Transfer Learning. ELMO, BERT и модификации

Хрыльченко Кирилл

Математические методы анализа текстов 2021

12 октября, 2021

Transfer Learning. Computer Vision

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

Хотим использовать большие нейросети для задач с небольшим количеством данных для обучения. Больше model capacity — лучше качество.

Проблема: переобучение — не можем обучить большую нейросеть, не хватает данных.

Решение:

- предобучим нейросеть на вспомогательную задачу с большим количеством данных (e.g. ImageNet)
- если у нас датасет среднего размера — можем её дообучить на целевую задачу
- если датасет небольшой — фризим веса модели и обучаем только новую голову на целевую задачу

Multi-task Learning

Пусть задача $d \in D$ — это 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 одну модель,решающую сразу все задачи[1]:

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

где θ — общая часть модели для всех задач, θ_{task} — часть модели под задачу task, т.е. task-specific параметры.

Зачем это нужно:

- ограниченное количество вычислительных ресурсов
- мало данных для отдельных задач, хочется «перенести знание» от одних задач к другим

[1] «One Model To Learn Them All», Kaiser et al, 2017.

Transfer Learning

Transfer Learning[2] — 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 обучение на целевой задаче

Пример: предобучение векторных представлений слов с последующим использованием для решения разнообразных задач — word2vec, fasttext, etc.

[2] перенос обучения

N-shot Learning

Виды transfer learning в зависимости от размера датасетов для целевой задачи:

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

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

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

В NLP достаточно просто реализовать и multi-task learning[3], и n-shot learning.

[3] «Text-to-Text Transfer Transformer (T5)» by Raffel et al, 2020.

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, e.g. word2vec.

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.

Проблема: Word2vec каждому слову сопоставляет только один эмбеддинг, независящий от контекста.
Решение: построим векторное представление слова как функцию от всего предложения.

Языковое моделирование. Recap

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

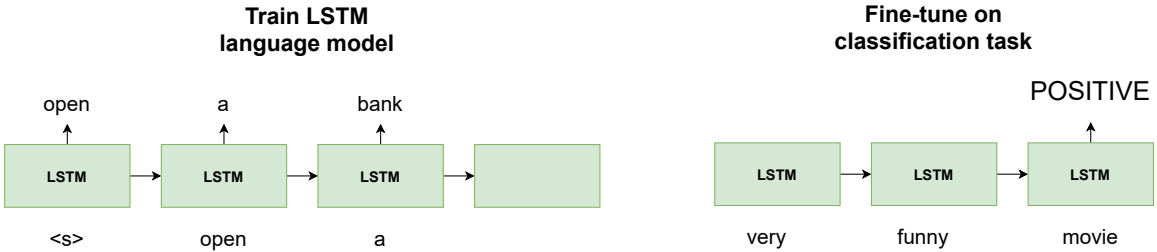
Авторегрессионная постановка задачи заключается в факторизации вероятности текста в произведение вероятностей слов:

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

Forward модель оценивает вероятность слова по левому контексту $P(w \mid w_1, \ldots, w_{i-1}, \theta_{\rightarrow})$, где $\theta_{\rightarrow}$ — параметры модели.

Аналогично, **backward** модель оценивает вероятность слова по правому контексту, параметры модели $\theta_{\leftarrow}$.

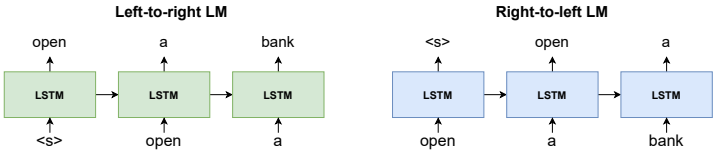
Semi-supervised Sequence Learning[4]



- Предобучили LSTM как языковую модель на имеющихся данных
- Дообучили на задачу классификации
- затухающий пулинг: $h = \sum_{t=1}^{T} \gamma_t h_t$, где γ_t двигаются от 0 к 1.

