Transfer Learning. ELMO, BERT и модификации

Хрыльченко Кирилл

Математические методы анализа текстов 2020

20 октября, 2020

Transfer Learning. Computer Vision

Умеем обучать большие нейросети (ResNet-152, 60 млн. параметров) на больших датасетах (ImageNet, 14 млн. изображений).

Хотим использовать большие нейросети на небольших датасетах.

Проблема: переобучение.

Решение:

- датасет среднего размера — инициализируем параметры сети значениями, полученными при обучении на ImageNet, затем обучаем сеть на новых данных
- маленький датасет — выкидываем голову, замораживаем оставшиеся слои, добавляем и обучаем новую голову

Multi-task Learning

Задача $d \in D$ — набор примеров $(x_i, y_i)_{i=1}^{N_d}$, где x_i — описание объекта, y_i — целевая переменная.

Пусть имеется набор задач $D = \{d_1, \ldots, d_n\}$. Хотим модель, решающую сразу все задачи[1]:

$$\mathrm{P}(\mathrm{output} \mid \mathrm{input}, \mathrm{task}, \theta, \theta_{\mathrm{task}}),$$

где θ — общая часть модели для всех задач, θ_{task} — часть модели под задачу task.

В NLP не нужны параметры под конкретную задачу. Одна модель: принимает текст — выдает текст[2]. Очень удобно!

[1] «One Model To Learn Them All», Kaiser et al, 2017.
[2] «Text-to-Text Transfer Transformer (T5)» by Raffel et al, 2020.

Transfer Learning

Transfer Learning[3] — нам важна только одна, **целевая** задача d из множества D, остальные вспомогательные.

Более популярная формулировка — обучение в два этапа:

- **Предобучение** (pretraining) — обучение модели на вспомогательных задачах
- **Дообучение** (finetuning) — инициализация части параметров модели весами, полученными при предобучении, затем дальнейшее[4] обучение на целевой задаче

Пример: предобучение векторных представлений слов с последующим использованием для решения NLP задач.

[3] перенос обучения
[4] возможно частичное

N-shot Learning

Для целевой задачи доступно:

- 0 примеров — **zero**-shot learning
- 1 – 10 примеров — **few**-shot learning

Человек может распознать объект, который он видел всего пару раз?

Человек может распознать объект, который он никогда не видел. Например, по текстовому описанию.

Человек может решать задачи, которые он никогда не решал, если ему объяснят суть задачи. Если же еще и пару примеров дадут, то вероятность решения еще больше.

Few-shot learning. Примеры

Вход модели:

«*Порфирий Петрович родился в Москве в 1938 году.*
Q: Где родился Порфирий Петрович?
A: В Москве.
Q: В каком году родился Порфирий Петрович?
A:»

Ожидаемый выход модели:

« В 1938 году.»

Zero-shot learning. Примеры

I'm not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I'm not a fool]**.

«I hate the word 'perfume'», Burr says. «It's somewhat better in French: **parfum**.»

Q: What is 65360 plus 16204?
A: **81564.**

Transfer Learning. Два подхода

1) **Feature-based** подход — предобучаем модель на вспомогательных задачах, используем её как один из «кирпичиков» в модели, решающей целевую задачу. Подбираем под каждую целевую задачу свою итоговую архитектуру.

Пример: предобучение векторных слов с последующим использованием для классификации текста.

2) **Finetuning** подход — предобучаем модель на вспомогательных задачах, затем дообучаем модель с небольшими изменениями на целевую задачу. Одна архитектура под все целевые задачи.

Contextualized embeddings

Полисемия — некоторые слова имеют несколько значений (смыслов). Контекст, в котором употребляется слово, влияет на конечный смысл.

Примеры:

- На этом фестивале мне удалось пострелять из **лука**.
- Из моих глаз полились слёзы. Кто-то рядом резал **лук**.
- Он спрятал **пистолет** за пиджаком.
- Давайте попробуем построить детерминированный конечный **автомат**, распознающий этот регулярный язык.

