обучить веса, с которыми обработка текущего слова будет учитывать другие слова:



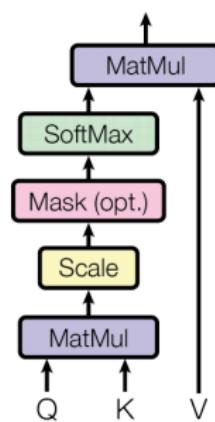
- Теперь детально...

TRANSFORMER

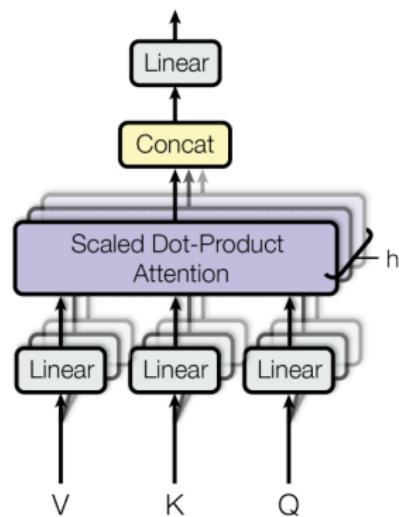
- Scaled Dot-Product Attention состоит из queries, keys и values:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{1}{\sqrt{d_k}} QK^\top \right) V$$

Scaled Dot-Product Attention



Multi-Head Attention



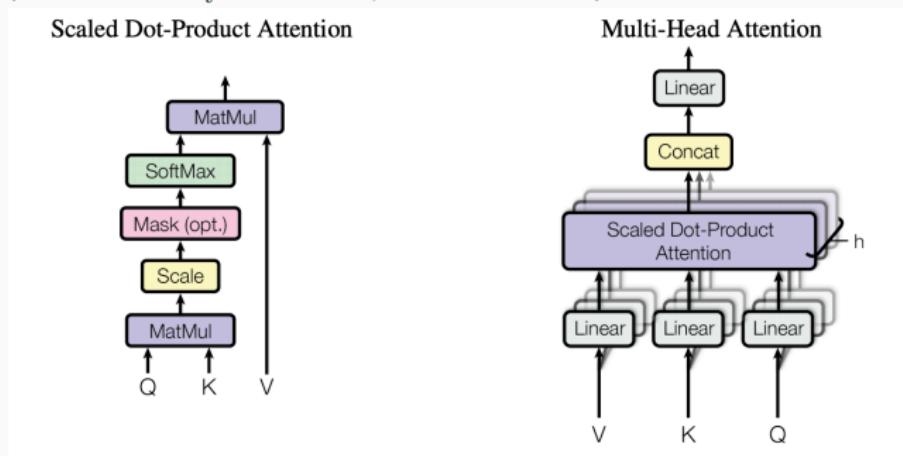
TRANSFORMER

- Multi-head attention объединяет несколько self-attention карт в общие матричные вычисления:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^O,$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V),$$

где проекции W_i^* – это обучаемые матрицы весов.



- Последний вопрос – сейчас модель совсем не учитывает порядок слов!
- Для этого добавляем к представлениям слов ещё positional embeddings:

$$\text{PE}_{(\text{pos}, 2i)} = \sin\left(\text{pos}/10000^{2i/d_{\text{model}}}\right),$$

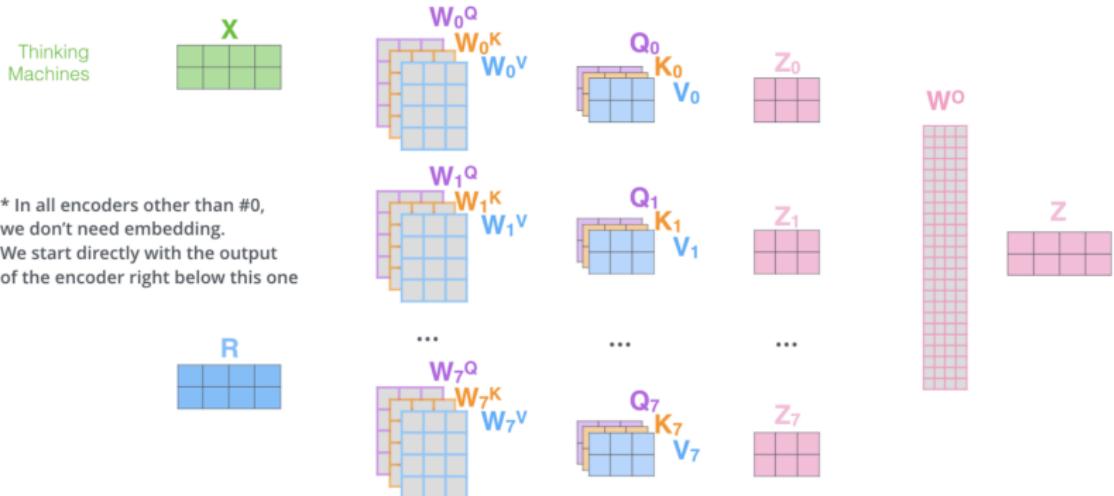
$$\text{PE}_{(\text{pos}, 2i + 1)} = \cos\left(\text{pos}/10000^{2i/d_{\text{model}}}\right),$$

т.е. по каждой размерности i идёт своя синусоида; идея \sin/\cos в том, чтобы для каждого фиксированного k $\text{PE}_{\text{pos}+k}$ было бы линейной функцией от PE_{pos} , и это облегчило бы обучение того, как смотреть на относительные смещения.

TRANSFORMER

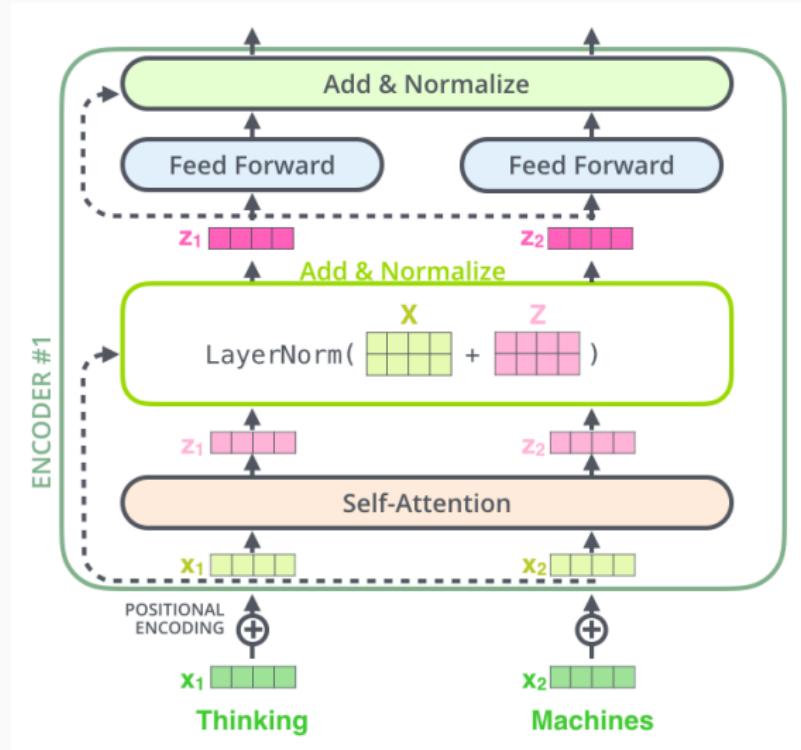
- Putting it all together:

- This is our input sentence*
- We embed each word*
- Split into 8 heads. We multiply X or R with weight matrices
- Calculate attention using the resulting $Q/K/V$ matrices
- Concatenate the resulting Z matrices, then multiply with weight matrix W^o to produce the output of the layer



TRANSFORMER

- Putting it all together:



TRANSFORMER

- Бонус от self-attention – во-первых, вычислительный, во-вторых, сокращает пути между словами, в-третьих, потенциальная интерпретируемость.

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(\log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

TRANSFORMER

- Работает лучше, обучается в сто раз быстрее:

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [15]	23.75			
Deep-Att + PosUnk [32]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [31]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [8]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [26]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [32]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [31]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [8]	26.36	41.29	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$3.3 \cdot 10^{18}$	
Transformer (big)	28.4	41.0	$2.3 \cdot 10^{19}$	

- Следующий шаг – OpenAI GPT (Generative Pretrained Transformer) (Radford et al., 2018)
- Используем Transformer в такой последовательности:
 - сначала обучаем обычную языковую модель

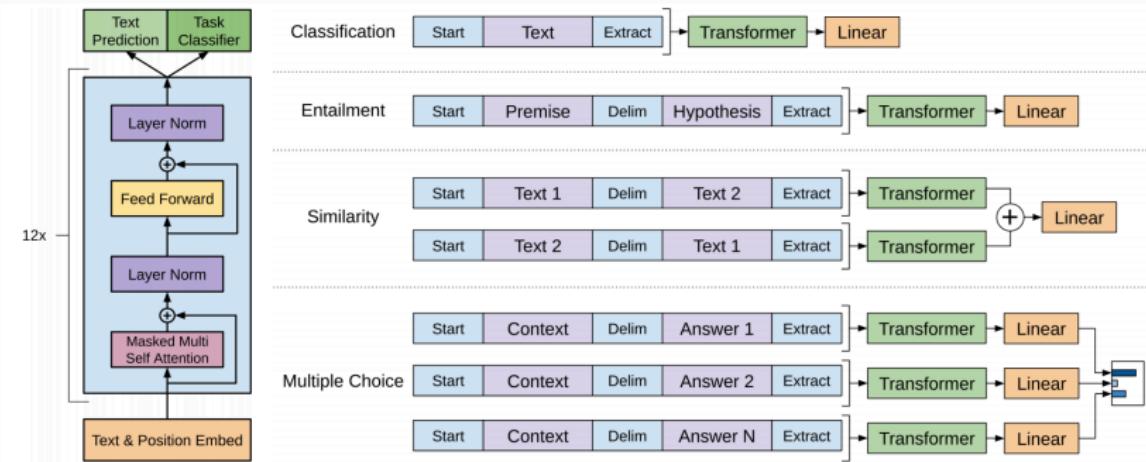
$$L_1(D) = \sum_i \log p(u_i | u_{i-k}, \dots, u_{i-1}; \theta);$$