[4]Semi-supervised Sequence Learning, Dai et al (2015)

ELMo: Deep Contextual Word Embeddings[5]



- Предобучили две LSTM как языковые модели: прямую и обратную
- Формируют с помощью них векторные представления, подаваемые на вход модели

[5] https://arxiv.org/abs/1802.05365

ELMo. Обучение biLM

Как выглядит обучение forward LM:

1. Для каждого слова вычисляются независящие от контекста представления x_k как явные эмбеддинги слов или сверточные сети по символам
2. Эти эмбеддинги проходят через L слоёв LSTM
3. Для k-того слова в последовательности j-тый слой LSTM формирует контекстуальное представление слова $h^{\rightarrow}_{k,j}$, зависящее от всех предыдущих слов
4. Выход последнего слоя $h^{\rightarrow}_{k,L}$ используется для предсказания следующего слова t_{k+1} с помощью Softmax слоя

У forward и backward LM:

- общие представления слов (Θ_x)
- общий Softmax слой (Θ_s)
- раздельные параметры в LSTM слоях ($\Theta_{\rightarrow}$ и $\Theta_{\leftarrow}$)

Forward и backward LM обучаются совместно:

$$\sum_{k=1}^{N}(\log \mathrm{P}(t_k \mid t_1, \ldots, t_{k-1}; \Theta_x, \Theta_{\rightarrow}, \Theta_s) + \log \mathrm{P}(t_k \mid t-k+1, \ldots, t_N; \Theta_x, \Theta_{\leftarrow}, \Theta_s)) \rightarrow \max_{\Theta}.$$

ELMo. Применение

ELMo — это task-specific комбинация представлений слова из слоев biLM. Для каждого слова t_k biLM с L слоями формирует $2L + 1$ вектор:

$$R_k = \{x_k, h^{\rightarrow}_{k,j}, h^{\leftarrow}_{k,j} \mid j = \overline{0, L}\} = \{h_{k,j} \mid j = \overline{0, L}\}, \quad \text{где } h_{k,j} = [h^{\rightarrow}_{k,j}, h^{\leftarrow}_{k,j}] \ \forall\ k > 0,\ h_{0,j} = x_j.$$

Формируем итоговый вектор слова как комбинацию всех доступных векторов слова:

$$\text{ELMo}^{\text{task}}_k = \gamma^{task} \sum_{j=0}^{L} s^{\text{task}}_j h_{k,j},$$

где s^{task} — софтмакс-нормализуемые веса, а γ^{task} — скаляр, корректирующий ELMo вектор.

Вход для task-specific модели формируется как $[x_k, \text{ELMo}^{\text{task}}_k]$.

ELMo

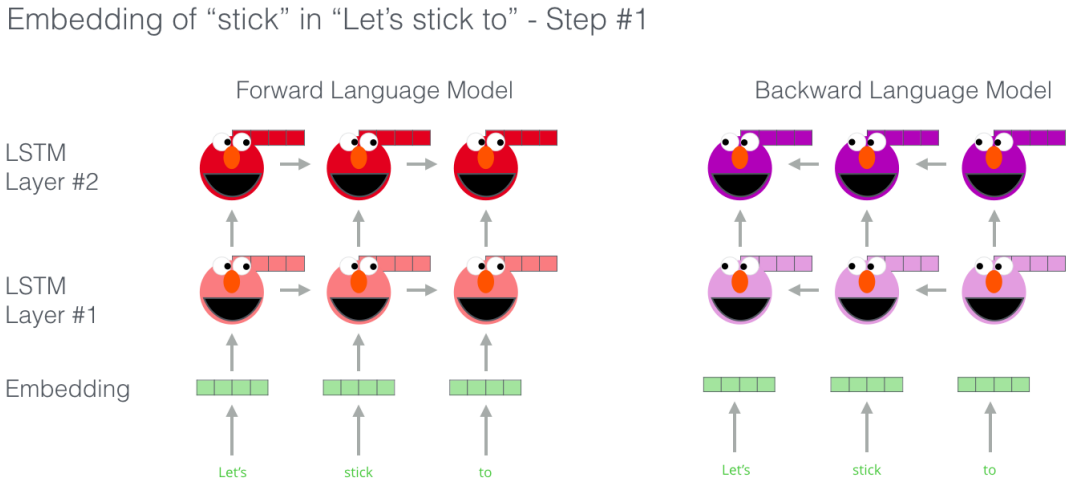
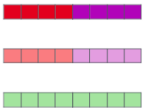


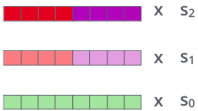
Figure: The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning), Jay Alammar.

