ELECTRA: PRE-TRAINING TEXT ENCODERS AS DISCRIMINATORS RATHER THAN GENERATORS

Kevin Clark

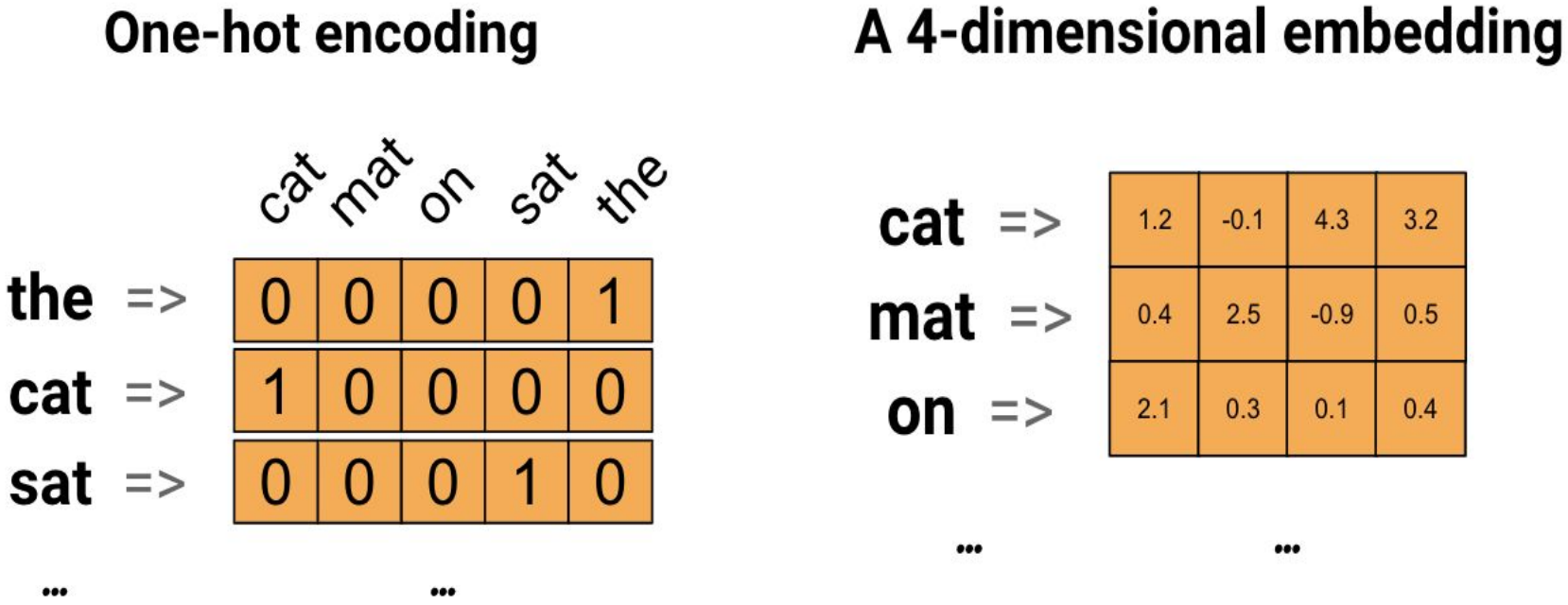
Minh-Thang Luong

Quoc V. Le

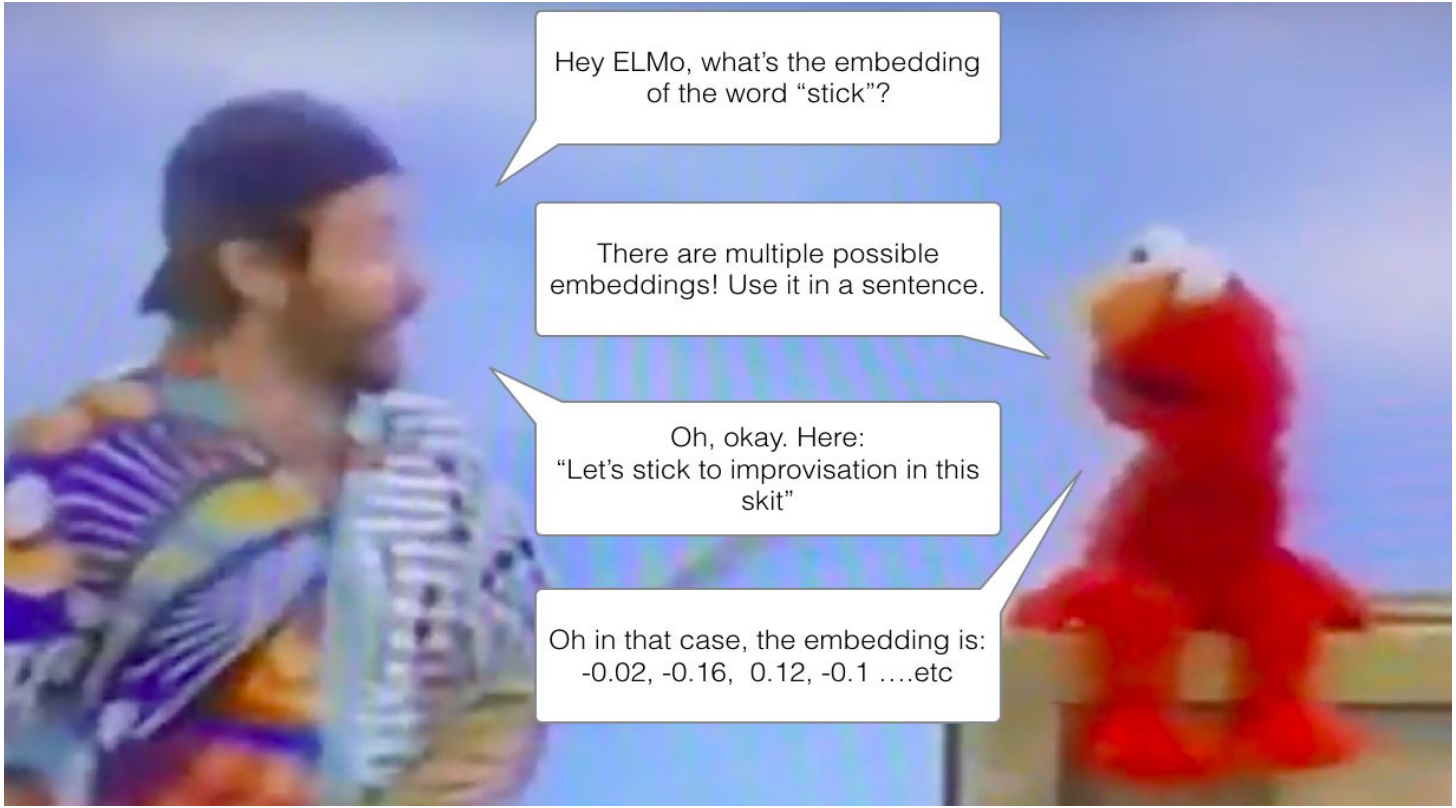
Christopher D. Manning

Антюх Михаил 172

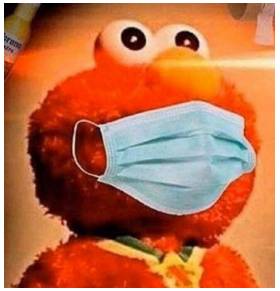
Обработка естественного языка



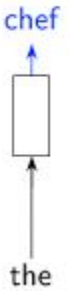
Обработка естественного языка



Как насчет масок?



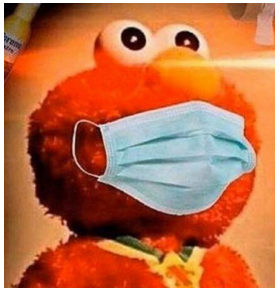
Language Modeling



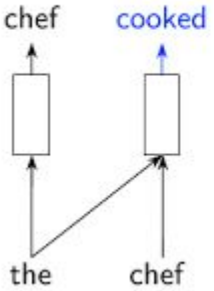
Masked Language Modeling

the chef cooked the meal

Как насчет масок?