- потом добавляем новый линейный слой для каждой задачи и делаем fine-tuning уже с учителем:

$$L(C, D) = \sum_{(\mathbf{x}, y)} \log p(y | \mathbf{x}) + \lambda L_1(D).$$

TRANSFORMER

- Идея в том, чтобы переиспользовать одну модель на МНОГО разных задач:



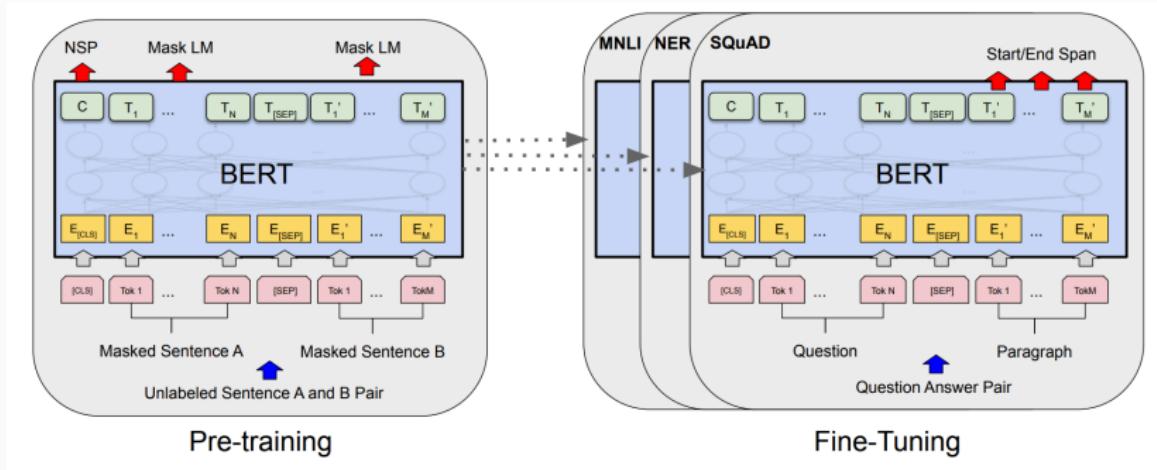
- Получают результаты гораздо лучше, чем были раньше, сразу для нескольких задач.

BERT-СЕМЕЙСТВО

- (Devlin et al., 2018): BERT – Bidirectional Encoder Representations from Transformers
- Фактически тот же Transformer, но теперь двунаправленный, при условии и левого, и правого контекста во всех слоях.
- Т.е. та же языковая модель, но теперь работает с контекстом и слева, и справа.
- Или так не работает? Что делать?..

BERT-СЕМЕЙСТВО

- Просто вместо обычной языковой модели будем маскировать случайные слова и пытаться их предсказывать.
- И вторая задача для pretraining – предсказание следующего предложения.



BERT-СЕМЕЙСТВО

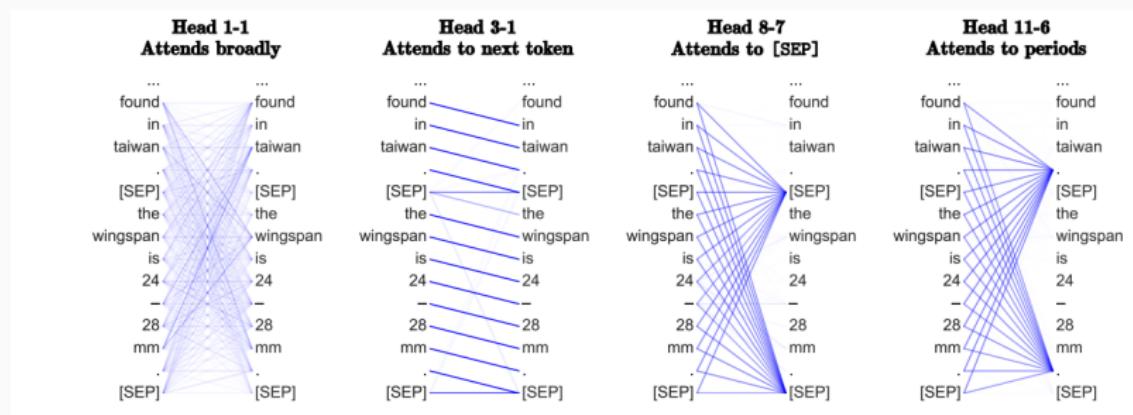
- И всё, в остальном обычный Transformer. Опять побили лучшие результаты для всех задач:

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
BERT _{BASE}	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT _{LARGE}	86.7/85.9	72.1	92.7	94.9	60.5	86.5	89.3	70.1	82.1

- Сейчас BERT – стандартная основа для conversational models и вообще чего угодно в NLP.

BERT-СЕМЕЙСТВО

- Ещё важные детали о BERT:
 - wordpiece embeddings: используем фиксированный словарь под слов разм 30К, а слова делим на части: electrodynamics → electro# #dy# #nami# #cs
 - можно найти «головы», которые смотрят по-разному, дают разный attention:

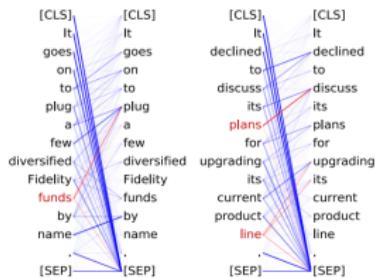


BERT-СЕМЕЙСТВО

- И даже более грамматически (Clark et al., 2019):

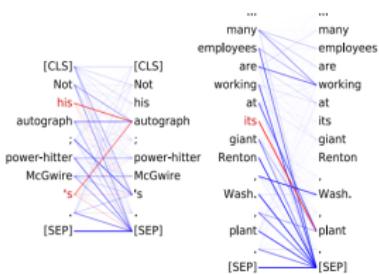
Head 8-10

- Direct objects attend to their verbs
 - 86.8% accuracy at the dobj relation



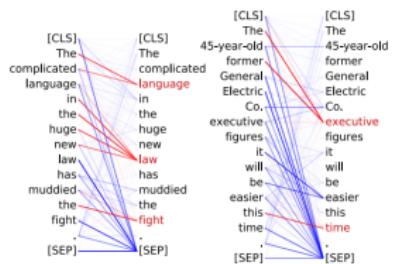
Head 7-6

- Possessive pronouns and apostrophes attend to the head of the corresponding NP
 - 80.5% accuracy at the poss relation



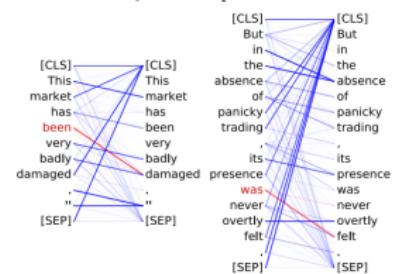
Head 8-11

- **Noun modifiers** (e.g., determiners) attend to their noun
 - 94.3% accuracy at the det relation



Head 4-10

- **Passive auxiliary verbs** attend to the verb they modify
 - 82.5% accuracy at the auxpass relation



BERT-СЕМЕЙСТВО

- Процесс обучения: сначала semi-supervised, потом fine-tuning

1 - Semi-supervised training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.

Semi-supervised Learning Step



Model:

Dataset:

Objective:



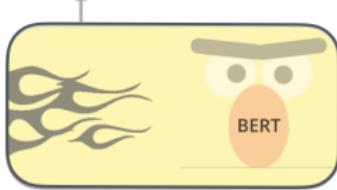
Predict the masked word
(language modeling)

2 - Supervised training on a specific task with a labeled dataset.

Supervised Learning Step



Model:
(pre-trained
in step #1)

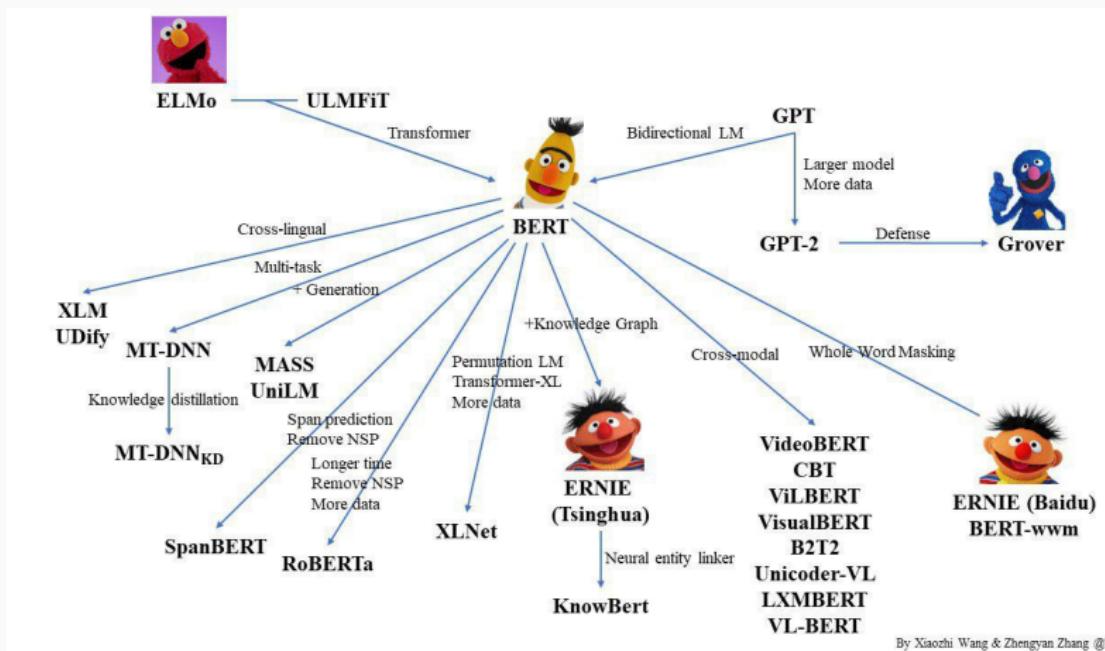


Dataset:

