

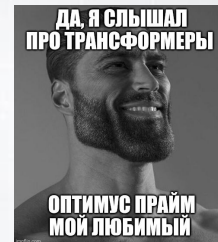
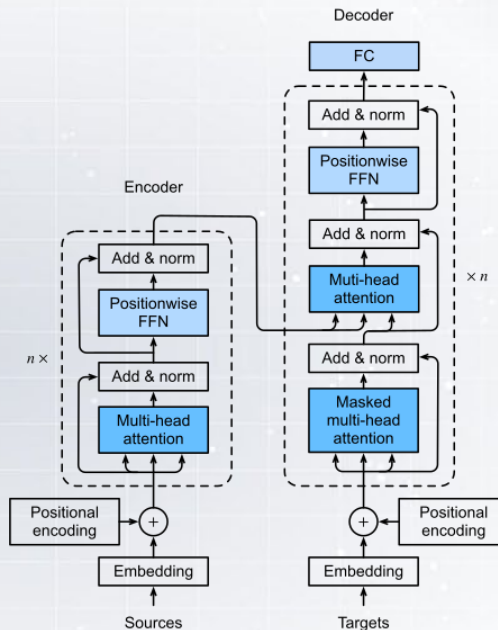
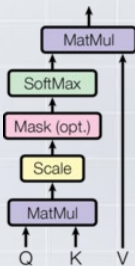


BERTology

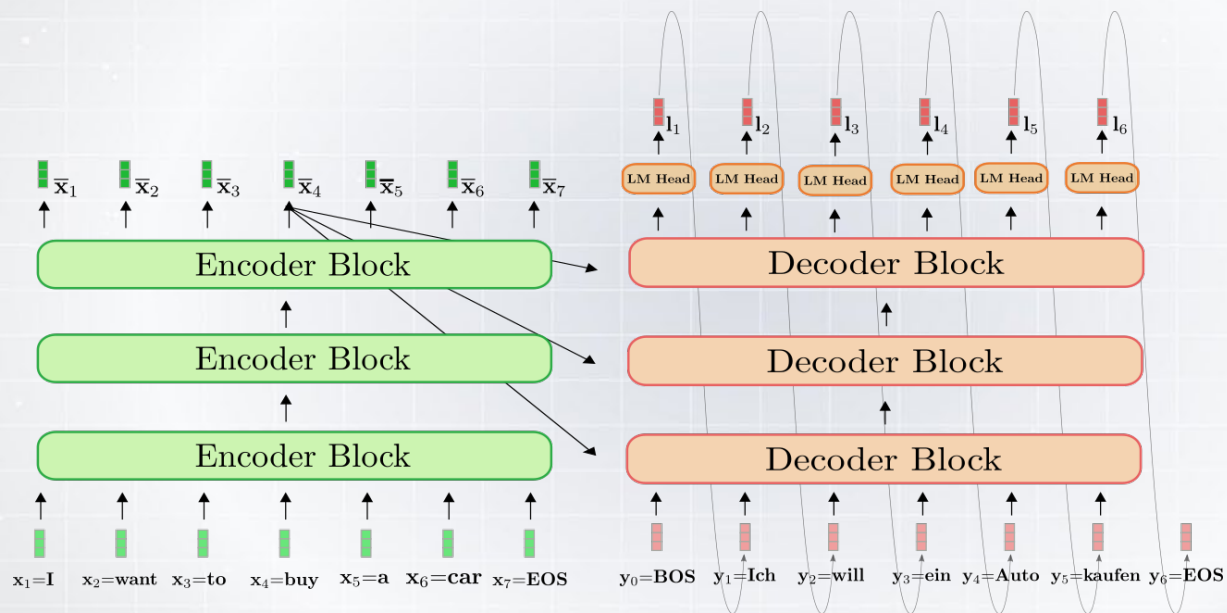
Encoder-Decoder Architecture

Вспомним: что здесь что и для чего нужно?

- Input embeddings
- Positional encoding
- Multi-head attention (attention?)
- Residual & LayerNorm
- Positionwise Feed Forward
- Key, Query, Value
- Формула Attention(Q , K , V)



Autoregression



LMs

BERT, GPT, T5, RoBERTa...