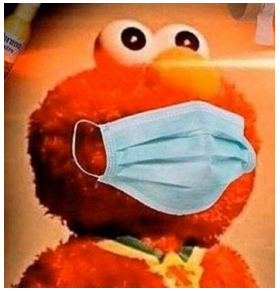
Language Modeling

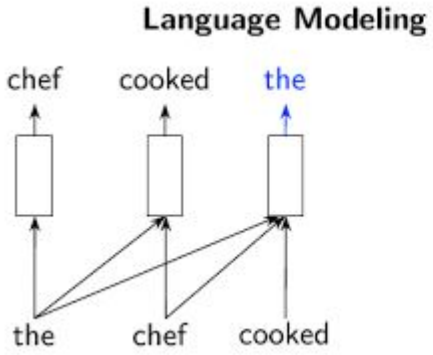


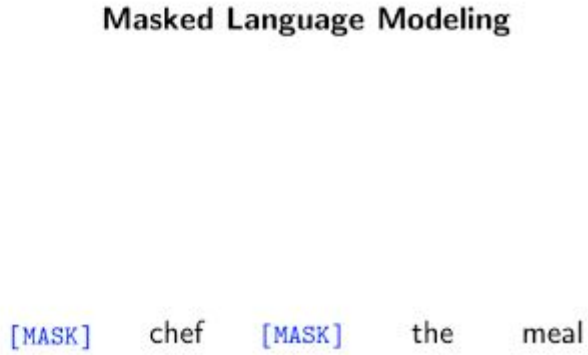
Masked Language Modeling

the chef cooked the meal

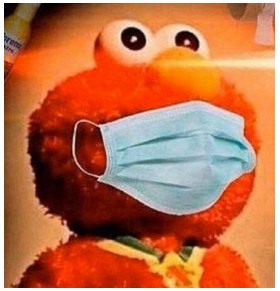
Как насчет масок?

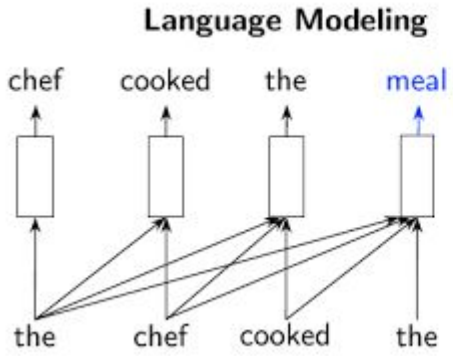


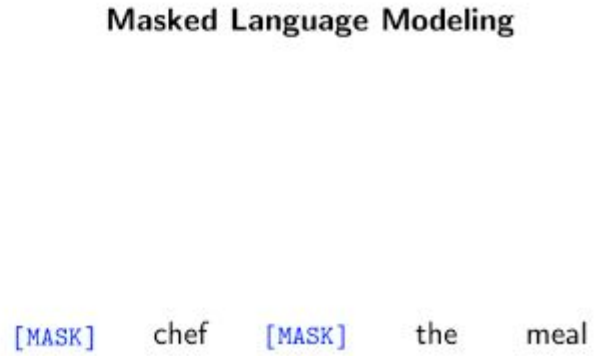




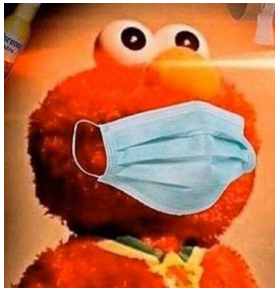
Как насчет масок?

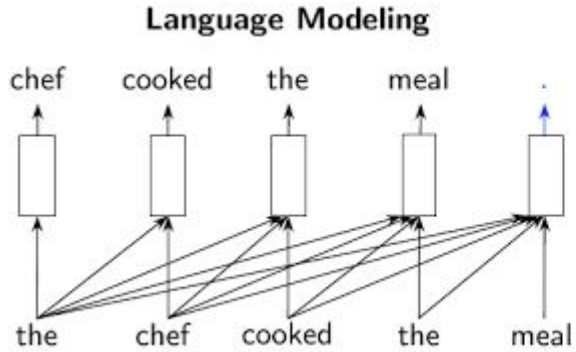


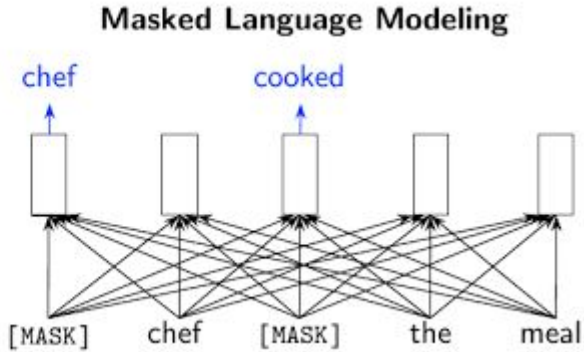




Как насчет масок?