Построим векторное представление слова как функцию от всего предложения!

Языковое моделирование. Recap

Задача **языкового моделирования** — научиться оценивать вероятность последовательности слов $w_1, \ldots, w_n$: $\mathrm{P}(w_1, \ldots, w_n)$.

Авторегрессионная постановка задачи заключается в факторизации вероятности следующего вида:

$$\mathrm{P}(w_1, \ldots, w_n) = \mathrm{P}(w_1) \cdot \mathrm{P}(w_2 \mid w_1) \ldots \mathrm{P}(w_n \mid w_1, \ldots, w_n) = \\ = \prod_{i=1}^{n} \mathrm{P}(w_i \mid w_1, \ldots, w_{i-1}).$$

Left-to-right модель: $P(w \mid w_1, \ldots, w_{i-1}, \theta_{\rightarrow})$, где $\theta_{\rightarrow}$ — параметры модели. Аналогично, **right-to-left** модель — с параметрами $\theta_{\leftarrow}$.

ELMo

Будем также использовать общий слой θ_x, формирующие векторные представления слов; а также общую голову θ_s. Максимизируем правдоподобие документа $d = \{w_1, \ldots, w_n\}$:

$$\sum_{i=1}^{n} \log \mathrm{P}(w_i \mid w_1^{i-1}, \theta_x, \theta_s, \theta_{\rightarrow}) + \log \mathrm{P}(w_i \mid w_{i+1}^{n} \theta_x, \theta_s, \theta_{\leftarrow}) \rightarrow \max_{\Theta},$$

где $\Theta = \{\theta_x, \theta_s, \theta_{\rightarrow}, \theta_{\leftarrow}\}$.

Архитектура ELMo[5]:

- θ_x — char cnn + highway layer
- $\theta_{\rightarrow}$, $\theta_{\leftarrow}$ — трехслойные однонаправленные LSTM
- θ_s — линейный слой

[5] «Deep contextualized word representations», Peters et al, 2018.

ELMo

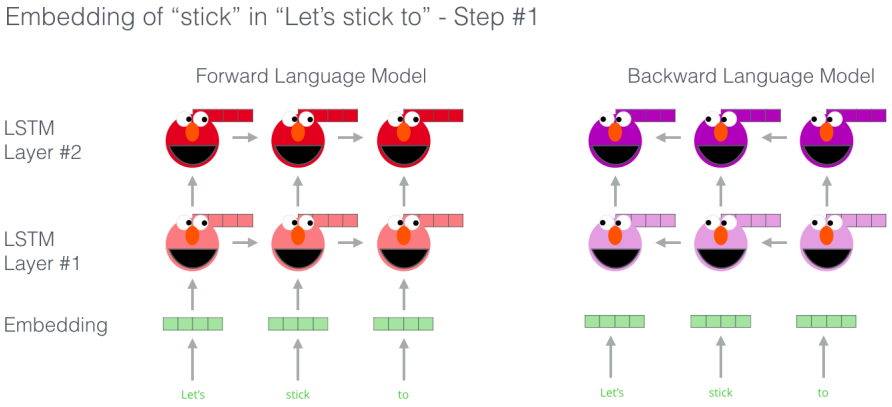
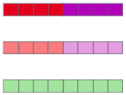


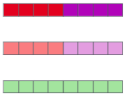
Figure: The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning), Jay Alammar

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

1- Concatenate hidden layers

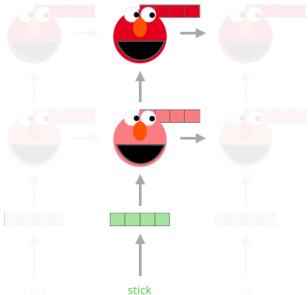


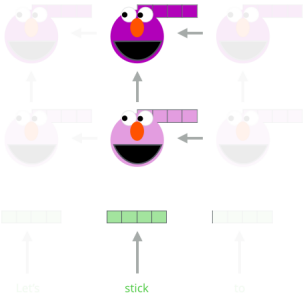
2- Multiply each vector by a weight based on the task