Эволюция языковых моделей

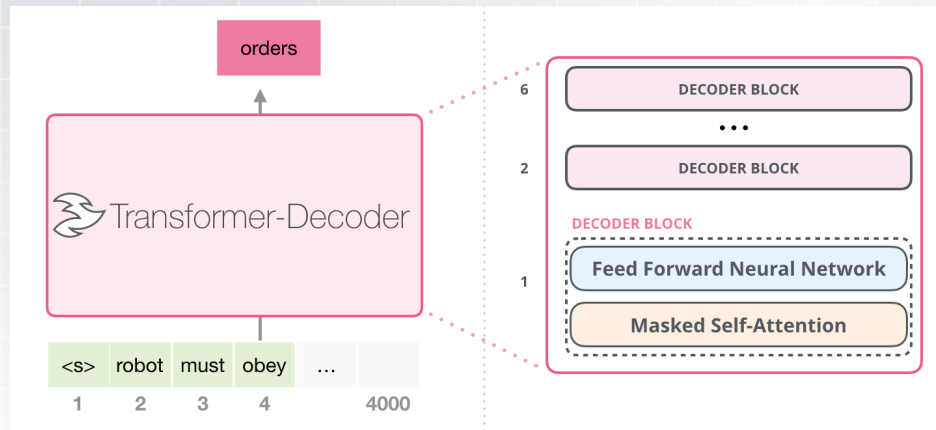


GPT



- Авторегрессивная
- Построена с использованием только блоков decoder
- Потеряла возможность видеть всю последовательность
- Зато отлично генерит
- Имеет несколько вариантов размеров

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}



GPT



Применения decoder-only transformer:

- Генерация текста
- Чат-боты
- Машинный перевод
- Саммаризация
- Генерация музыки (што?) (ga!)

BERT

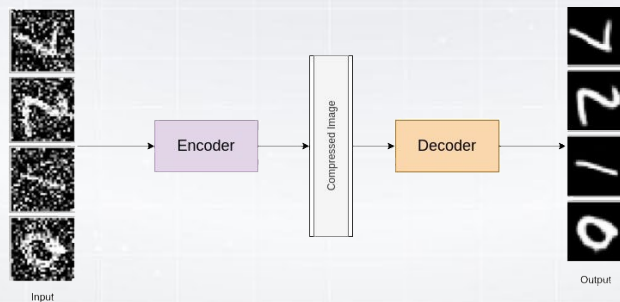
- Обучен задаче: masked language modelling
- Bidirectional:
 - Не авторегрессивный
 - Построен с использованием только блоков encoder
 - Видит контексты слева и справа
- Auxiliary task: next sentence prediction
- Имеет несколько вариантов размеров

Version	Hidden units	#layers	#parameters
BERT-large	1024	24-layer	340 million
BERT-base	768	12-layer	110 million



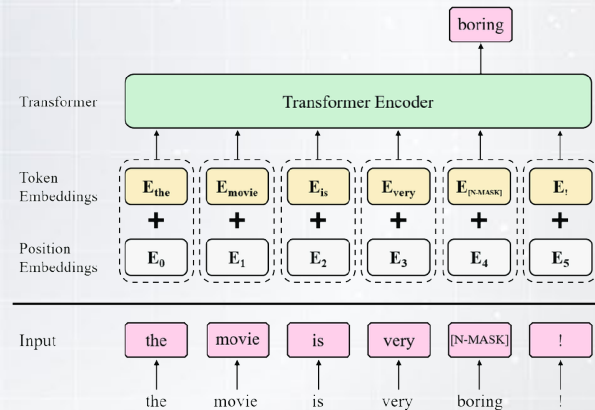
Задача masked language modelling

- Autoencoder: основная идея
- Будем пытаться восстановить исходный объект по собранным с него фичам
- Но этого недостаточно: исходный объект еще попортим
- То же можно сделать с языковыми данными: заменить часть токенов на маски



Задача masked language modelling

- Замаскируем 15% токенов:
 - 80% из этих 15 заменим на токен [MASK]
 - 10% заменим на случайный другой токен
 - 10% оставим неизменными
- Если больше маскировать, то слишком сильно испортим
- Из-за этого медленно учится (в отличие от GPT)



Задача next sentence prediction

- Подаем сразу несколько предложений с разделителем
- 50% в обучающей выборке действительно следующее предложение
- 50% - случайные сэмплы из корпуса
- NSP токен – [CLS]