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

1- Concatenate hidden layers

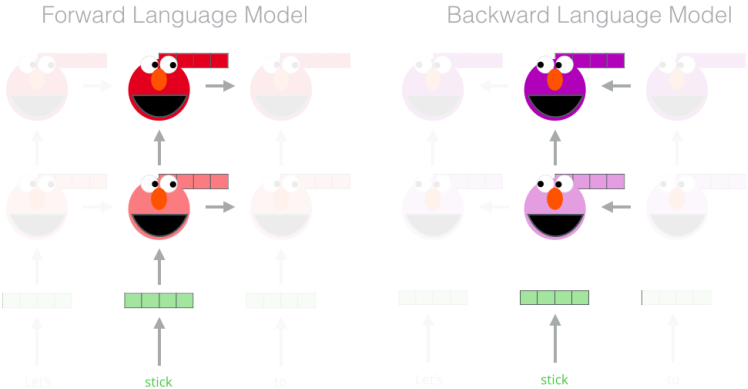


2- Multiply each vector by a weight based on the task



3- Sum the (now weighted) vectors

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



ELMo. 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.

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[6]. Три этапа обучения

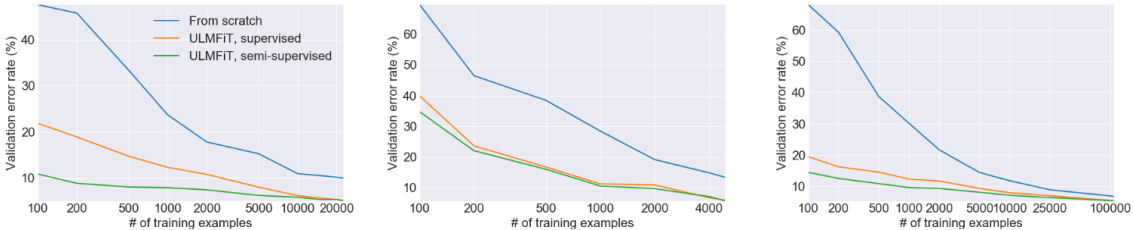


Figure: Ошибка на валидации для различных подходов обучения на датасетах IMDb, TREC-6, AG.

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

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

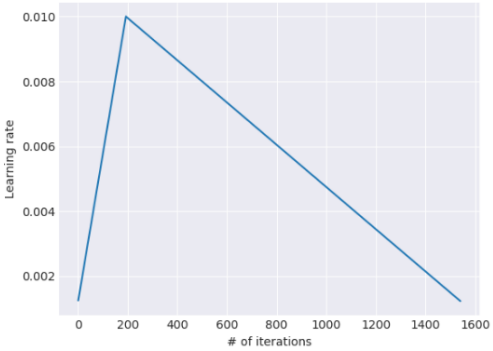


Figure: STLR — «треугольное» расписание для learning rate.

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

Honorary mentions

- **Skip-Thought Vectors, Kiros et al**
  - encoder-decoder модель, восстанавливающая прошлое и следующее предложение
  - используют GRU
- **Learned in Translation: Contextualized Word Vectors, McCann et al**
  - предобучение двухслойной LSTM в качестве энкодера в MT[7] задачах
  - на вход векторы GloVe, на выходе конкатенация с GloVe

Вопрос: нужны ли нам RNN-ки?

[7] Machine Translation

BERT[8]

Проблемы ELMo:

- последовательный характер вычислений препятствует параллелизации
- конкатенация однонаправленных моделей, нет реальной двунаправленности; в то время как понимание языка двунаправленно
- locality bias — близкий контекст важнее дальнего

Решение:

- используем трансформероподобную архитектуру
- cloze task — маскируем слова, учим модель их восстанавливать

Вопрос: какая вычислительная сложность у трансформера?