ELECTRA

Основная идея: заменить токены на правдоподобные

Replaced Token Detection

the chef cooked the meal

ELECTRA

Основная идея: заменить токены на правдоподобные

Replaced Token Detection

~~the~~ chef ~~cooked~~ the meal

ELECTRA

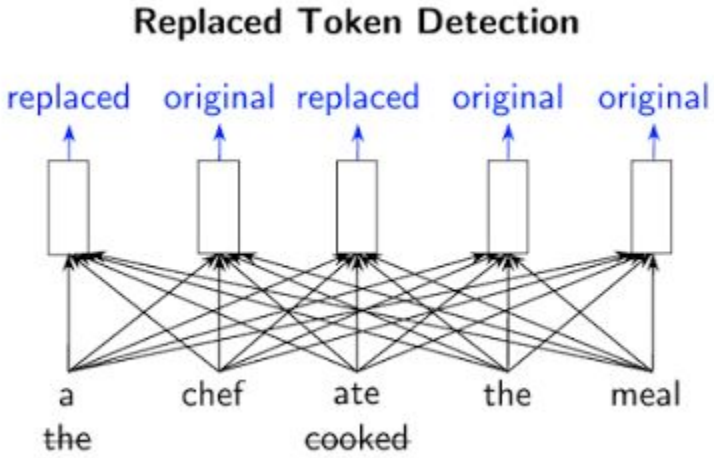
Основная идея: заменить токены на правдоподобные