3- Sum the (now weighted) vectors

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





ELMo. Детали

- LSTM — трехслойная, для каждого слова имеем 4 векторных представления.
- Представление с более высокого слоя содержит более верхнеуровневые смысловые абстракции.
- Обучаются[6] софтмакс-нормализованные веса - важность каждого уровня:

$$\mathbf{ELMo}_k^{task} = \gamma^{task} \sum_{j=0}^{L} s_j^{task} h_{k,j}^{LM},$$

где s^{task} — софтмакс-нормализованные веса, $h_{k,j}^{LM}$ — векторное представление слова на k-й позиции, полученное конкатенацией предобученных векторных представлений с j-го уровня forward и backward языковых моделей.

- Конкатенация с векторами glove и использование в модели следующего уровня.

[6] с l_2 регуляризацией

GLUE

Dataset	Description	Data example	Metric
CoLA	Is the sentence grammatical or ungrammatical?	"This building is than that one." = **Ungrammatical**	Matthews
SST-2	Is the movie review positive, negative, or neutral?	"The movie is funny , smart , visually inventive , and most of all , alive ." = **.93056 (Very Positive)**	Accuracy
MRPC	Is the sentence B a paraphrase of sentence A?	A) "Yesterday , Taiwan reported 35 new infections , bringing the total number of cases to 418 ." B) "The island reported another 35 probable cases yesterday , taking its total to 418 ." = **A Paraphrase**	Accuracy / F1
STS-B	How similar are sentences A and B?	A) "Elephants are walking down a trail." B) "A herd of elephants are walking along a trail." = **4.6 (Very Similar)**	Pearson / Spearman
QQP	Are the two questions similar?	A) "How can I increase the speed of my internet connection while using a VPN?" B) "How can Internet speed be increased by hacking through DNS?" = **Not Similar**	Accuracy / F1
MNLI-mm	Does sentence A entail or contradict sentence B?	A) "Tourist Information offices can be very helpful." B) "Tourist Information offices are never of any help." = **Contradiction**	Accuracy
QNLI	Does sentence B contain the answer to the question in sentence A?	A) "What is essential for the mating of the elements that create radio waves?" B) "Antennas are required by any radio receiver or transmitter to couple its electrical connection to the electromagnetic field." = **Answerable**	Accuracy
RTE	Does sentence A entail sentence B?	A) "In 2003, Yunus brought the microcredit revolution to the streets of Bangladesh to support more than 50,000 beggars, whom the Grameen Bank respectfully calls Struggling Members." B) "Yunus supported more than 50,000 Struggling Members." = **Entailed**	Accuracy
WNLI	Sentence B replaces sentence A's ambiguous pronoun with one of the nouns - is this the correct noun?	A) "Lily spoke to Donna, breaking her concentration." B) "Lily spoke to Donna, breaking Lily's concentration." = **Incorrect Referent**	Accuracy

Figure: https://mccormickml.com/2019/11/05/GLUE/

ELMo. Результаты

Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

Table: Сравнение ELMo с предыдущими SOTA моделями на тестовых выборках популярных NLP задач

ELMo. Примеры

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular play on Alusik 's grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play .
	Olivia De Havilland signed to do a Broadway play for Garson {...}	{...} they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement .

Table: Сравнение ближайших соседей к слову «play» из GloVe и ближайших соседей при использовании ELMo

ULMFiT[7]. Три этапа обучения

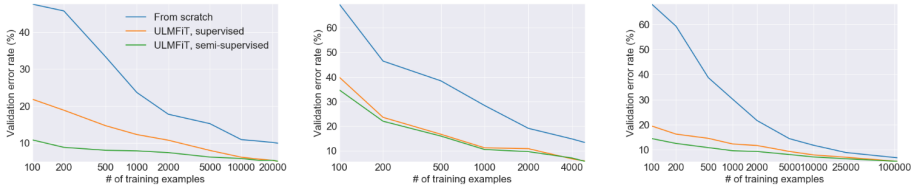


Figure: Ошибка на валидации для различных подходов обучения на датасетах IMDb, TREC-6, AG

1. Обучение языковой модели на сторонних выборках
2. Дообучение языковой модели на доменных данных
3. Обучение классификатора на целевую задачу

[7] Universal Language Model Fine-tuning for Text Classification, Jeremy Howard et al.

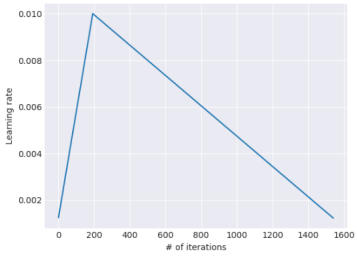


Figure: STLR — «треугольное» расписание для learning rate

- Gradual unfreezing — постепенно «размораживаем» сеть сверху вниз
- Discriminative fine-tuning — чем ниже слой, тем меньше learning rate
- Конкатенация $[h_T, maxpool(h), avgpool(h)]$ перед линейным слоем

Honorary mentions

- **Skip-Thought Vectors, Kiros et al**
  - encoder-decoder модель, восстанавливающая прошлое и следующее предложение
  - используют GRU
- **Semi-supervised Sequence Learning, Dai et al**
  - совместное предобучение LSTM как языковой модели и как автоэнкодера
  - *затухающий* пулинг: $h = \sum_{t=1}^{T} \gamma_t h_t$, где γ_t двигаются от 0 к 1.
- **Learned in Translation: Contextualized Word Vectors, McCann et al**
  - предобучение двухслойной LSTM в качестве энкодера в MT[8] задачах
  - на вход векторы GloVe, на выходе конкатенация с GloVe

[8] Machine Translation

BERT[9]

Проблемы ELMO:

- долгое обучение: $O(n^2 d)$, где n — длина последовательности, d — размерность скрытого состояния
- отсутствие двунаправленности языковой модели - «shallow» (пустая) конкатенация однонаправленных моделей, никак не связанных друг с другом

Решение:

- используем трансформероподобную архитектуру — $O(nd^2)$ по времени
- cloze task — обучение действительно двунаправленной языковой модели

[9] BERT: Pre-training of Deep Bidirectional Transformers forLanguage Understanding, Devlin et al

BERT. Masked Language Modeling

Процедура MLM: Исходное предложение: *my dog is hairy*. Случайным образом выбираем слова для маскирования: *my* и *hairy*.

- в 80 % случаев заменяем исходное слово токеном *[MASK]*
- в 10 % случаев заменяем слово другим случайным словом из словаря
- в 10 % случаев оставляем слово без изменений

Предсказываем исходные слова для трансформированных *my* и *hairy*.

Другими словами, BERT — **denoising autoencoder**; портим исходный текст и пытаемся восстановить оригинал.

BERT. MLM. Примеры

- **original**: добрый день! совершала перевод на сторонний банк, по реквизитам счета прошло уже 8 рабочих дней. сумма так и не поступила, как вернуть денежные средства?
- **masked**: добрый день! [MASK] перевод на сторонний банк, по реквизитам счета прошло уже 8 рабочих дней. сумма так и не поступила [MASK] как вернуть [MASK] [MASK]?
- **BERT**: добрый день! сделала перевод на сторонний банк, по реквизитам счета прошло уже 8 рабочих дней. сумма так и не поступила. как вернуть деньги обратно?