[8] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Devlin et al

Сложность самовнимания

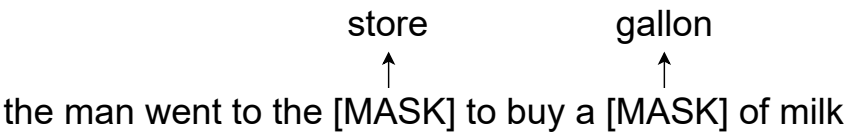
Пусть всего h голов внимания, у каждой размерность $\frac{d}{h}$. Входные эмбеддинги — $X \in \mathbb{R}^{n \times d}$. Тогда:

- Вычисление матриц $Q, K, V \in \mathbb{R}^{n \times d/h}$: $O(n \cdot d \cdot d/h)$.
- Вычисление скоров внимания $A = QK^T$: $O(n \cdot n \cdot d/h)$.
- Вычисление контекстных векторов $\text{softmax}(A)V$: $O(n \cdot d/h \cdot d/h)$.
- Конкатенация контекстных векторов всех h голов и линейный слой, приводящий конкатенацию к размерности вектора d: $O(n \cdot d \cdot d)$.

Итого: $O(nd^2 + n^2 d + \frac{d^2}{h} + nd^2) = O(nd^2 + n^2 d)$

BERT. Masked Language Modeling

MLM: Маскируем 15% слов во входном тексте, затем пытаемся восстанавить замаскированные слова.



- Мало маскирования: долго обучать
- Много маскирования: недостаточно контекста для предсказания

Masked Language Modeling

Проблема: токен [MASK] не используется в дообучении.

Решение: не всегда заменять токен маской. Вместо этого:

- 80% заменять на [MASK]: *went to the* ***store*** $\rightarrow$ *went to the* ***[MASK]***
- 10% заменять случайным словом: *went to the* ***store*** $\rightarrow$ *went to the* ***running***
- 10% оставлять в исходном виде: *went to the* ***store*** $\rightarrow$ *went to the* ***store***

Теперь модель вынуждена на верхнем уровне формировать адекватные векторные представления для всех слов; никогда не знает, для какого слова придется что-то предсказывать.

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

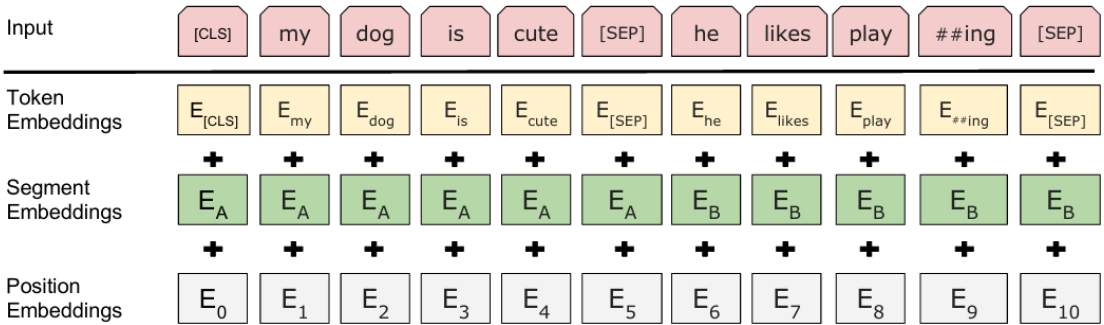
BERT. Next Sentence Prediction

Чтобы выучить взаимосвязь между предложениями, будем предсказывать является ли предложение B продолжением предложения A, или же каким-то другим случайным предложением:

Sentence A: The man went to the store.
Sentence B: He bought a gallon of milk.
Label: IsNextSentence

Sentence A: The man went to the store.
Sentence B: Penguins are flightless.
Label: NotNextSentence

BERT. Входной слой



- обучаемые позиционные эмбеддинги
- сегментные эмбеддинги
- словарь из 30000 эмбеддингов для WordPiece токенов