Replaced Token Detection

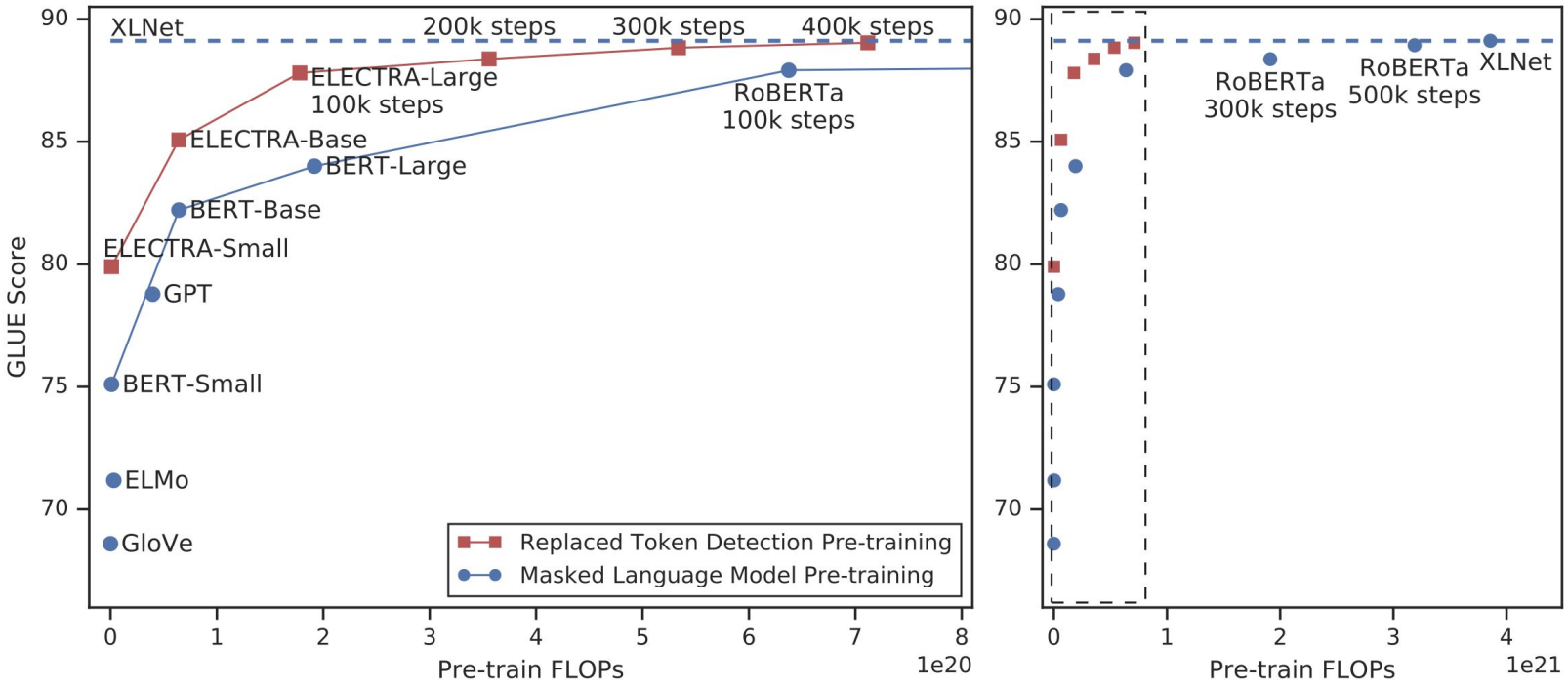
a	chef	ate	the	meal
~~the~~		~~cooked~~		

ELECTRA

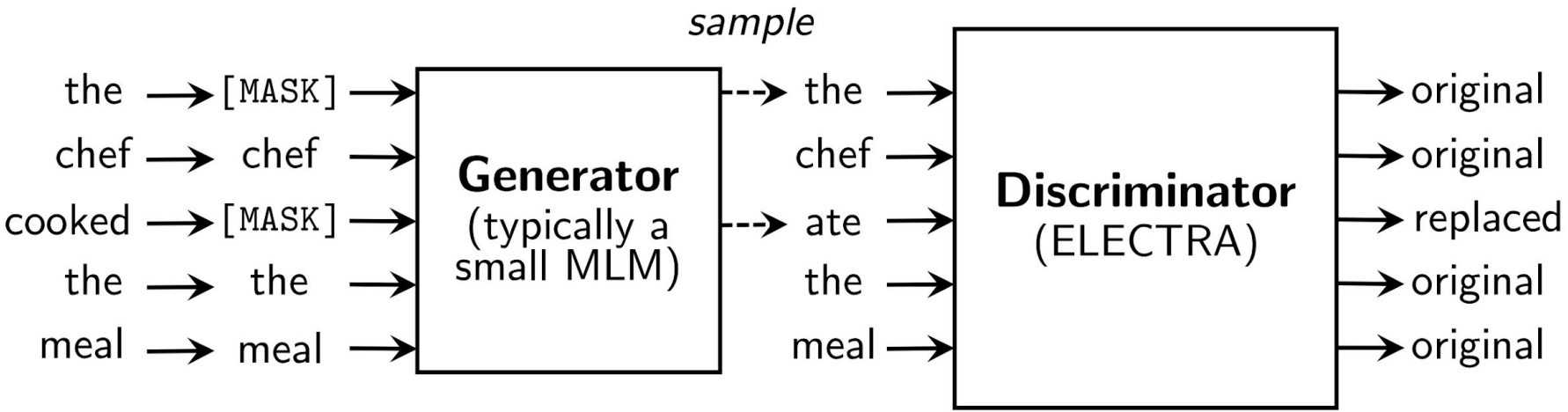
Основная идея: заменить токены на правдоподобные



ELECTRA VS OTHERS



Архитектура



Генератор

Вход: последовательность токенов x = [x_1,..., x_n] и позиция t, для которой верно, x_t = MASK.

Выход: вероятность генерации определенного токена x_t:

$$p_G(x_t|\boldsymbol{x}) = \exp\left(e(x_t)^T h_G(\boldsymbol{x})_t\right) / \sum_{x'} \exp\left(e(x')^T h_G(\boldsymbol{x})_t\right)$$

h(x) = [h_1,..., h_n] - контекстуализированные эмбединги из hidden layer,

e(x_t) - эмбединг токена.

Дискриминатор

Вход: последовательность токенов x = [x_1,..., x_n] и позиция t, для которой верно, x_t = CORRUPT.