BERT. MLM. Примеры

- **original**: здравствуйте, недели две назад мне пришло смс с вашего банка **о** увеличении кредитного лимита до 600. 000 рублей, объясните пожалуйста по какой причине это не произошло
- **masked**: [MASK], недели две назад мне пришло смс с вашего [MASK] [MASK] увеличении кредитного лимита выслан 600. 000 рублей, объясните пожалуйста по какой причине [MASK] не произошло
- **BERT**: здравствуйте, недели две назад мне пришло смс с вашего банка **об** увеличении кредитного лимита в 600. 000 рублей, объясните пожалуйста по какой причине это не произошло

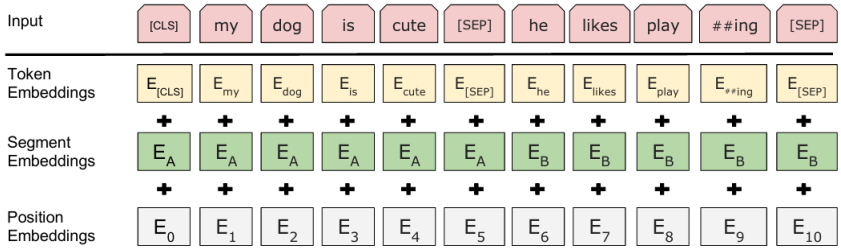
BERT. Next Sentence Prediction

Предсказываем связность предложений.

$$\text{Input} = \texttt{[CLS] the man went to [MASK] store [SEP]}$$
$$\texttt{he bought a gallon [MASK] milk [SEP]}$$
$$\text{Label} = \texttt{IsNext}$$

$$\text{Input} = \texttt{[CLS] the man [MASK] to the store [SEP]}$$
$$\texttt{penguin [MASK] are flight ##less birds [SEP]}$$
$$\text{Label} = \texttt{NotNext}$$

BERT. Входной слой



- обучаемые позиционные эмбеддинги
- сегментные эмбеддинги
- эмбеддинги последовательности подслов, на которую разбивается исходный текст токенизацией

BERT.Энкодер трансформера

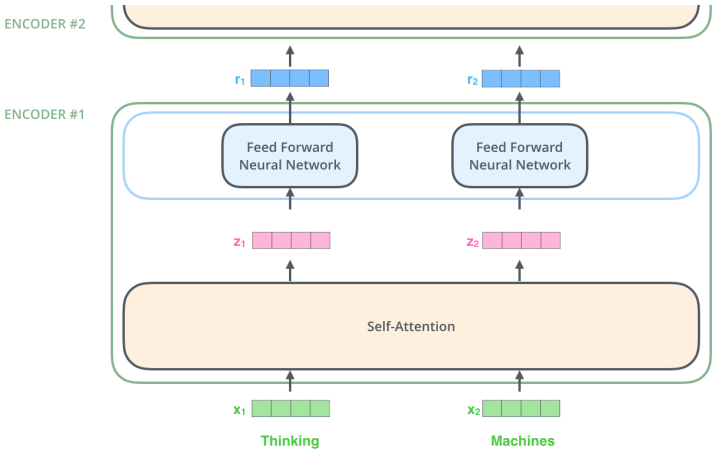


Figure: Основной блок модели BERT. The Illustrated Transformer by Jay Allamar

BERT. Векторные представления

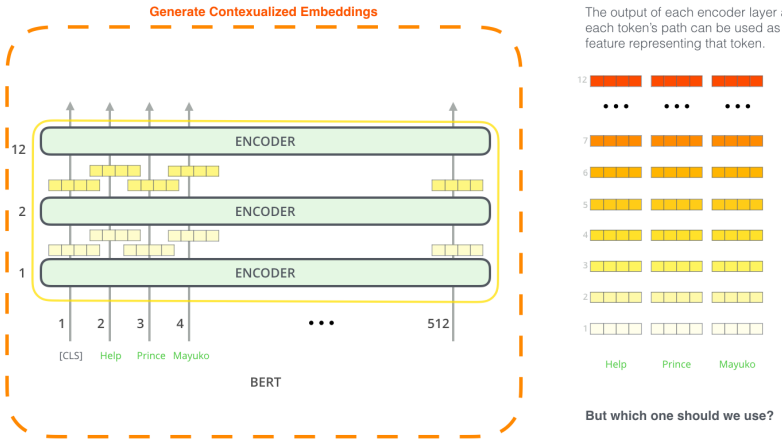


Figure: BERT как sequence to sequence модель. The Illustrated BERT, ELMo, and co.