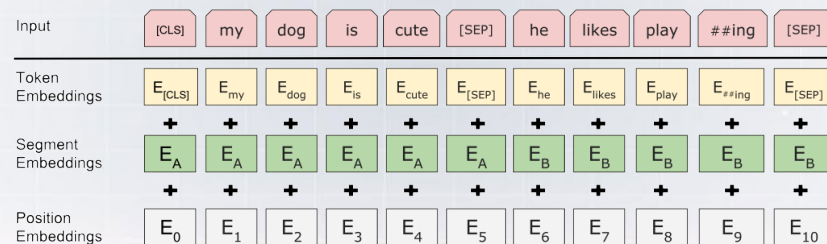
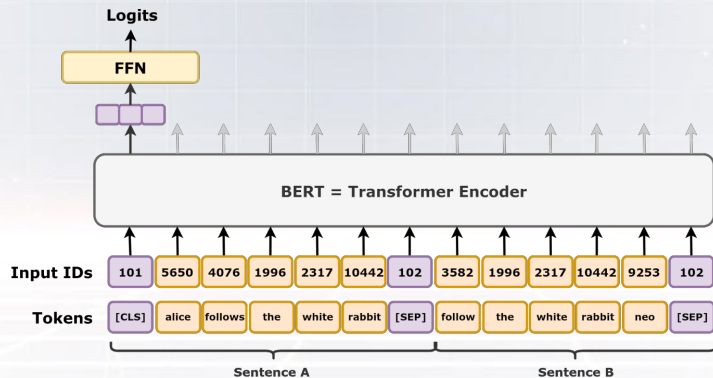


Figure 2: BERT input representation. The input embeddings is the sum of the token embeddings, the segmentation embeddings and the position embeddings.

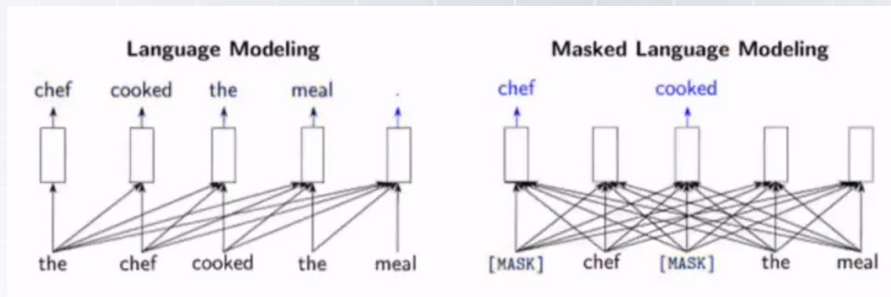
Задача next sentence prediction

- Подаем сразу несколько предложений с разделителем
- 50% в обучающей выборке действительно следующее предложение
- 50% - случайные сэмплы из корпуса
- NSP токен – [CLS]
- По этому токenu предсказываем 0 или 1



GPT vs BERT

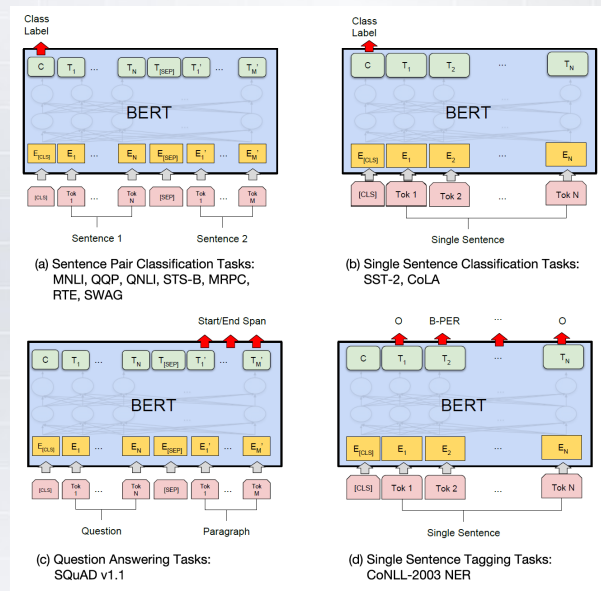
- GPT: language modelling (по слову за раз)
- BERT: masked language modelling (все слова сразу)



BERT: задачи

- Классификация по парам предложений
- Классификация по одному предложению
- Question Answering
- Классификация по токенам

- MNLI – textual entailment
- QQP – Quora Question Pairs dataset
- QNLI – question-paragraph pairs
- STS-B – Semantic Textual Similarity Benchmark
- MRPC – Microsoft Research Paraphrase Corpus
- RTE – Recognizing Textual Entailment
- SWAG – 113k multiple choice questions about grounded situations
- SST-2 – Stanford Sentiment Treebank
- CoLA – The Corpus of Linguistic Acceptability
- SQuAD – Stanford Question Answering Dataset



BERT Layers

What is the best contextualized embedding for “Help” in that context?