Email message	Class
Buy these pills	Spam
Win cash prizes	Spam
Dear Mr. Atreides, please find attached...	Not Spam

BERT-СЕМЕЙСТВО

- BERT-подобных моделей уже очень много:



BERT-СЕМЕЙСТВО

- ELMo: embeddings с контекстом; ведь у слова МНОГО СМЫСЛОВ:



BERT-СЕМЕЙСТВО

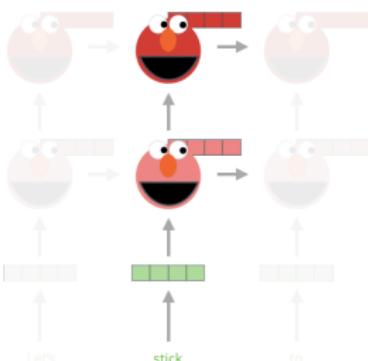
- ELMo берёт линейную комбинацию состояний двунаправленного LSTM с весами, зависящими от конкретной задачи:

Embedding of “stick” in “Let’s stick to” - Step #2

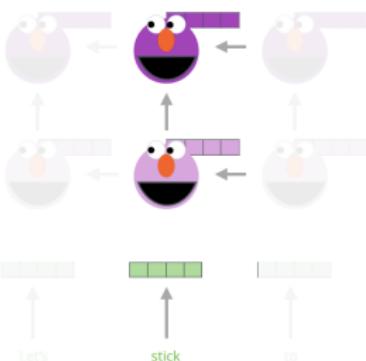
1- Concatenate hidden layers



Forward Language Model



Backward Language Model



2- Multiply each vector by a weight based on the task



3- Sum the (now weighted) vectors

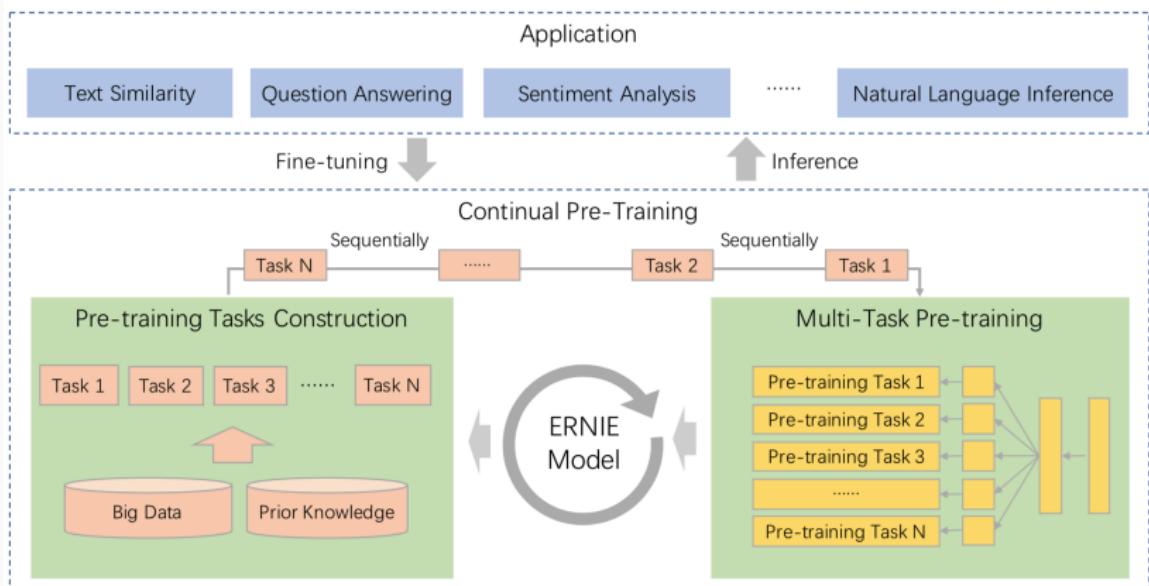


ELMo embedding of “stick” for this task in this context

BERT-СЕМЕЙСТВО

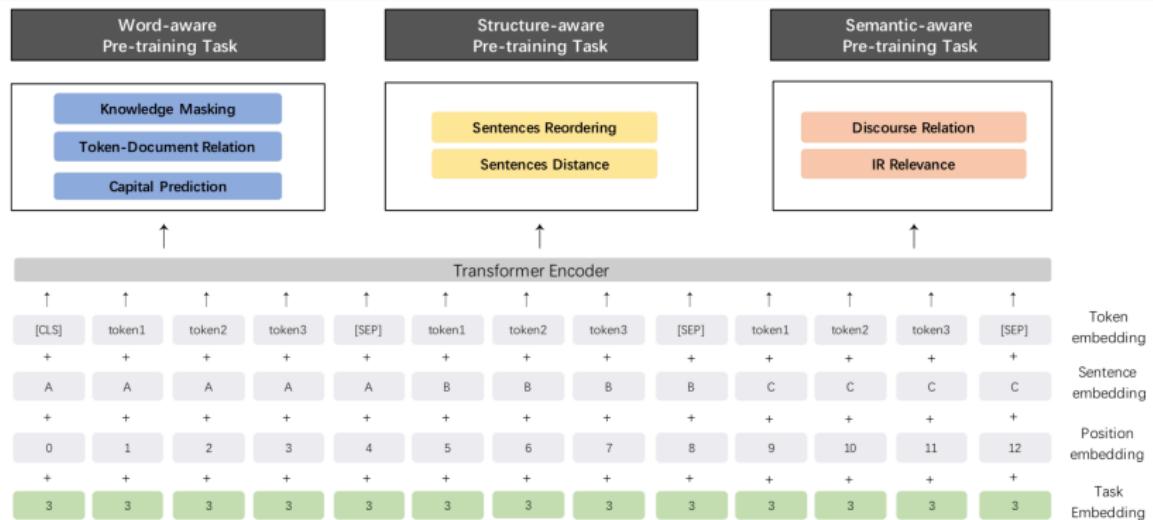
- ERNIE (Zhang et al., 2019) строит пайплайн предобучения на основе разных задач, к которым можно легко породить примеры:

ERNIE 2.0 : A Continual Pre-training framework for Language Understanding



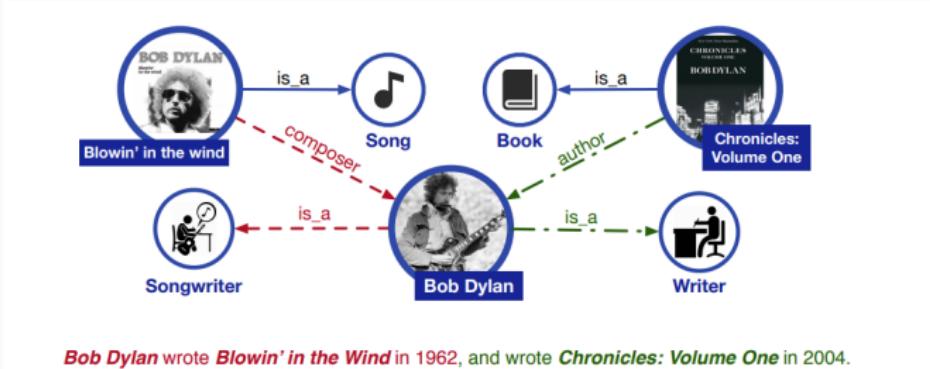
BERT-СЕМЕЙСТВО

- Задачи типа найти в контексте что-то, предсказать капитализацию:



BERT-СЕМЕЙСТВО

- Но не только, ещё добавляем знания из knowledge graph:



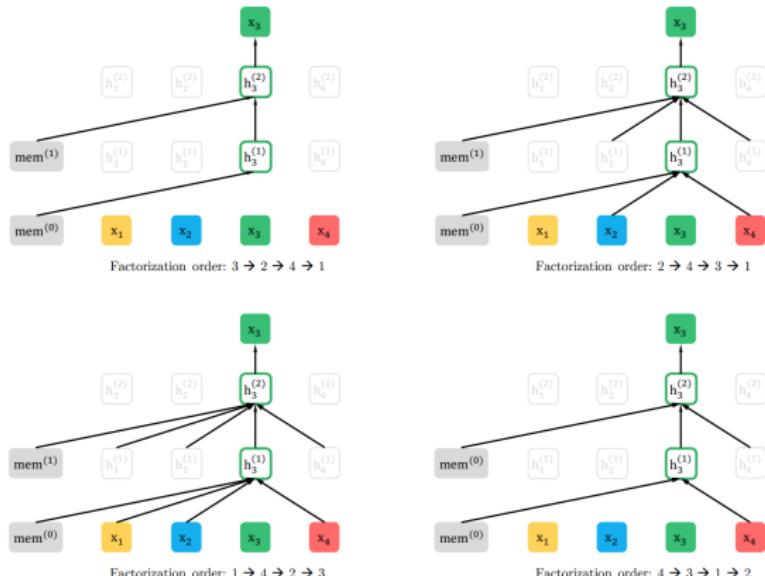
- В реальности это значит, что маски по-умному строятся:

```
- Learned by BERT : [mask] Potter is a series [mask] fantasy novel [mask] by J. [mask] Rowli
```

```
- Learned by ERNIE : Harry Potter is a series of [mask] [mask] written by [mask] [mask] [mask]
```