BERT. Векторные представления

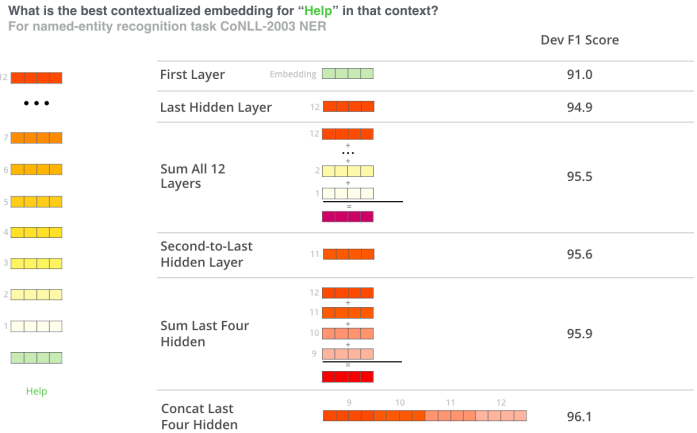


Figure: BERT как sequence to sequence модель. The Illustrated BERT, ELMo, and co.

BERT. Оптимизация

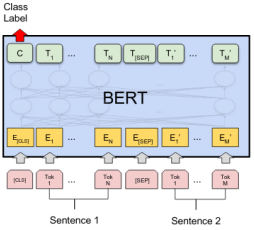
В качестве оптимизатора используется **Adam**[10] или **AdamW**[11]. В AdamW поправили l_2 регуляризацию — сделали ее «поверх» Адама.

Linear Scaling Rule — увеличивая размер батча в k раз, увеличивай в k раз learning rate. Ломается при слишком больших размерах, например, при 8126. **Linear warmup**: со старта линейно увеличиваем learning rate до некоторой пиковой величины.

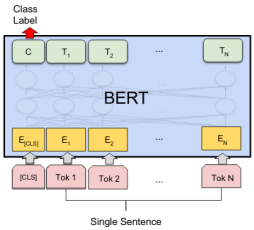
Адам хранит на каждый параметр значение производной функционала ошибки, а также ее квадрат — увеличиваем в три раза затраты по памяти при обучении. Критично для больших моделей. Работают над модификациями в **Adafactor**[12] и **LAMB**[13].

[10] Adam: A Method for Stochastic Optimization, Kingma et al
[11] Decoupled Weight Decay Regularization, Loshchilov et al
[12] Adafactor: Adaptive Learning Rates with Sublinear Memory Cost, Shazeer et al
[13] Large Batch Optimization for Deep Learning: Training BERT in 76 minutes

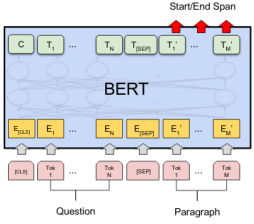
BERT. Finetuning



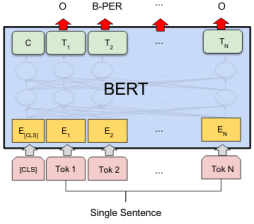
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA



(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

BERT. Гиперпараметры

Предобучение.

- размер батча 256
- пиковый learning rate 0.001
- linear warmup первые 10000 шагов
- равные веса у функций ошибки **MLM** и **NSP**
- для каждого предложения заранее генерируются 10 разных «масок»

Дообучение.