For named-entity recognition task CoNLL-2003 NER

12



...

7



6



5



4



3



2



1



Help

First Layer

Embedding



91.0

Last Hidden Layer

12



94.9

Sum All 12
Layers

12



+

...

+

2



+

1



=



95.5

Second-to-Last
Hidden Layer

11



95.6

Sum Last Four
Hidden

12



+

11



+

10



+

9

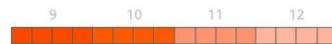


=



95.9

Concat Last
Four Hidden

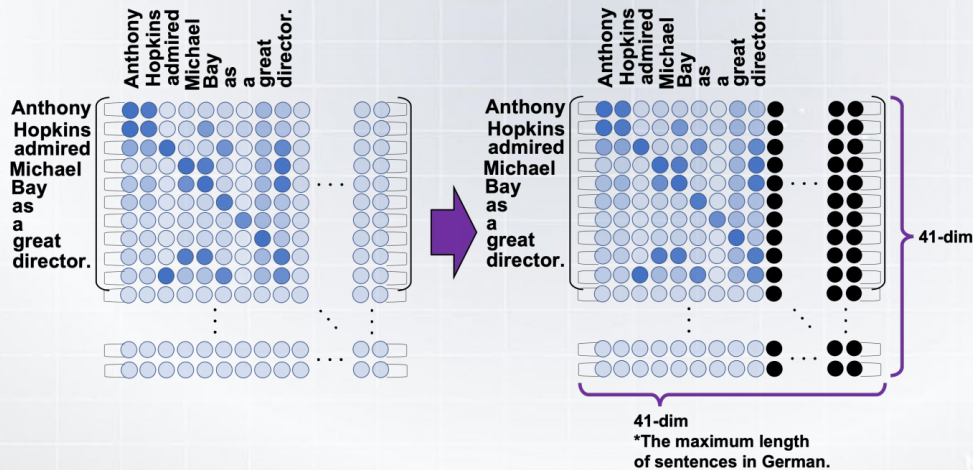


96.1

Pad masking

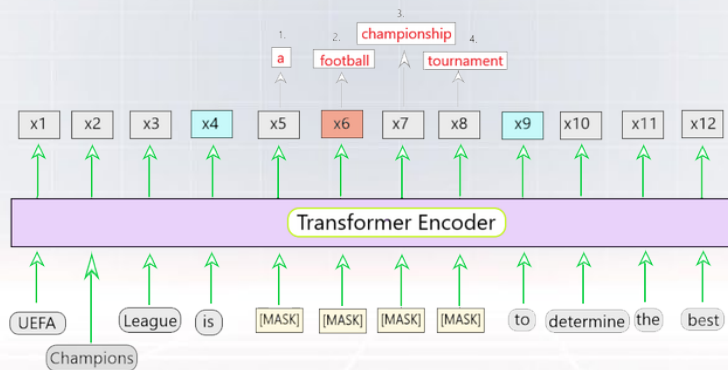
- Последовательности в батче могут быть разной длины – добиваем их $-\infty$
- Стоит удостовериться, что модель выдает одинаковые ответы вне зависимости от падов: $x.\text{mean}(\text{dim}=1)$ не должно меняться в зависимости от их количества

Encoder padding mask in practice



Span prediction

- Маскируем сразу по несколько токенов
- Учитывая, что у нас BPE-токенизация, это полезно



RoBERTa

Тот же BERT, только в профиль:

- Больше тренировочных данных, батчи большего размера, дольше учили
- Убрали задачу NSP – решили, что она бесполезная
- SeqLen длиннее
- Токены маскируем динамически (в стандартном Бертe один раз замаскировали и учили)
- [Статья](#)