BERT.Энкодер трансформера

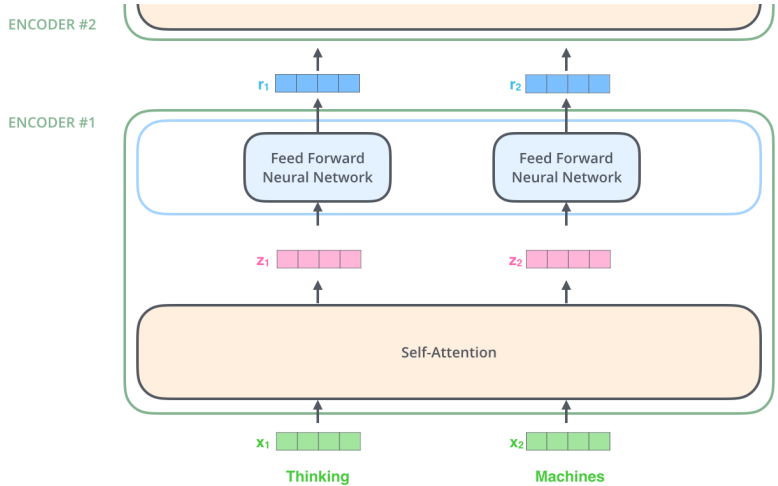
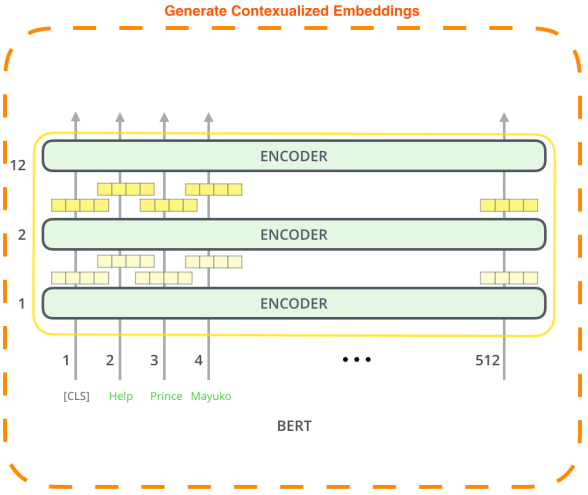


Figure: Основной блок модели BERT. The Illustrated Transformer by Jay Allamar

BERT. Векторные представления



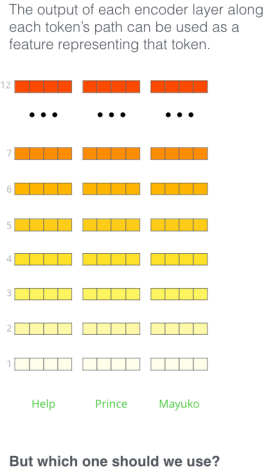
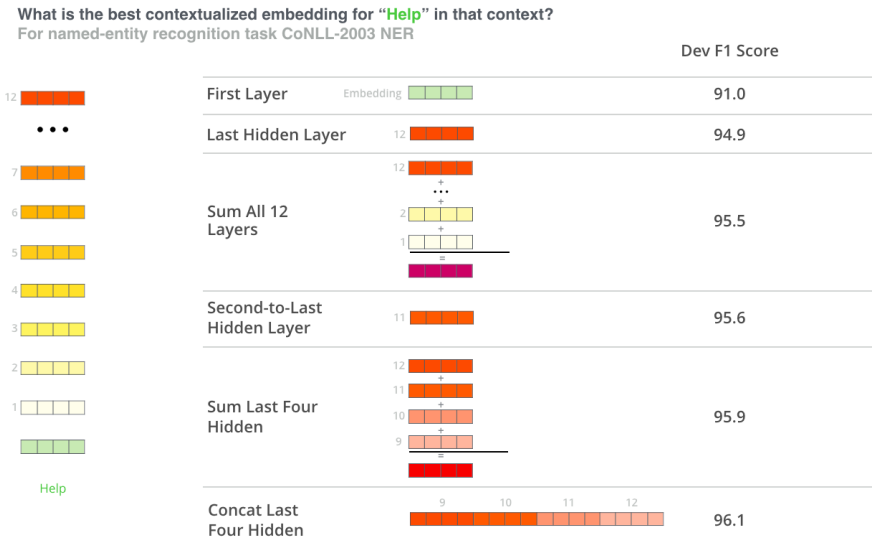
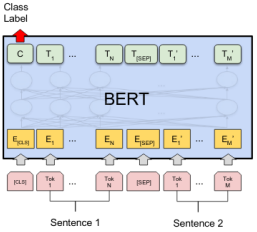


Figure: The Illustrated BERT, ELMo, and co.

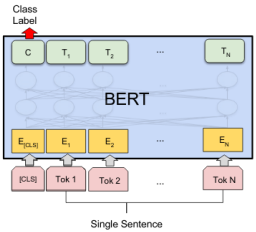
BERT. Векторные представления



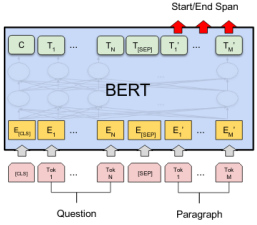
BERT. Finetuning



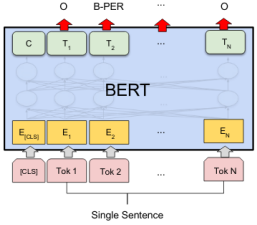
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



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



(c) Question Answering Tasks: SQuAD v1.1



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

BERT. Оптимизация

В качестве оптимизатора используется **Adam**[9] или **AdamW**[10]. 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**[11] и **LAMB**[12].