BERT-СЕМЕЙСТВО

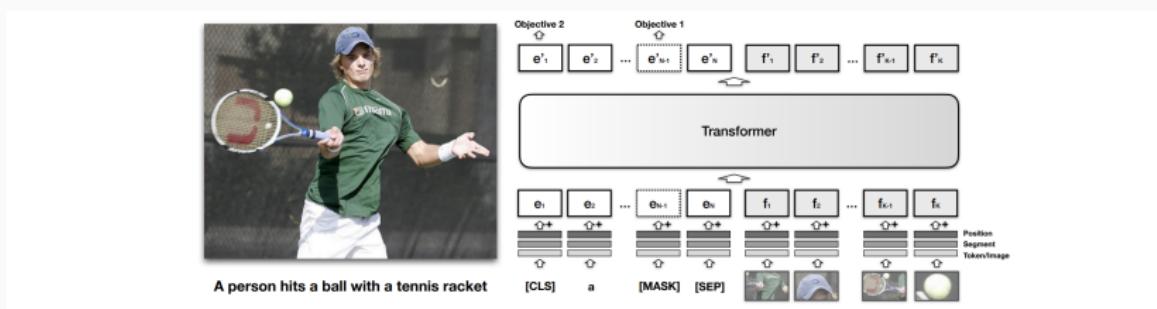
- XLNet (Yang et al., 2019): BERT предсказывает маскированные слова, а XLNet пытается предсказывать данное слово сразу во всех перестановках входного предложения:



- На самом деле сэмплируют при обучении один случайный порядок, но параметры остаются общими

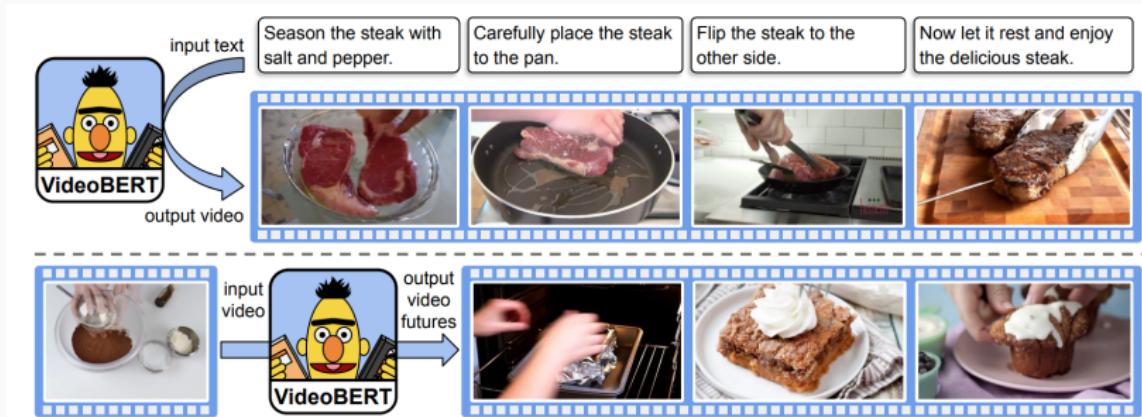
BERT-СЕМЕЙСТВО

- Visual BERT (Li et al., 2019) – Show, Attend, and Tell, только через BERT:



BERT-СЕМЕЙСТВО

- VideoBERT (Sun et al., 2019) – давайте к предложениям ещё видео приложим:



BERT-СЕМЕЙСТВО

- В результате получается, например, video captioning:



GT: add some chopped basil leaves into it

VideoBERT: chop the basil and add to the bowl

S3D: cut the tomatoes into thin slices



GT: cut yu choy into diagonally medium pieces

VideoBERT: chop the cabbage

S3D: cut the roll into thin slices



GT: remove the calamari and set it on paper towel

VideoBERT: fry the squid in the pan

S3D: add the noodles to the pot

BERT-СЕМЕЙСТВО

- Следующая новость – GPT-2 (Radford et al., 2019).
- Это тот же GPT по сути, но:
 - гораздо больше размером: 1.5B параметров (у GPT было 110M, у BERT 340M);
 - обученный на огромном датасете: WebText – все ссылки с Reddit с кармой ≥ 3 , 40GB текста;
 - без всякого fine-tuning и вообще без supervision, проверялось качество в zero-shot контексте.

BERT-СЕМЕЙСТВО

- Результаты:

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	21.8
117M	35.13	45.99	87.65	83.4	29.41	65.85	1.16	1.17	37.50	75.20
345M	15.60	55.48	92.35	87.1	22.76	47.33	1.01	1.06	26.37	55.72
762M	10.87	60.12	93.45	88.0	19.93	40.31	0.97	1.02	22.05	44.575
1542M	8.63	63.24	93.30	89.05	18.34	35.76	0.93	0.98	17.48	42.16

- Конечно, гораздо хуже специализированных моделей, но работает прямо само по себе, из небольшого контекста.
- https://www.reddit.com/user/GPT-2_Bot/
- <https://openai.com/blog/better-language-models/#sample1>

BERT-СЕМЕЙСТВО

- Вопросы (есть на BERT хорошая модель, Alberti et al., 2019):

Question	Generated Answer	Correct	Probability
Who wrote the book the origin of species?	Charles Darwin	✓	83.4%
Who is the founder of the ubuntu project?	Mark Shuttleworth	✓	82.0%
Who is the quarterback for the green bay packers?	Aaron Rodgers	✓	81.1%
Panda is a national animal of which country?	China	✓	76.8%
Who came up with the theory of relativity?	Albert Einstein	✓	76.4%
When was the first star wars film released?	1977	✓	71.4%
What is the most common blood type in sweden?	A	✗	70.6%
Who is regarded as the founder of psychoanalysis?	Sigmund Freud	✓	69.3%
Who took the first steps on the moon in 1969?	Neil Armstrong	✓	66.8%
Who is the largest supermarket chain in the uk?	Tesco	✓	65.3%
What is the meaning of shalom in english?	peace	✓	64.0%
Who was the author of the art of war?	Sun Tzu	✓	59.6%
Largest state in the us by land mass?	California	✗	59.2%
Green algae is an example of which type of reproduction?	parthenogenesis	✗	56.5%
Vikram samvat calender is official in which country?	India	✓	55.6%
Who is mostly responsible for writing the declaration of independence?	Thomas Jefferson	✓	53.3%
What us state forms the western boundary of montana?	Montana	✗	52.3%
Who plays ser davos in game of thrones?	Peter Dinklage	✗	52.1%
Who appoints the chair of the federal reserve system?	Janet Yellen	✗	51.5%
State the process that divides one nucleus into two genetically identical nuclei?	mitosis	✓	50.7%
Who won the most mvp awards in the nba?	Michael Jordan	✗	50.2%
What river is associated with the city of rome?	the Tiber	✓	48.6%
Who is the first president to be impeached?	Andrew Johnson	✓	48.3%
Who is the head of the department of homeland security 2017?	John Kelly	✓	47.0%
What is the name given to the common currency to the european union?	Euro	✓	46.8%
What was the emperor name in star wars?	Palpatine	✓	46.5%
Do you have to have a gun permit to shoot at a range?	No	✓	46.4%
Who proposed evolution in 1859 as the basis of biological development?	Charles Darwin	✓	45.7%
Nuclear power plant that blew up in russia?	Chernobyl	✓	45.7%
Who played john connor in the original terminator?	Arnold Schwarzenegger	✗	45.2%

ПРИМЕР ПОРОЖДЕНИЯ

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.

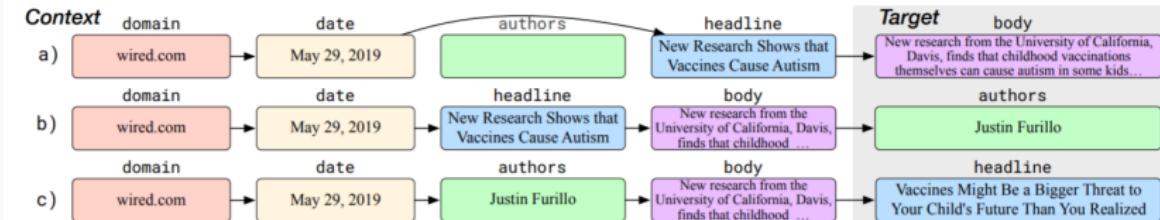
While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Pérez, "In South America, such incidents seem to be quite common."

However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA. "But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization," said the scientist.

GROVER

- (Zellers et al., 2019): GROVER, модель для распознавания и порождения фейковых новостей на основе GPT-2.
- Моделируем как

$$p(\text{domain}, \text{date}, \text{authors}, \text{headline}, \text{body}).$$



- Получается, что машинная пропаганда людьми оценивается лучше, чем человеческая:

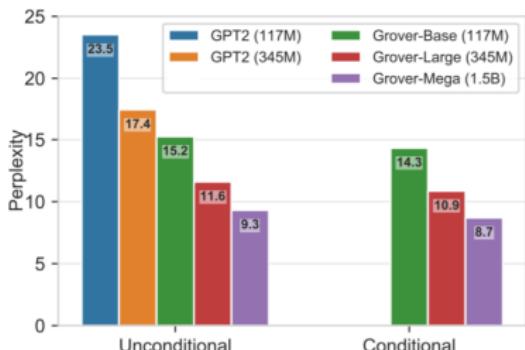


Figure 3: Language Modeling results on the body field of April 2019 articles. We evaluate in the *Unconditional* setting (without provided metadata) as well as in the *Conditional* setting (with all metadata). GROVER sees over a 0.6 point drop in perplexity when given metadata.