Выход: оригинал / замена.

$$D(\boldsymbol{x}, t) = \text{sigmoid}(w^T h_D(\boldsymbol{x})_t)$$

Шаги обучения

1) $$m_i \sim \text{unif}\{1, n\} \text{ for } i = 1 \text{ to } k$$

2) $$\boldsymbol{x}^{\text{masked}} = \text{REPLACE}(\boldsymbol{x}, \boldsymbol{m}, \texttt{[MASK]})$$

3) $$\hat{x}_i \sim p_G(x_i | \boldsymbol{x}^{\text{masked}}) \text{ for } i \in \boldsymbol{m}$$

4) $$\boldsymbol{x}^{\text{corrupt}} = \text{REPLACE}(\boldsymbol{x}, \boldsymbol{m}, \hat{\boldsymbol{x}})$$

Функции потерь

$$\mathcal{L}_{\text{MLM}}(\boldsymbol{x}, \theta_G) = \mathbb{E}\left(\sum_{i \in \boldsymbol{m}} -\log p_G(x_i | \boldsymbol{x}^{\text{masked}})\right)$$

$$\mathcal{L}_{\text{Disc}}(\boldsymbol{x}, \theta_D) = \mathbb{E}\left(\sum_{t=1}^{n} -\mathbb{1}(x_t^{\text{corrupt}} = x_t) \log D(\boldsymbol{x}^{\text{corrupt}}, t) - \mathbb{1}(x_t^{\text{corrupt}} \neq x_t) \log(1 - D(\boldsymbol{x}^{\text{corrupt}}, t))\right)$$

Итоговая функция потерь: $$\min_{\theta_G, \theta_D} \sum_{\boldsymbol{x} \in \mathcal{X}} \mathcal{L}_{\text{MLM}}(\boldsymbol{x}, \theta_G) + \lambda \mathcal{L}_{\text{Disc}}(\boldsymbol{x}, \theta_D)$$

$\mathcal{X}$ - текстовый корпус

Отличие от GAN

- Генератор ELECETRA может генерировать реальные токены.

- Генератор ELECETRA максимизирует правдоподобие, а в GAN генератор и дискриминатор соревновательные.

Состязательное обучение генератора является сложной задачей, поскольку невозможно выполнить обратное распространение.