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

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

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

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[13] benchmark

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

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

RoBERTa[14]

Первое улучшение после BERT:

- 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.

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

XLNet.[15] Относительные позиционные эмбеддинги

Предложение: John ate a hot dog.

Абсолютное внимание: Насколько токен *dog* должен обратить внимание на *hot* (на любой позиции) и насколько *dog* на 4 позиции должен обратить внимание на токен на третьей позиции?

Относительное внимание: Насколько токен *dog* должен обратить внимание на *hot* на любой позиции и насколько *dog* должен обратить внимание на **прошлую позицию**?

[15] Generalized Autoregressive Pretraining for Language Understanding, Yang et al, 2019

XLNet. Пермутации

В GPT мы предсказываем сначала 1-е, затем 2-е, затем 3-е слово и т.д.

XLNet: перемешаем слова и будем предсказывать их в произвольном порядке: сначала 5-е слово, затем 2-е, затем 3-е и т.д.

Реализуется в виде специальной маски внимания.

Результат: более эффективно используем сэмплы, предсказываем что-то для каждого слова.

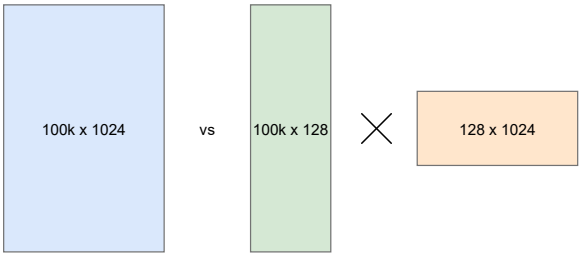
Model	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B	WNLI
Single-task single models on dev									
BERT [2]	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-
RoBERTa [21]	90.2/90.2	94.7	92.2	**86.6**	96.4	**90.9**	68.0	92.4	-
XLNet	**90.8/90.8**	**94.9**	**92.3**	85.9	**97.0**	90.8	**69.0**	**92.5**	-

Figure: Результаты на GLUE для XLNet и RoBERTa.

ALBERT[16]

Уменьшили количество параметров:

- Общие веса у слоев энкодера — больше слоев, столько же параметров (cross-layer parameter sharing)
- Факторизация матрицы эмбеддингов. Большие эмбеддинги на входе не нужны, так как они context-free



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

ALBERT

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
- отключают dropout через миллион шагов

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

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg	Speedup
BERT	base	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3	4.7x
	large	334M	92.2/85.5	85.0/82.2	86.6	93.0	73.9	85.2	1.0
ALBERT	base	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1	5.6x
	large	18M	90.6/83.9	82.3/79.4	83.5	91.7	68.5	82.4	1.7x
	xlarge	60M	92.5/86.1	86.1/83.1	86.4	92.4	74.8	85.5	0.6x
	xxlarge	235M	**94.1/88.3**	**88.1/85.1**	**88.0**	**95.2**	**82.3**	**88.7**	0.3x

Models	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS
Single-task single models on dev								
BERT-large	86.6	92.3	91.3	70.4	93.2	88.0	60.6	90.0
XLNet-large	89.8	93.9	91.8	83.8	95.6	89.2	63.6	91.8
RoBERTa-large	90.2	94.7	**92.2**	86.6	96.4	**90.9**	68.0	92.4
ALBERT (1M)	90.4	95.2	92.0	88.1	96.8	90.2	68.7	92.7
ALBERT (1.5M)	**90.8**	**95.3**	**92.2**	**89.2**	**96.9**	**90.9**	**71.4**	**93.0**

Figure: BERT, XLNet, RoBERTa, ALBERT.