	BERT	RoBERTa	DistilBERT	XLNet
Size (millions)	Base: 110 Large: 340	Base: 110 Large: 340	Base: 66	Base: ~110 Large: ~340
Training Time	Base: 8 x V100 x 12 days* Large: 64 TPU Chips x 4 days (or 280 x V100 x 1 days*)	Large: 1024 x V100 x 1 day; 4-5 times more than BERT.	Base: 8 x V100 x 3.5 days; 4 times less than BERT.	Large: 512 TPU Chips x 2.5 days; 5 times more than BERT.
Performance	Outperforms state-of-the-art in Oct 2018	2-20% improvement over BERT	3% degradation from BERT	2-15% improvement over BERT
Data	16 GB BERT data (Books Corpus + Wikipedia). 3.3 Billion words.	160 GB (16 GB BERT data + 144 GB additional)	16 GB BERT data. 3.3 Billion words.	Base: 16 GB BERT data Large: 113 GB (16 GB BERT data + 97 GB additional). 33 Billion words.
Method	BERT (Bidirectional Transformer with MLM and NSP)	BERT without NSP**	BERT Distillation	Bidirectional Transformer with Permutation based modeling

ALBERT

- Вместо 12 слоев стандартного Берта – один слой, но применяется 12 раз
- То есть, у наших слоев общие параметры
- Следовательно, параметров меньше в разы
- По качеству чуть лучше Роберты
- Добавили задачу Sentence Order Prediction
- Статья



DistilBERT

- Учимся предсказывать не на сырых текстах, а на предсказаниях исходного большого Берта
- Качество сопоставимое: скорость гораздо быстрее
- Выучивать предсказания проще, потому что большой Берт за нас уже сырые данные обработал
- Статья

Table 1: DistilBERT retains 97% of BERT performance. Comparison on the dev sets of the GLUE benchmark. ELMo results as reported by the authors. BERT and DistilBERT results are the medians of 5 runs with different seeds.

Model	Score	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3

T5 (Text-to-Text Transfer Transformer)





















- Основная задача – Text to Text (на входе получаем текст – генерируем другой текст)
- Внутри – обычный трансформер
- Использует relative positional encodings (позиция токена вычисляется не абсолютно, а при помощи key-query значений)
- Обучались на Colossal Clean Common Crawl – не просто взяли сырые тексты, а немного их почистили
- (огромная) [статья](#)

Language Transfer

- BERT и RoBERTa имеют мультязычные версии: mBERT и XLM-R
- Обычно учим downstream task на размеченных данных конкретного языка
- Но модель знает все свои языки одновременно
- язык-донор – язык, на котором мы учили downstream task
- язык-реципиент – язык, на котором делаем инференс
- zero-shot task – не показывали модели размеченных данных на языке-реципиенте
- few-shot task – немножечко данных все-таки показали
- Доказано, что качество при переносе с англ падает не больше, чем на 25%

Оценка языковых моделей

- **Бенчмарк** – набор задач и данных для них, по результатам решения которых можно оценить качество модели
- Для английского языка: SuperGLUE

Name	Identifier	Download	More Info	Metric
Broadcoverage Diagnostics	AX-b			Matthew's Corr
CommitmentBank	CB			Avg. F1 / Accuracy
Choice of Plausible Alternatives	COPA			Accuracy
Multi-Sentence Reading Comprehension	MultiRC			F1a / EM
Recognizing Textual Entailment	RTE			Accuracy
Words in Context	WIC			Accuracy
The Winograd Schema Challenge	WSC			Accuracy
BoolQ	BoolQ			Accuracy
Reading Comprehension with Commonsense Reasoning	ReCoRD			F1 / Accuracy
Winogender Schema Diagnostics	AX-g			Gender Parity / Accuracy

Оценка языковых моделей

- Для русского языка: [Russian SuperGLUE](#)

Name	Task type	Identifier	Download	Info	Metrics	Train/Val/Test size
Linguistic Diagnostic for Russian	NLI & diagnostics	LiDiRus	↓	More	Matthews Corr	0/0/1104
Russian Commitment Bank	NLI	RCB	↓	More	Avg. F1 / Accuracy	438/220/438
Choice of Plausible Alternatives for Russian language	Common Sense	PARus	↓	More	Accuracy	400/100/500
Russian Multi-Sentence Reading Comprehension	Machine Reading	MuSeRC	↓	More	F1a / EM	500/100/322
Textual Entailment Recognition for Russian	NLI	TERRa	↓	More	Accuracy	2616/307/3198
Russian Words in Context (based on RUSSE)	Common Sense	RUSSE	↓	More	Accuracy	19845/8508/18892
The Winograd Schema Challenge (Russian)	Reasoning	RWSD	↓	More	Accuracy	606/204/154
Yes/no Question Answering Dataset for the Russian	World Knowledge	DaNetQA	↓	More	Accuracy	1749/821/805
Russian Reading Comprehension with Commonsense Reasoning	Machine Reading	RuCoS	↓	More	F1 / EM	72193/7577/7257