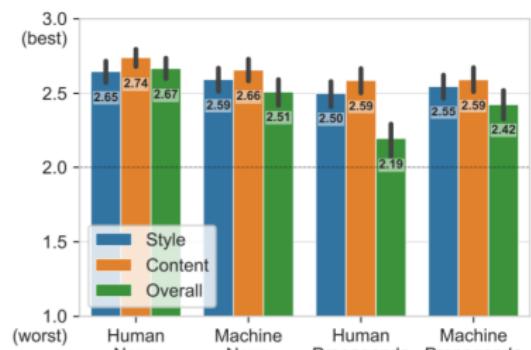


Figure 4: Human evaluation. For each article, three annotators evaluated style, content, and the overall trustworthiness; 100 articles of each category were used. The results show that propaganda generated by GROVER is rated more plausible than the original human-written propaganda.

- Вот наше будущее:

Original Headline: Don't drink the water: The dark side of water fluoridation

Human-written Propaganda

[Don't drink the water: The dark side of water fluoridation](#)

March 13, 2019 [naturalnews.com](#)

(Natural News) There are 7.7 billion people on this planet (as at March 2019). Only about 5 percent of them drink fluoridated water. Why? Because their governments recognize that fluoride in large amounts becomes a toxic chemical that is not fit for human consumption. The 328,000,000 citizens of the United States drink more fluoridated water than all other countries combined. Why? Because the U.S. government continues to doggedly insist that it is safe and improves dental health.

But what do the facts say? As reported by Waking Times, dozens of peer-reviewed studies published in prestigious journals like The Lancet, have confirmed that fluoride is in fact toxic – especially to the developing brains of children. These chemicals are derived from unprocessed toxic waste which is not purified in any way before being pumped into the water supply. How could it possibly be anything but harmful?

The history of water fluoridation in the United States

So, what prompted the government to start adding something so obviously harmful to our precious water supply?

Waking Times, quoting from an article by The Children's Health Defense Team, explains a little about the history of this practice:

During World War II, fluoride (a compound formed from the chemical element fluorine) came into large-scale production and use as part of the Manhattan Project. According to declassified government documents summarized by Project Censored, Manhattan Project scientists discovered early on that fluoride was a "leading health hazard to bomb program workers and surrounding communities." In order to stave off lawsuits, government scientists "embarked on a campaign to calm the social panic about fluoride...by promoting its usefulness in preventing tooth decay."

The power of the elements: Discover Colloidal Silver Mouthwash with quality, natural ingredients like Sangre de Drago sap, black walnut hulls, menthol crystals and more. Zero artificial sweeteners, colors or alcohol. Learn more at the Health Ranger Store and help support this news site.

To back up its decision, the government embarked on a series of flawed and poorly designed "scientific" studies, which an expert later lambasted as "especially rich in fallacies, improper

Machine-written Propaganda

[Fluoride in Your Water Isn't Healthy, Even When It's Recommended](#)

March 13, 2019 [huffingtonpost.com](#)

Fluoride is regarded by researchers around the world as the "gold standard" in tooth care, and a safe, common ingredient that has almost been universally found to be safe in past studies of health effects. It's absorbed without interference from the body's natural minerals.

Researchers at the U.S. Geological Survey (USGS) published the results of a multi-state environmental health study last month. It showed that during the first three decades of fluoridation of tap water systems, fluoride produced from the process alone increased rates of dental caries (the biggest contributor to tooth decay) by 16 percent in Mississippi and a whopping 45 percent in Arizona, which implemented fluoridation systems back in 1942. This increase was seen after a decade when fluoride levels didn't change.

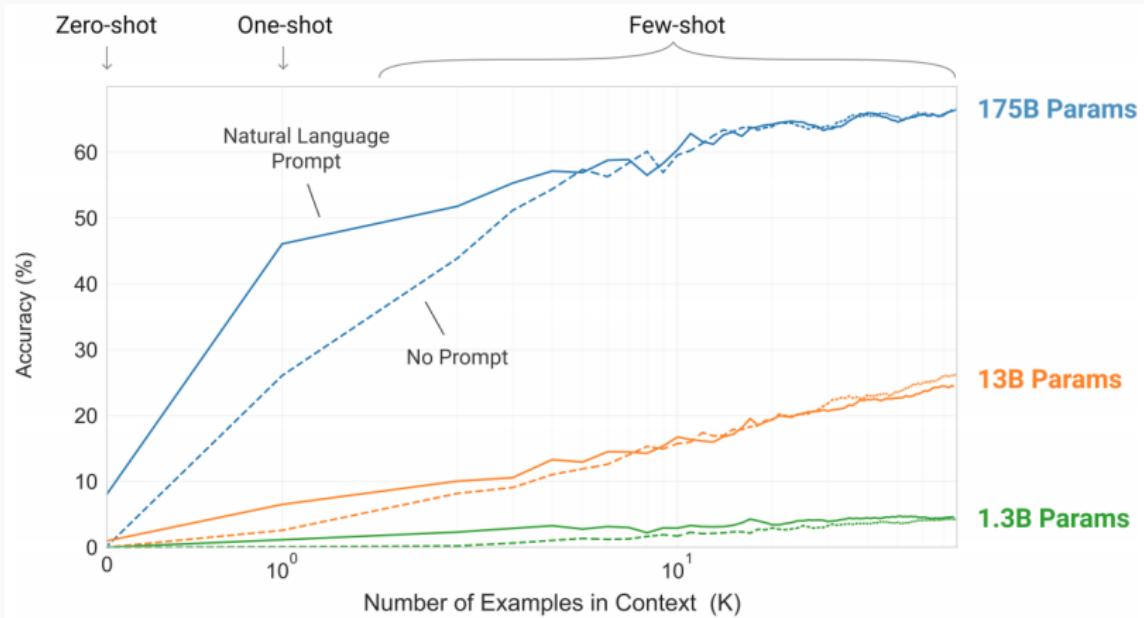
USGS also found that fluoridation increased rates of other toxicants and petrochemicals, as well as deaths from brain, lung, kidney and bladder cancer.

It bears noting that there is no clear proof that these specific contaminants were caused by fluoridation, but the USGS study at least hints that this was the case. The epidemic of brain cancers across the U.S. — especially in teenagers — has confounded researchers for decades. The USGS study points to links to numerous studies that have linked water fluoridation with increased risks of cancer.

Even though the majority of studies on water fluoridation have not produced such alarming results, the mainstream medical community is, apparently, still skeptical. Two years ago, doctors from Harvard and Duke universities suggested that fluoride is associated with lower IQ scores and autoantibodies to water. The results of a recent study that followed more than 700 children over a period of four years demonstrated that the kids were more likely to have symptoms of illness, more likely to have higher blood pressure and sleep problems, had higher mean energy expenditure,

- Ну и, конечно, последние новости – GPT-3 (Brown et al., 2020).
- Это тот же GPT по сути, но:
 - гораздо больше размером: 175B параметров! (у GPT-2 было 1.5B, у самой большой модели на тот момент 17B);
 - обученный на огромном датасете: WebText – все ссылки с Reddit с кармой ≥ 3 , 40GB текста;
 - вместо one-shot и zero-shot learning переходят-таки к few-shot.

- Это помогает, и помогает увеличить контекст:



- Разные типы вспомогательных задач:

The three settings we explore for in-context learning

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



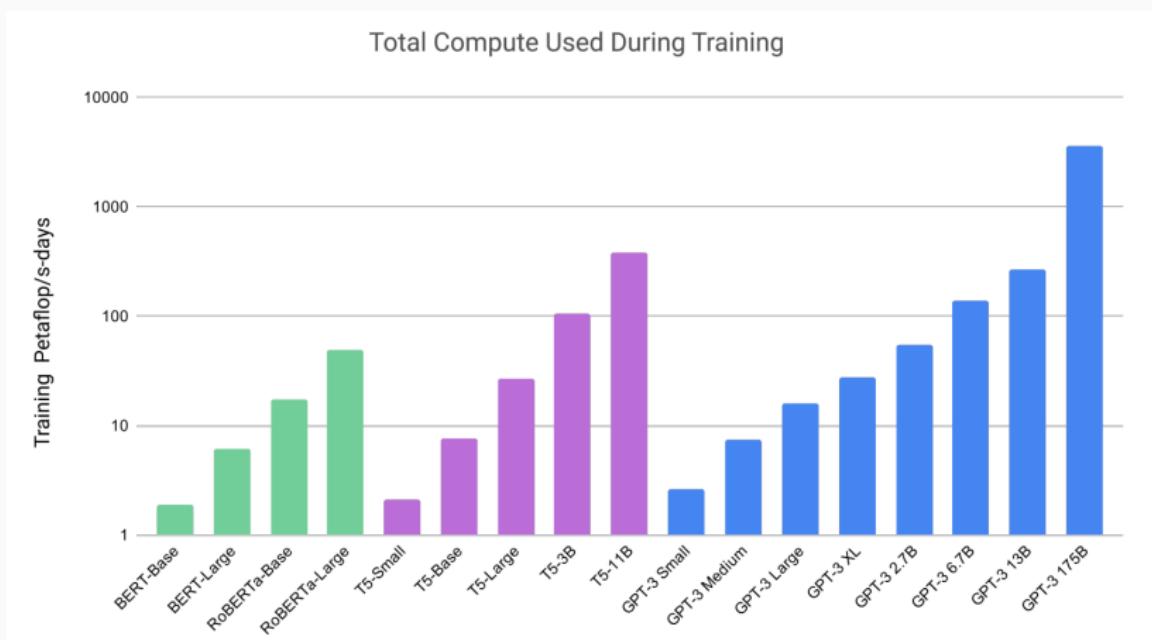
Traditional fine-tuning (not used for GPT-3)

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.