ERNIE[17]

Модификация MLM:

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

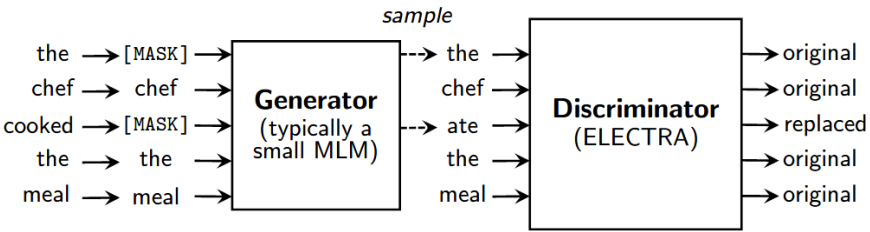
Новая задача — **Dialogue Language Modeling (DLM)**:

- 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

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

ELECTRA[18]

Генератор учим также как BERT, дискриминатор учится отличать исходные слова от восстановленных генератором.



[18]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[20]

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:

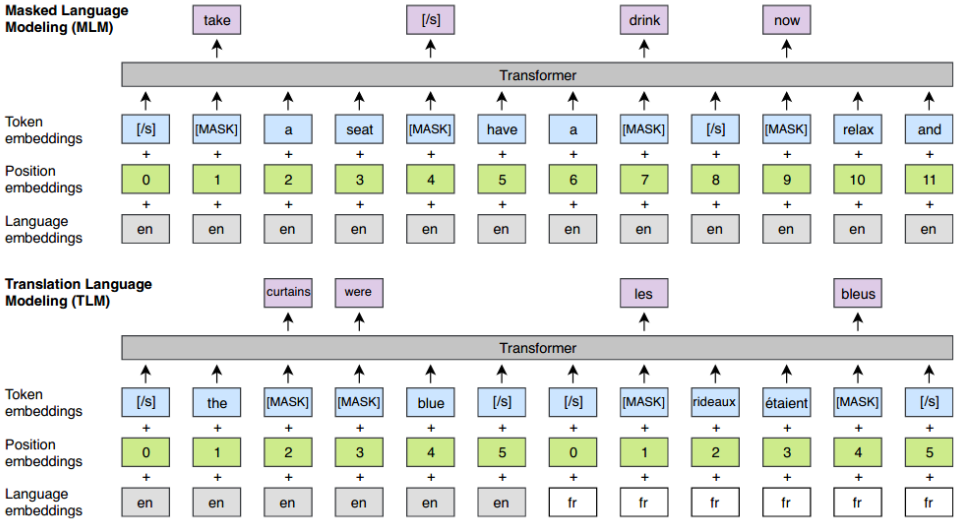
- CLM – causal language modeling, обычная left-to-right языковая модель
- MLM — masked language modeling
- **TLM** — MLM поверх текстов вида "[CLS] source [SEP] target", где source и target — разные языки

Уже есть XLM-E[19] = XLM + ELECTRA

[19] XLM-E: Cross-lingual Language Model Pre-training via ELECTRA

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

XLM. Предобучение



BERT. Дистилляция[21]

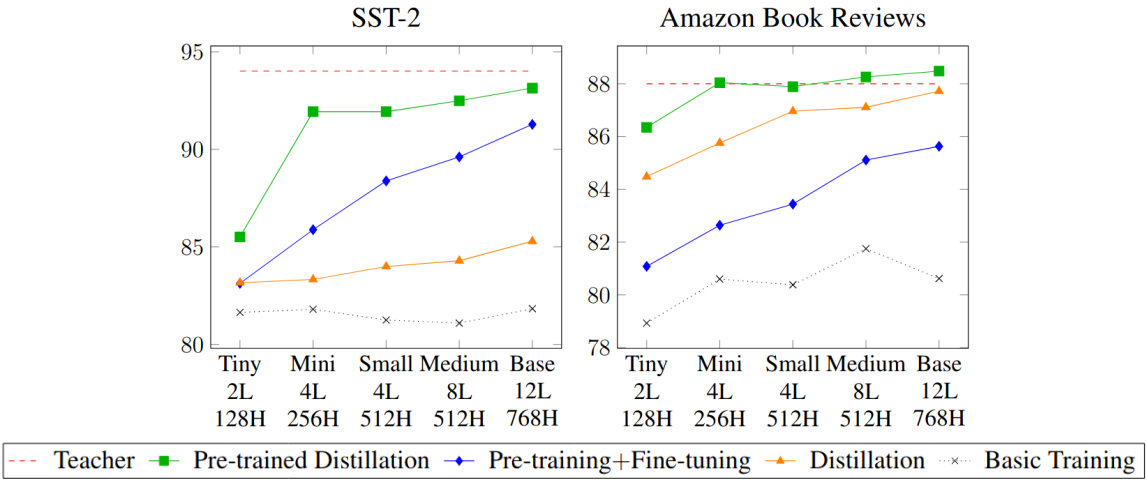
Проблема: хотим трансформеры в продакшне.