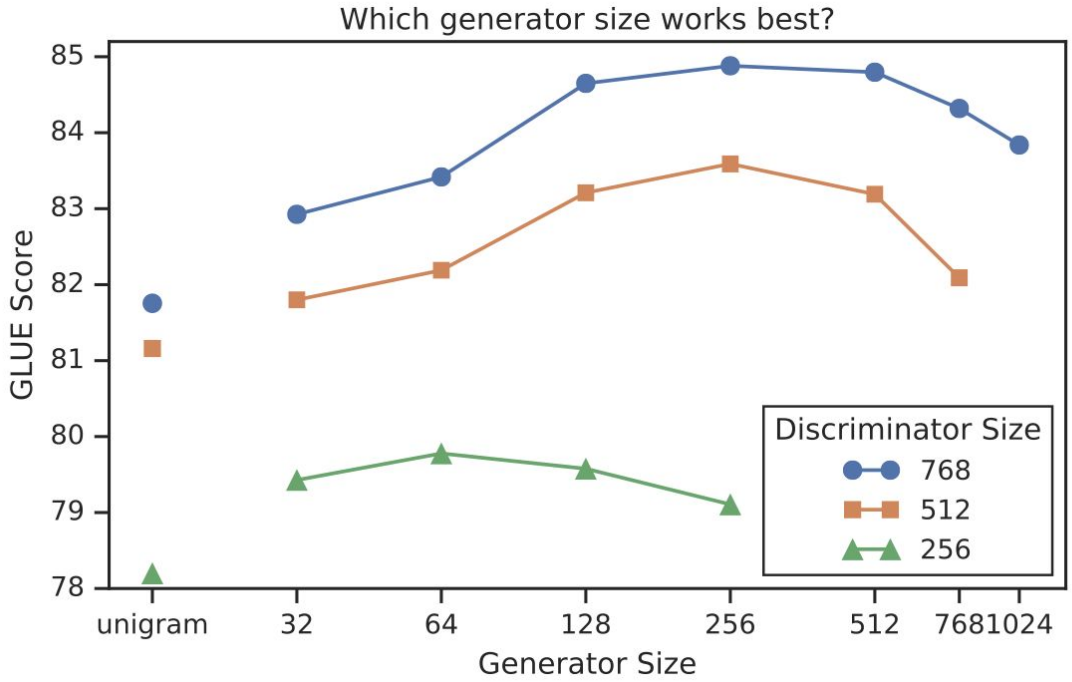
Эксперименты

GLUE - оценивает системы понимания естественного языка

SQuAD - Stanford Question Answering

FLOPS - количество операций с плавающей запятой в секунду

Эксперименты: размер генератора и дискриминатора



Оптимальный размер генератора: 1/4-1/2 размера дискриминатора

Эксперименты: маленькие модели

Model	Train / Infer FLOPs	Speedup	Params	Train Time + Hardware	GLUE
ELMo	3.3e18 / 2.6e10	19x / 1.2x	96M	14d on 3 GTX 1080 GPUs	71.2
GPT	4.0e19 / 3.0e10	1.6x / 0.97x	117M	25d on 8 P6000 GPUs	78.8
BERT-Small	1.4e18 / 3.7e9	45x / 8x	14M	4d on 1 V100 GPU	75.1
BERT-Base	6.4e19 / 2.9e10	1x / 1x	110M	4d on 16 TPUv3s	82.2
ELECTRA-Small	1.4e18 / 3.7e9	45x / 8x	14M	4d on 1 V100 GPU	79.9
50% trained	7.1e17 / 3.7e9	90x / 8x	14M	2d on 1 V100 GPU	79.0
25% trained	3.6e17 / 3.7e9	181x / 8x	14M	1d on 1 V100 GPU	77.7
12.5% trained	1.8e17 / 3.7e9	361x / 8x	14M	12h on 1 V100 GPU	76.0
6.25% trained	8.9e16 / 3.7e9	722x / 8x	14M	6h on 1 V100 GPU	74.1
ELECTRA-Base	6.4e19 / 2.9e10	1x / 1x	110M	4d on 16 TPUv3s	85.1

Эксперименты: большие модели

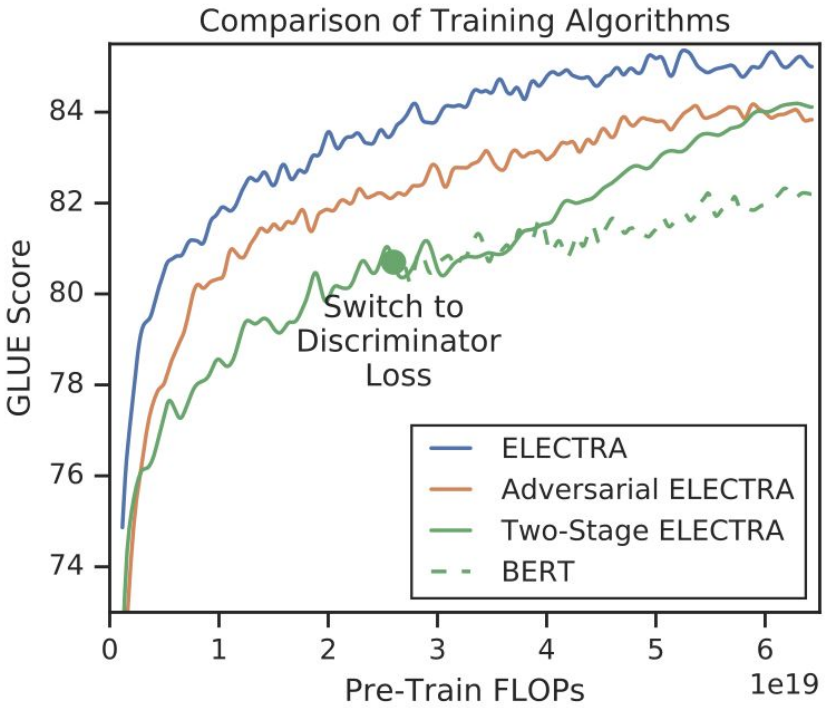
Model	Train FLOPs	Params	CoLA	SST	MRPC	STS	QQP	MNLI	QNLI	RTE	Avg.
BERT	1.9e20 (0.27x)	335M	60.6	93.2	88.0	90.0	91.3	86.6	92.3	70.4	84.0
RoBERTa-100K	6.4e20 (0.90x)	356M	66.1	95.6	**91.4**	92.2	92.0	89.3	94.0	82.7	87.9
RoBERTa-500K	3.2e21 (4.5x)	356M	68.0	96.4	90.9	92.1	92.2	90.2	94.7	86.6	88.9
XLNet	3.9e21 (5.4x)	360M	69.0	**97.0**	90.8	92.2	92.3	90.8	94.9	85.9	89.1
BERT (ours)	7.1e20 (1x)	335M	67.0	95.9	89.1	91.2	91.5	89.6	93.5	79.5	87.2
ELECTRA-400K	7.1e20 (1x)	335M	**69.3**	96.0	90.6	92.1	**92.4**	90.5	94.5	86.8	89.0
ELECTRA-1.75M	3.1e21 (4.4x)	335M	69.1	96.9	90.8	**92.6**	**92.4**	**90.9**	**95.0**	**88.0**	**89.5**