- Очень много compute:



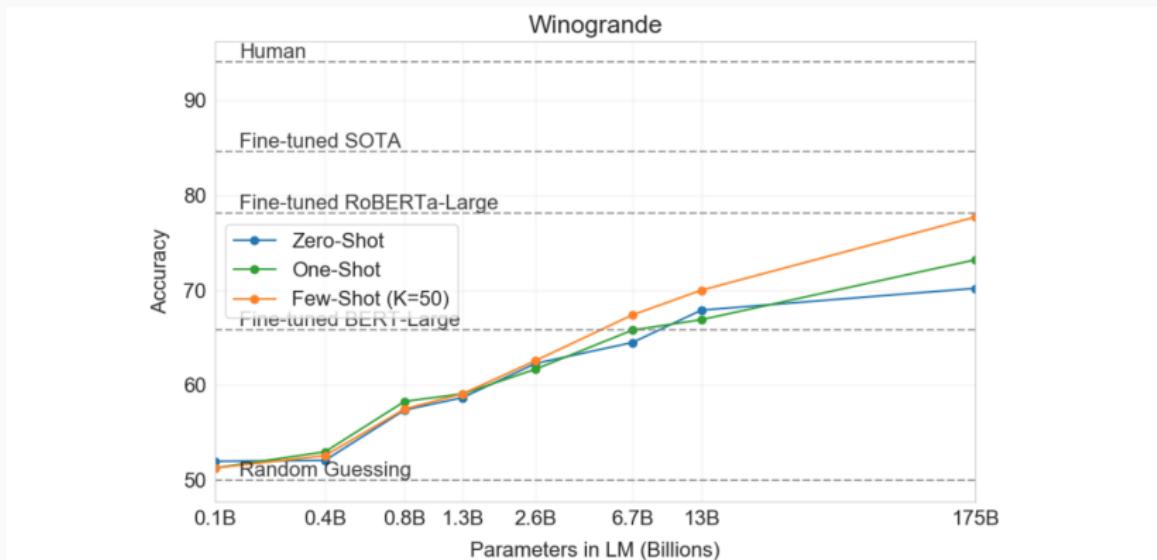
- Очень большие датасеты:

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

- Улучшенные результаты на стандартных задачах (86.4% на вот таких):

Alice was friends with Bob. Alice went to visit her friend _____. → Bob
George bought some baseball equipment, a ball, a glove, and a _____. →

- Лучше результаты на задачах на понимание:



- Примерно таких (это вот и есть Winograd schema):

		Twin sentences	Options (answer)
✓ (1)	a	The trophy doesn't fit into the brown suitcase because <u>it's</u> too <i>large</i> . b The trophy doesn't fit into the brown suitcase because <u>it's</u> too <u>small</u> .	trophy / suitcase trophy / suitcase
✓ (2)	a	Ann asked Mary what time the library closes, <i>because</i> she had forgotten. b Ann asked Mary what time the library closes, <u>but</u> she had forgotten.	Ann / Mary Ann / Mary
✗ (3)	a	The tree fell down and crashed through the roof of my house. Now, I have to get <u>it</u> <i>removed</i> . b The tree fell down and crashed through the roof of my house. Now, I have to get <u>it</u> <i>repaired</i> .	tree / roof tree / roof
✗ (4)	a	The lions ate the zebras because <u>they</u> are <i>predators</i> . b The lions ate the zebras because <u>they</u> are <i>meaty</i> .	lions / zebras lions / zebras

Table 1: WSC problems are constructed as pairs (called *twin*) of nearly identical questions with two answer choices. The questions include a *trigger word* that flips the correct answer choice between the questions. Examples (1)-(3) are drawn from WSC (Levesque, Davis, and Morgenstern 2011) and (4) from DPR (Rahman and Ng 2012)). Examples marked with ✗ have language-based bias that current language models can easily detect. Example (4) is undesirable since the word “predators” is more often associated with the word “lions”, compared to “zebras”

- Или таких:

Context → Title: The Blitz

Background: From the German point of view, March 1941 saw an improvement. The Luftwaffe flew 4,000 sorties that month, including 12 major and three heavy attacks. The electronic war intensified but the Luftwaffe flew major inland missions only on moonlit nights. Ports were easier to find and made better targets. To confuse the British, radio silence was observed until the bombs fell. X- and Y-Gerät beams were placed over false targets and switched only at the last minute. Rapid frequency changes were introduced for X-Gerät, whose wider band of frequencies and greater tactical flexibility ensured it remained effective at a time when British selective jamming was degrading the effectiveness of Y-Gerät.

Q: How many sorties were flown in March 1941?

A: 4,000

Q: When did the Luftwaffe fly inland missions?

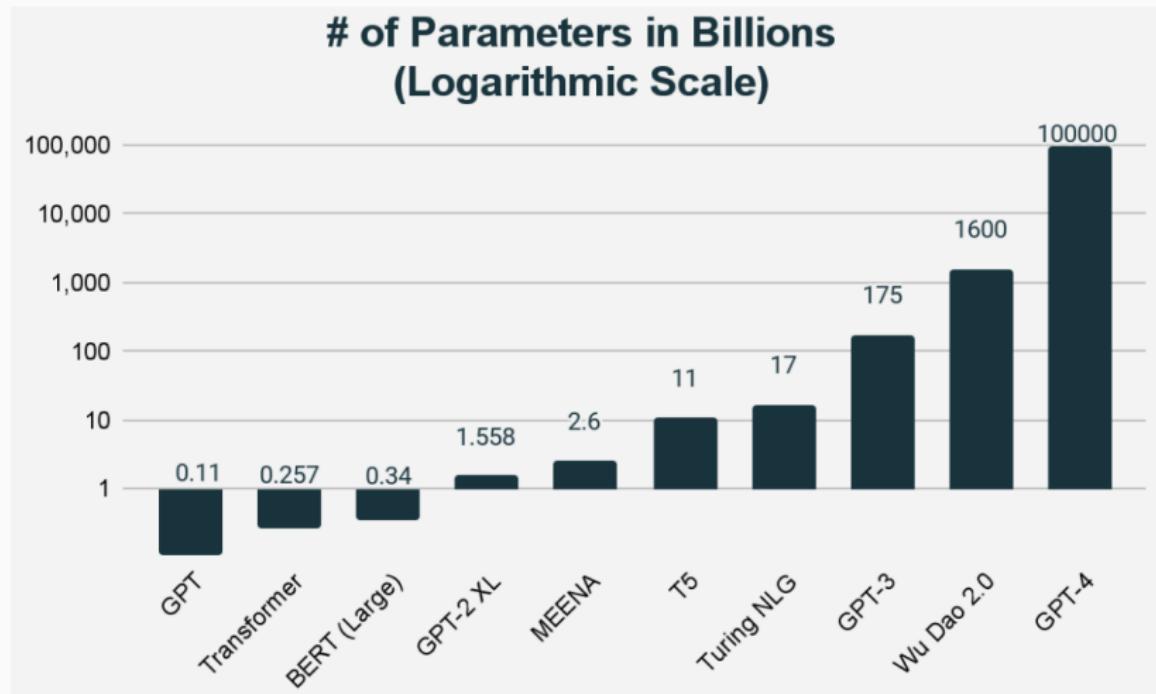
A:

Target Completion → only on moonlit nights

Figure G.28: Formatted dataset example for SQuADv2

А ТЕПЕРЬ...

- Модели растут, и конца-края этому не видно:

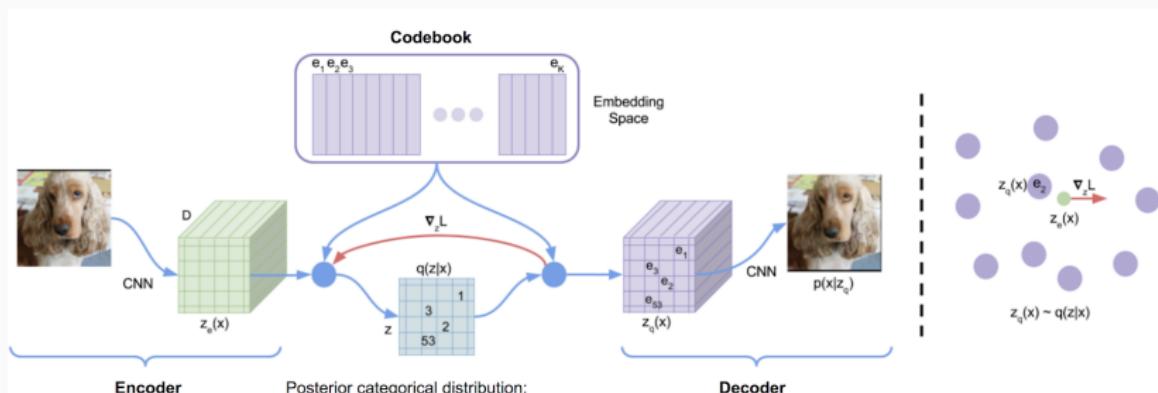
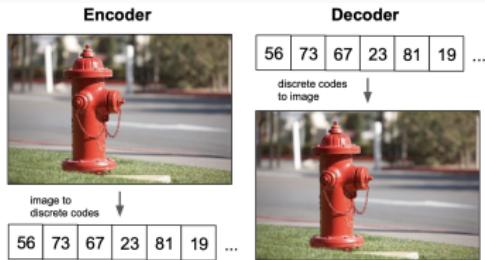


- Кстати, это отдельный разговор.