- Обучаем «учителя» — большую, тяжелую SOTA модель
- Размечаем большое количество неразмеченных данных
- Обучаем маленькую модель, «ученика», повторять за «учителем»
- функция ошибки — кросс-энтропия или MSE со сглаженными предсказаниями учителя в качестве истинных меток

Amazon Book Reviews — 50k размеченных примеров, 8 млн. неразмеченных.

[21] дистилляцию также используют для few shot learning

Well-Read Students Learn Better, Turc et al[22]



[22] https://arxiv.org/pdf/1908.08962.pdf

Дистилляция

Другие варианты:

- **PKD**[23] — MSE между векторными представлениями
- **DistillBert** — на 60% быстрее, на 40% меньше, сохраняет 97% качества

Model	SST-2 (67k)	MRPC (3.7k)	QQP (364k)	MNLI-m (393k)	MNLI-mm (393k)	QNLI (105k)	RTE (2.5k)
$BERT_{12}$ (Google)	93.5	88.9/84.8	71.2/89.2	84.6	83.4	90.5	66.4
$BERT_{12}$ (Teacher)	94.3	89.2/85.2	70.9/89.0	83.7	82.8	90.4	69.1
$BERT_6$-FT	90.7	85.9/80.2	69.2/88.2	80.4	79.7	86.7	63.6
$BERT_6$-KD	91.5	**86.2/80.6**	70.1/88.8	80.2	79.8	88.3	64.7
$BERT_6$-PKD	**92.0**	85.0/79.9	**70.7/88.9**	**81.5**	**81.0**	**89.0**	**65.5**
$BERT_3$-FT	86.4	80.5/**72.6**	65.8/86.9	74.8	74.3	84.3	55.2
$BERT_3$-KD	86.9	79.5/71.1	67.3/87.6	75.4	74.8	84.0	56.2
$BERT_3$-PKD	**87.5**	**80.7**/72.5	**68.1/87.8**	**76.7**	**76.3**	**84.7**	**58.2**

Figure: Результаты на GLUE. FT — дообучение без дистилляции, KD — обычная дистилляция, PKD — дистилляция промежуточных выходов трансформера.

[23] Patient Knowledge Distillation for BERT Model Compression, Sun et. al

Additional topics

Что еще есть:

- авторегрессивные модели — Transformer-XL, GPT 1 – 3; борятся с длиной документов
- полные трансформеры — T5, BART, MASS; text-to-text подход
- оптимизация трансформеров — longformer, reformer, performer, big bird
  - попытки снизить сложность по d или n
  - попытки увеличить возможную длину входной последовательности
- multitask, одновременное решение всех задач — NLP Decathlon
- BERT не для NLP — BERT4Rec, vilBERT, PRM

Лекция от Jacob Devlin — `https://www.youtube.com/watch?v=knTc-NQSjKA`

Survey of efficient transformers

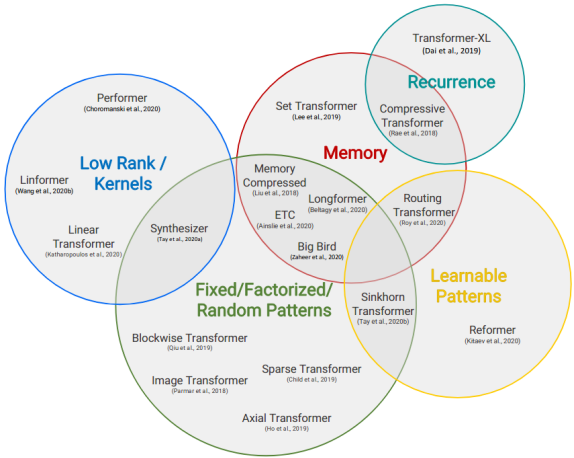


Figure: Efficient Transformers: A Survey, Tay et al