- размер батча в $\{32, 64\}$
- learning rate в $\{2e-5, \ldots, 5e-5\}$
- 2 – 4 эпохи
- клиппинг градиента, единичная максимальная норма

BERT. Результаты

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
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Figure: Результаты на GLUE[14] benchmark

- BERT_{base} — $L = 12, H = 768, A = 12$, 110 млн. параметров. Для честного сравнения с GPT
- BERT_{large} — $L = 24, H = 1024, A = 16$, 340 млн. параметров

[14] https://gluebenchmark.com/leaderboard

RoBERTa[15]

Что сделали:

- предобучали модель дольше и на большем количестве данных; увеличили длину входных последовательностей
- размер батча — 1024, 2048, 8192; при файнтюнинге linear warmup в течение 6% всех шагов оптимизации
- убрали NSP
- динамическое маскирование вместо статического — генерируют маски по мере обучения
- BPE вместо wordpiece токенизации
- тщательное ablation study всех гиперпараметров — даже эпсилон у Адама подбирают

Результаты — SOTA на 4 задачах из GLUE бенчмарка; а также на SQUAD.

[15] RoBERTa: A Robustly Optimized BERT Pretraining Approach, Liu et al

ALBERT[16]

Уменьшили количество параметров:

- Общие веса у слоев энкодера, а значит можно добавлять слои без добавления параметров
- Факторизация матрицы эмбеддингов. Большие эмбеддинги на входе не нужны, так как они context-free

Изменили предобучение:

- заменили NSP на **Sentence Order Prediction (SOP)** — предсказание порядка предложений
- маскируют подряд идущие слова (сэмплируют длину маски)
- используют LAMB; отключают dropout через миллион шагов

Результаты: при той же конфигурации в 18 раз меньше параметров, быстрее в 1.7 раз.

[16] ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, Lan et al

BERT. Дистилляция[18]

Возьмем большую модель, **учителя**, и «сожмём» ее знания без потери точности в легковесную модель — **студента**.

Общие моменты:

- Основная функция ошибки для обучения студента — кросс-энтропия со сглаженными[17] предсказаниями учителя в качестве истинных меток
- Пробуют придумывать свои дополнительные функции ошибки

Варианты:

- **PKD** — MSE между векторными представлениями
- **DistillBert** — на 60% быстрее, на 40% меньше, сохраняет 97% качества
- **TinyBERT**

[17] например, софтмакс с температурой

[18] дистилляцию также используют для few shot learning

ERNIE[19]

Модификация MLM:

- выделение из текста сущностей (инициалов, названий, просто фраз) с последующим маскированием

Новая задача — **Dialogue Language Modeling (DLM)**:

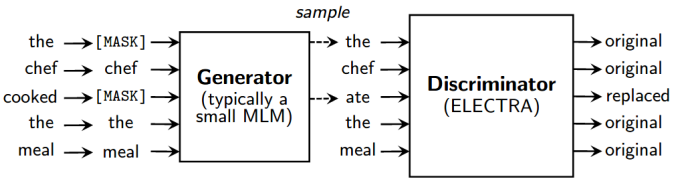
- есть запросы (queries) и ответы (responses). Как обучить BERT?
- сегментные эмбеддинги — query эмбеддинг и response эмбеддинг
- предсказываем «цельность» диалога — соответствие реплик запросам, заменяя реплики на случайные
- между репликами [SEP] токен

[19] ERNIE: Enhanced Representation through Knowledge Integration, Sun et al

ELECTRA[20]

Проблема: BERT при предобучении использует «пускает градиент» только через 15% процентов всех токенов, поэтому обучение медленное. Делаем больший процент — модель учитывает меньше информации, делаем меньший процент — упрощаем задачу и замедляем обучение еще больше.

Решение: GAN'ы! Генератор учится восстанавливать испорченные слова, дискриминатор пытается угадать, какие слова в предложении исправлял генератор, а какие — из оригинального текста.