DALL-E

DVAE + TRANSFORMER = DALL-E

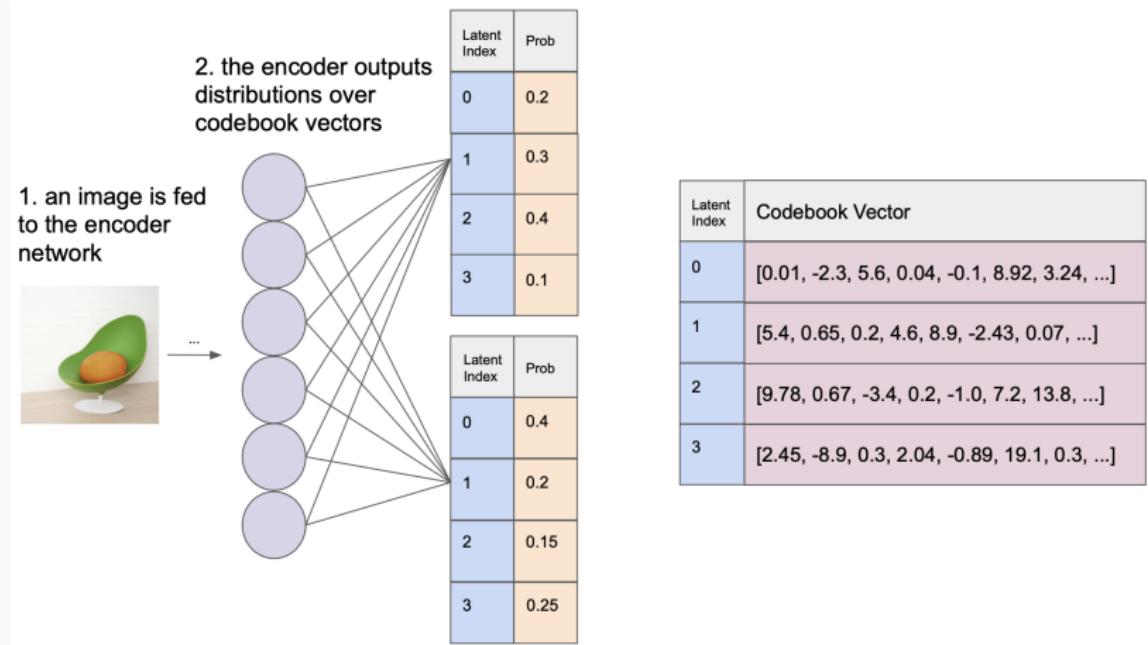
- Мы уже обсуждали VQ-VAE:



$$q(\mathbf{z} = \mathbf{e}_k | \mathbf{x}) = \begin{cases} 1 & \text{if } k = \arg \min_i \|\mathbf{z}_e(\mathbf{x}) - \mathbf{e}_i\|_2 \\ 0 & \text{otherwise.} \end{cases}$$

dVAE + TRANSFORMER = DALL-E

- dVAE – это VQ-VAE, который выдаёт не один вектор из словаря, а распределение на словаре:

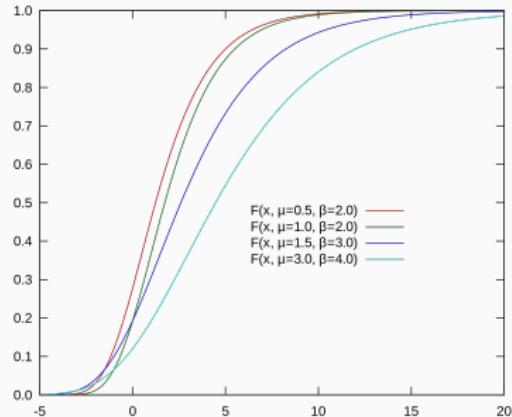
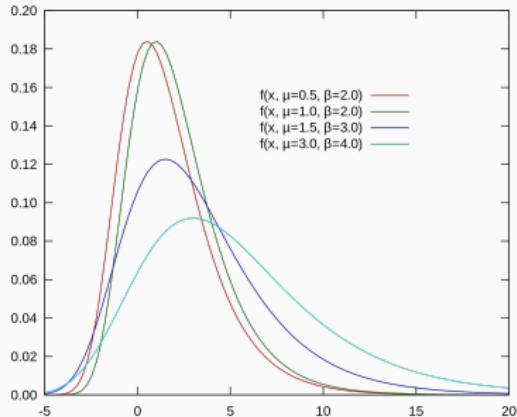


- Как протащить градиент через сэмплирование?

dVAE + TRANSFORMER = DALL-E

- dVAE считает градиенты не так, как VQ-VAE; для дискретизации служит *Gumbel-Softmax distribution*:
- Gumbel-Max trick: можно сэмплировать из дискретного распределения с вероятностями классов π_i как $z = \arg \max_i (g_i + \log \pi_i)$, где g_i берутся из распределения Гумбеля:

$$p(g_i) = e^{-(g_i + e^{-g_i})}, \quad F(g_i) = e^{-e^{-g_i}}.$$



DVAE + TRANSFORMER = DALL-E

- Объяснение трюка:

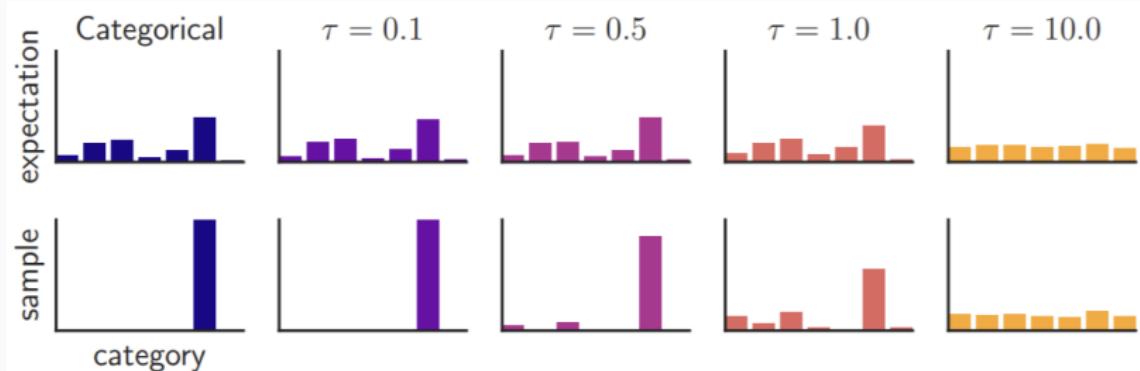
$$\begin{aligned} p(z = k) &= p(g_k + \log \pi_k \geq g_j + \log \pi_j \text{ для всех } j) = \\ &= \int_{-\infty}^{\infty} \prod_{j \neq k} p(g_k + \log \pi_k \geq g_j + \log \pi_j \mid g_k) p(g_k) dg_k = \\ &= \int_{-\infty}^{\infty} \prod_{j \neq k} e^{-e^{-g_k - \log \pi_k + \log \pi_j}} e^{-(g_k + e^{-g_k})} dg_k = \\ &= \int_{-\infty}^{\infty} e^{-\sum_{j \neq k} \pi_j e^{-g_k - \log \pi_k}} \pi_k e^{-(g_k + \log \pi_k + \pi_k e^{-g_k - \log \pi_k})} dg_k = \\ &= \pi_k \int_{-\infty}^{\infty} e^{-g_k - \log \pi_k - (\pi_k + \sum_{j \neq k} \pi_j) e^{-g_k - \log \pi_k}} dg_k = \pi_k. \end{aligned}$$

dVAE + TRANSFORMER = DALL-E

- A Gumbel-Softmax — это релаксация дискретного распределения:

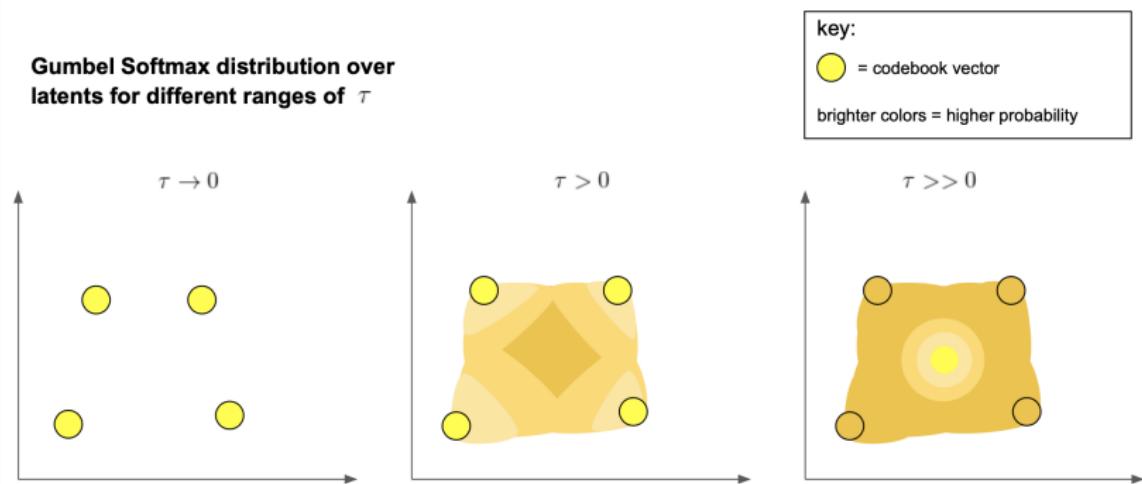
$$y_i = \text{softmax} \left(\frac{g_i + \log \pi_i}{\tau} \right) = \frac{e^{\frac{1}{\tau}(g_i + \log \pi_i)}}{\sum_j e^{\frac{1}{\tau}(g_j + \log \pi_j)}},$$

и при $\tau \rightarrow 0$ это стремится к дискретному распределению с вероятностями π_i .