Model	Train FLOPs	CoLA	SST	MRPC	STS	QQP	MNLI	QNLI	RTE	WNLI	Avg.*	Score
BERT	1.9e20 (0.06x)	60.5	94.9	85.4	86.5	89.3	86.7	92.7	70.1	65.1	79.8	80.5
RoBERTa	3.2e21 (1.02x)	67.8	96.7	89.8	91.9	90.2	90.8	95.4	88.2	89.0	88.1	88.1
ALBERT	3.1e22 (10x)	69.1	**97.1**	**91.2**	92.0	90.5	**91.3**	–	89.2	91.8	89.0	–
XLNet	3.9e21 (1.26x)	70.2	**97.1**	90.5	**92.6**	90.4	90.9	–	88.5	**92.5**	89.1	–
ELECTRA	3.1e21 (1x)	**71.7**	**97.1**	90.7	92.5	**90.8**	**91.3**	**95.8**	**89.8**	**92.5**	**89.5**	**89.4**

Эксперименты: SQuAD

Model	Train FLOPs	Params	SQuAD 1.1 dev		SQuAD 2.0 dev		SQuAD 2.0 test	
			EM	F1	EM	F1	EM	F1
BERT-Base	6.4e19 (0.09x)	110M	80.8	88.5	–	–	–	–
BERT	1.9e20 (0.27x)	335M	84.1	90.9	79.0	81.8	80.0	83.0
SpanBERT	7.1e20 (1x)	335M	88.8	94.6	85.7	88.7	85.7	88.7
XLNet-Base	6.6e19 (0.09x)	117M	81.3	–	78.5	–	–	–
XLNet	3.9e21 (5.4x)	360M	**89.7**	**95.1**	87.9	**90.6**	87.9	90.7
RoBERTa-100K	6.4e20 (0.90x)	356M	–	94.0	–	87.7	–	–
RoBERTa-500K	3.2e21 (4.5x)	356M	88.9	94.6	86.5	89.4	86.8	89.8
ALBERT	3.1e22 (44x)	235M	89.3	94.8	87.4	90.2	88.1	90.9
BERT (ours)	7.1e20 (1x)	335M	88.0	93.7	84.7	87.5	–	–
ELECTRA-Base	6.4e19 (0.09x)	110M	84.5	90.8	80.5	83.3	–	–
ELECTRA-400K	7.1e20 (1x)	335M	88.7	94.2	86.9	89.6	–	–
ELECTRA-1.75M	3.1e21 (4.4x)	335M	**89.7**	94.9	**88.0**	**90.6**	**88.7**	**91.4**

Эксперименты



Заключение

Авторы придумали новую задачу в области понимания естественного языка. Ключевая идея - обучить кодировщик текста отличать настоящие токены от сгенерированных.

Данный подход показывает неплохие результаты в последующих задачах даже при использовании относительно небольших объемов вычислений.

Авторы надеются, что данный подход сделает разработку и применение предварительно обученных кодировщиков текста более доступными для исследователей и практиков с меньшим доступом к вычислительным ресурсам.

Вопросы

1. Опишите принцип работы метода Electra

2. Выпишите функцию потерь для генератора и дискриминатора

3. Какой оптимальный размер генератора по отношению к дискриминатору и почему плохо использовать одинаковые размеры для генератора и дискриминатора

Список литературы

- https://openreview.net/pdf?id=r1xMH1BtvB
- https://arxiv.org/pdf/1810.04805.pdf