[20] ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators, Clark et al

ELECTRA. Результаты

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**

Figure: Результаты на GLUE dev. выборке

XLM[21]

Данные для обучения:

- параллельные корпусы — сопоставленные друг другу предложения из разных языков
- monolingual корпусы — текст на одном языке. Таких гораздо больше

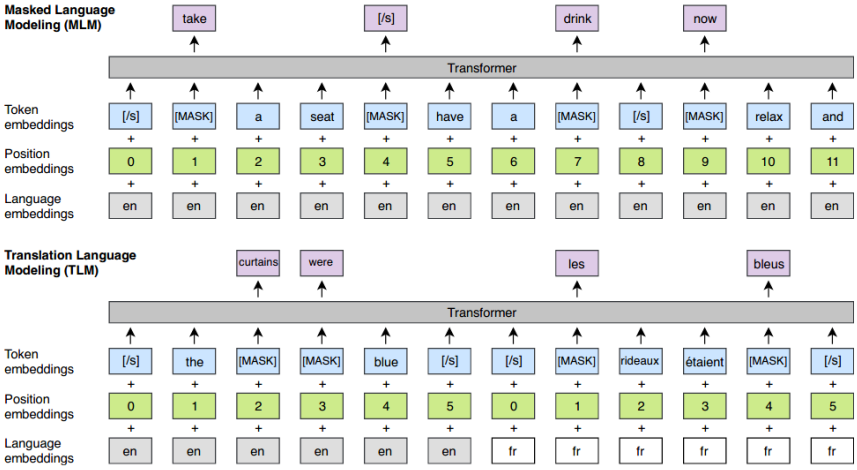
Задача: обучить модель, способную решать межязыковые задачи, не забывая про слабо представленные языки.

Предобучение:

- CLM – causal language modeling, обычная left-to-right языковая модель
- MLM — masked language modeling
- **TLM** — MLM поверх текстов вида "[CLS] source [SEP] target", где source и target — разные языки

[21] Cross-lingual Language Model Pretraining, Lample et al

XLM. Предобучение



Additional topics

Что еще есть:

- авторегрессивные модели — Transformer-XL, XLNet, GPT 1 – 3; борятся с длиной документов
  - относительные позиционные эмбеддинги
- полные трансформеры — T5, BART, MASS; text-to-text подход
- оптимизация трансформеров — longformer reformer, big bird
  - попытки снизить сложность по d — размерности эмбеддингов; например, заменить механизм внимания на что-то не квадратичное по d
  - попытки увеличить возможную длину входной последовательности
- multitask, одновременное решение всех задач — NLP Decathlon
- BERT не для NLP — BERT4Rec, vilBERT

Survey of efficient transformers

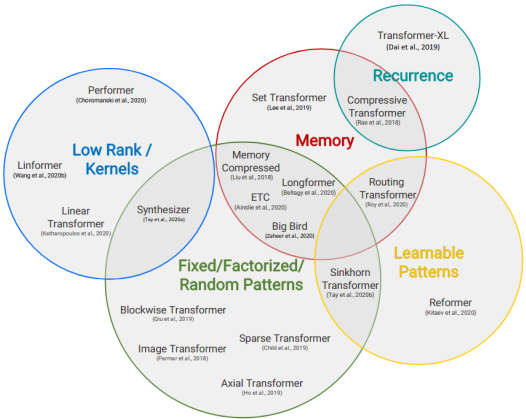


Figure: Efficient Transformers: A Survey, Tay et al

Третье практическое задание

- библиотека **transformers**[22] от «huggingface» — не надо писать реализацию BERT с нуля
- библиотека **tokenizers** от «huggingface» — не надо писать реализацию wordpiece или BPE с нуля
- импорт предобученного BERT:

```
from transformers import BertTokenizer, BertModel

tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
model = BertModel.from_pretrained("bert-base-uncased")
text = "Replace me by any text you'd like."
encoded_input = tokenizer(text, return_tensors='pt')
output = model(**encoded_input)
```

[22] https://github.com/huggingface