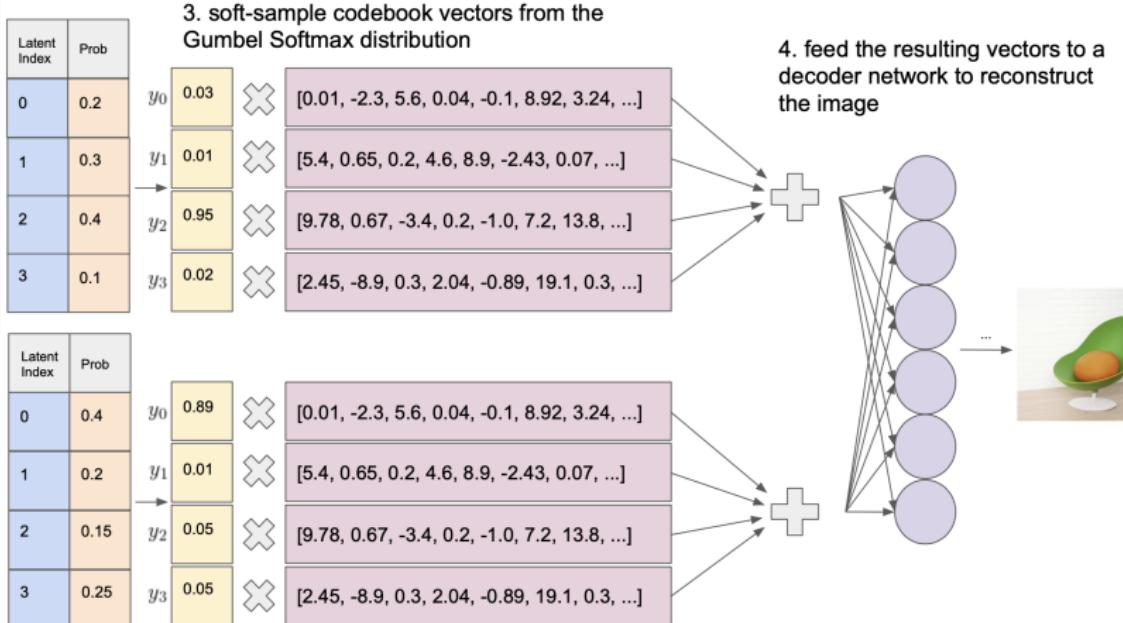
DVAE + TRANSFORMER = DALL-E

- Это то, что мы делаем с кодовыми векторами — сэмплируем выпуклую комбинацию $\mathbf{z} = \sum_{j=1}^k y_j \mathbf{e}_j$ через Gumbel-Softmax (это тоже, кстати, reparametrization trick: теперь g_i сэмплируются отдельно), и теперь все градиенты проходят, а во время обучения потихоньку устремляем $\tau \rightarrow 0$:



DVAE + TRANSFORMER = DALL-E

- Общая картина:

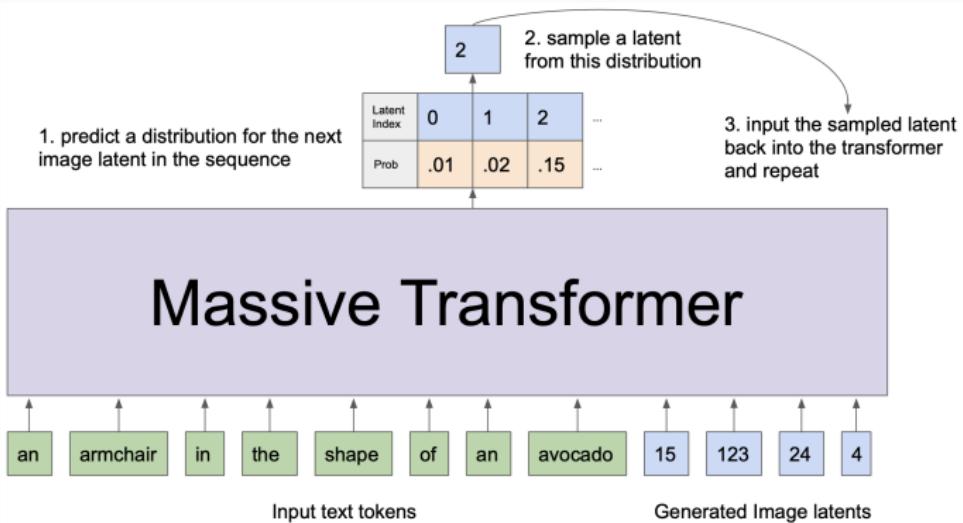


dVAE + TRANSFORMER = DALL-E

- И теперь DALL-E моделирует код **z** из текста **y**, а потом использует его в dVAE, чтобы получить картинку **x**:

$$p_{\theta, \psi}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p_{\theta}(\mathbf{x}|\mathbf{z}, \mathbf{y}) p_{\psi}(\mathbf{y}, \mathbf{z}).$$

- Трансформер порождает только **z**, как последовательность:



dVAE + TRANSFORMER = DALL-E

- То есть фактически это одна большая вариационная оценка:

$$\ln p_{\theta, \psi}(\mathbf{x}, \mathbf{y}) \geq \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\ln p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z}) - \beta \text{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\psi(\mathbf{y}, \mathbf{z}))] :$$

- $q_\phi(\mathbf{z}|\mathbf{x})$ — распределение над 32×32 токенами, порождённое энкодером dVAE по картинке \mathbf{x} ;
- $p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z})$ — распределение на картинках, порождённое декодером dVAE по токенам \mathbf{z} ; здесь мы предположим $p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})$;
- $p_\psi(\mathbf{y}, \mathbf{z})$ — совместное распределение на текстах и латентных кодах, которое моделирует трансформер.

dVAE + TRANSFORMER = DALL-E

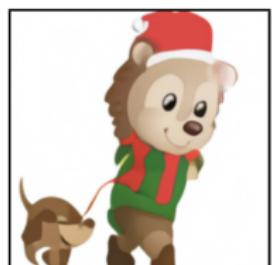
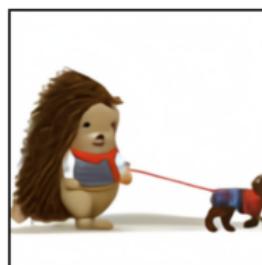
- То есть фактически это одна большая вариационная оценка:

$$\ln p_{\theta, \psi}(\mathbf{x}, \mathbf{y}) \geq \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\ln p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z}) - \beta \text{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\psi(\mathbf{y}, \mathbf{z}))].$$

- И обучение идёт в два приёма:
 - сначала максимизируем оценку по ϕ и θ , т.е. обучаем dVAE просто на картинках, без текстов; здесь считаем p_ψ равномерным, а q_ϕ релаксируем через Gumbel-Softmax;
 - потом фиксируем ϕ и θ и обучаем ψ , т.е. обучаем трансформер моделировать текст (BPE encoding) и картинки (их коды \mathbf{z}) совместно.
- И дальше очень, очень много трюков с распределённым обучением, потому что модели очень большие.

DVAE + TRANSFORMER = DALL-E

- Получается очень круто, и с выдумкой:



(a) a tapir made of accordion.
a tapir with the texture of an
accordion.

(b) an illustration of a baby
hedgehog in a christmas
sweater walking a dog

DVAE + TRANSFORMER = DALL-E

- Получается очень круто, и с выдумкой:

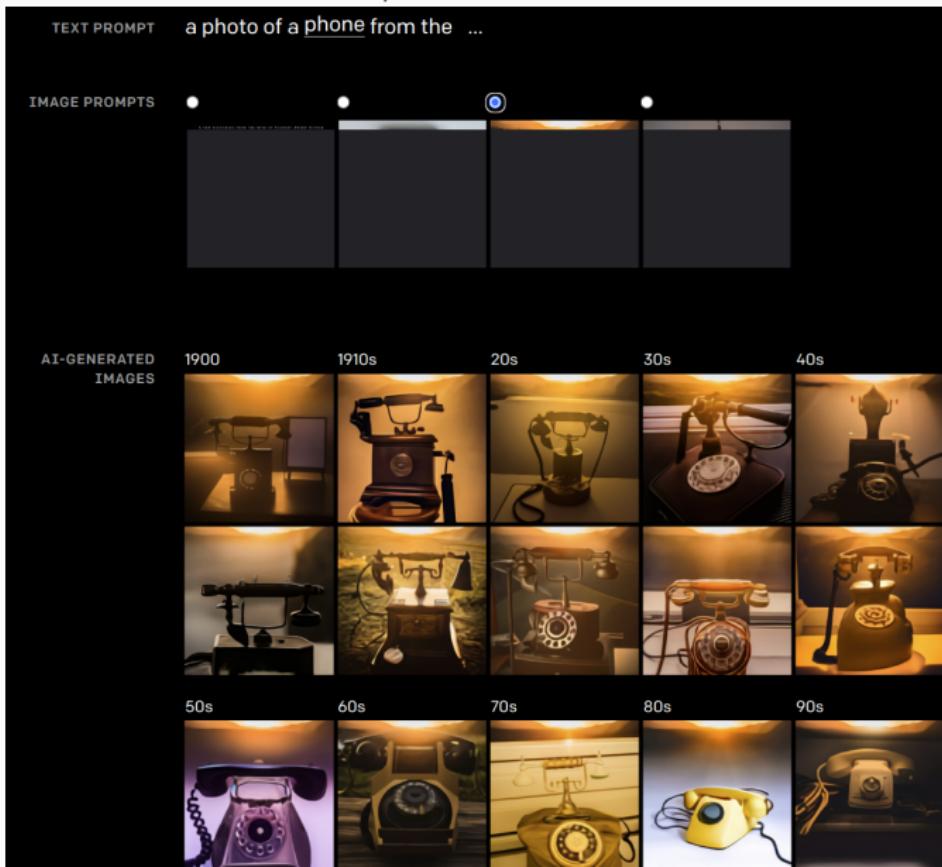


(c) a neon sign that reads “backprop”. a neon sign that reads “backprop”. backprop neon sign

(d) the exact same cat on the top as a sketch on the bottom

DVAE + TRANSFORMER = DALL-E

- В том числе сложные запросы (<https://openai.com/blog/dall-e/>):



Спасибо!

Спасибо за внимание!

