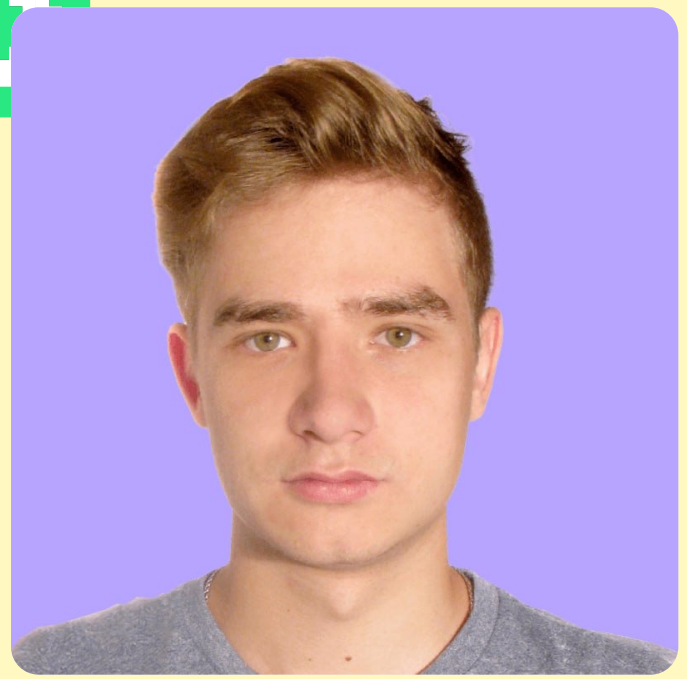
Семинар 2

YOUNG &78 YANDEX



Лысяков Аркадий
Выпускник и преподаватель МФТИ,
ex-руководитель команды продуктовой аналитики Яндекс Картинок

Содержание

01 Рекап теории: машинный перевод, Transformer, BERT, GPT, T5

02 Практика: работа с BERT, NMT

Задача перевода

- Получить функцию преобразования текстов на исходном языке в эквивалентные по смыслу тексты на целевом языке
- “Текстами” считаем любую последовательность символов из алфавита

$$f : X \to Y$$

или

$$f : x^{\|\infty\|} \to y^{\|\infty\|}$$

Word-based statistical translation

Строим **порождающую вероятностную модель**

$$p(\mathbf{e}, a|\mathbf{f}) = \frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})$$

перевод

исходный текст

пословная вероятностная модель

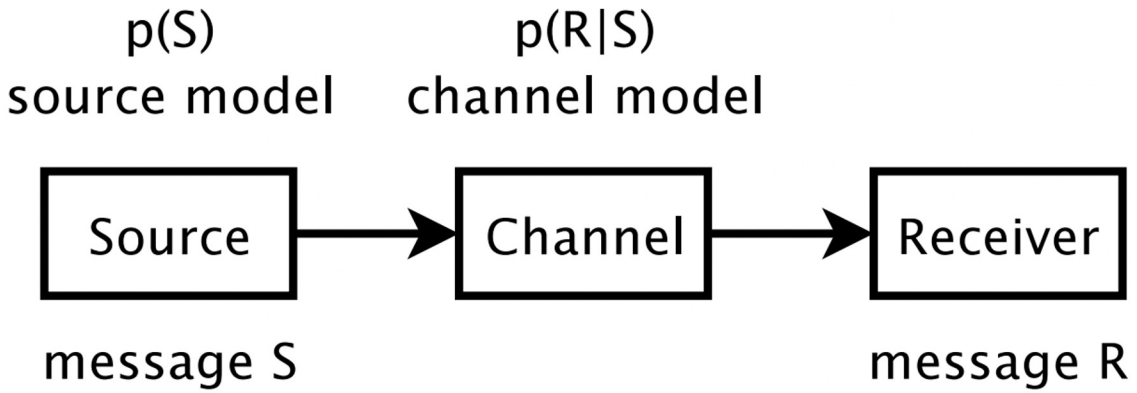
модель выравнивания

Word-based SMT: Noisy channel

По формуле Байеса

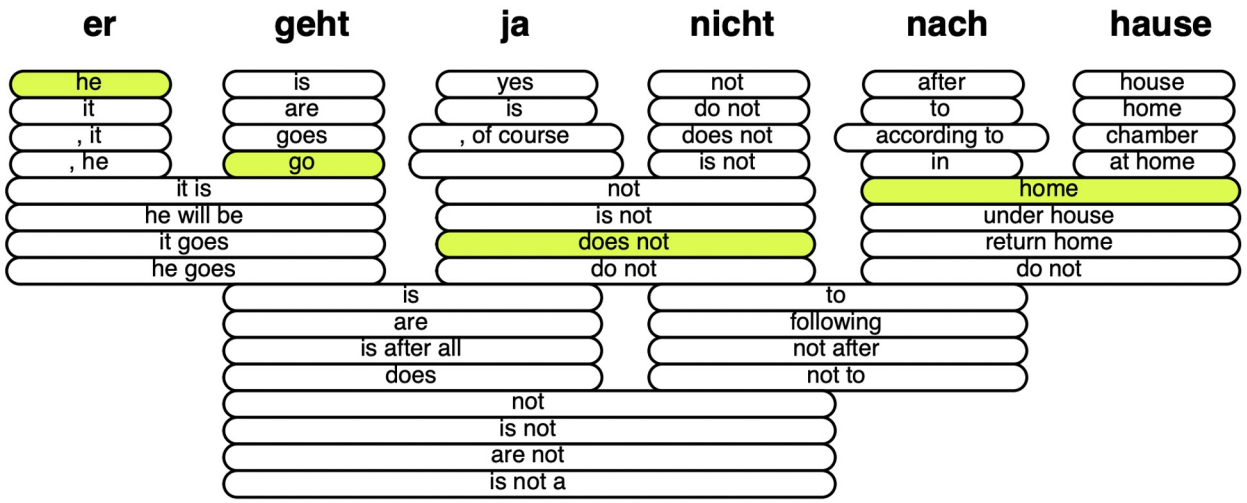
$$\text{argmax}_{\mathbf{e}}\, p(\mathbf{e}|\mathbf{f}) = \text{argmax}_{\mathbf{e}} \frac{p(\mathbf{f}|\mathbf{e})\, p(\mathbf{e})}{p(\mathbf{f})}$$

$$= \text{argmax}_{\mathbf{e}}\, p(\mathbf{f}|\mathbf{e})\, p(\mathbf{e})$$



Модель “зашумленного канала”

PBMT (Phrase-Based Machine Translation)



Sequence to sequence models

- Текст делится на токены
- Авторегрессионное предсказание следующего токена

$$P(y|x) = P(y_2|y_1, x)P(y_3|y_1, y_2, x)\ldots\underbrace{P(y_T|y_1, y_2, \ldots, x)}$$

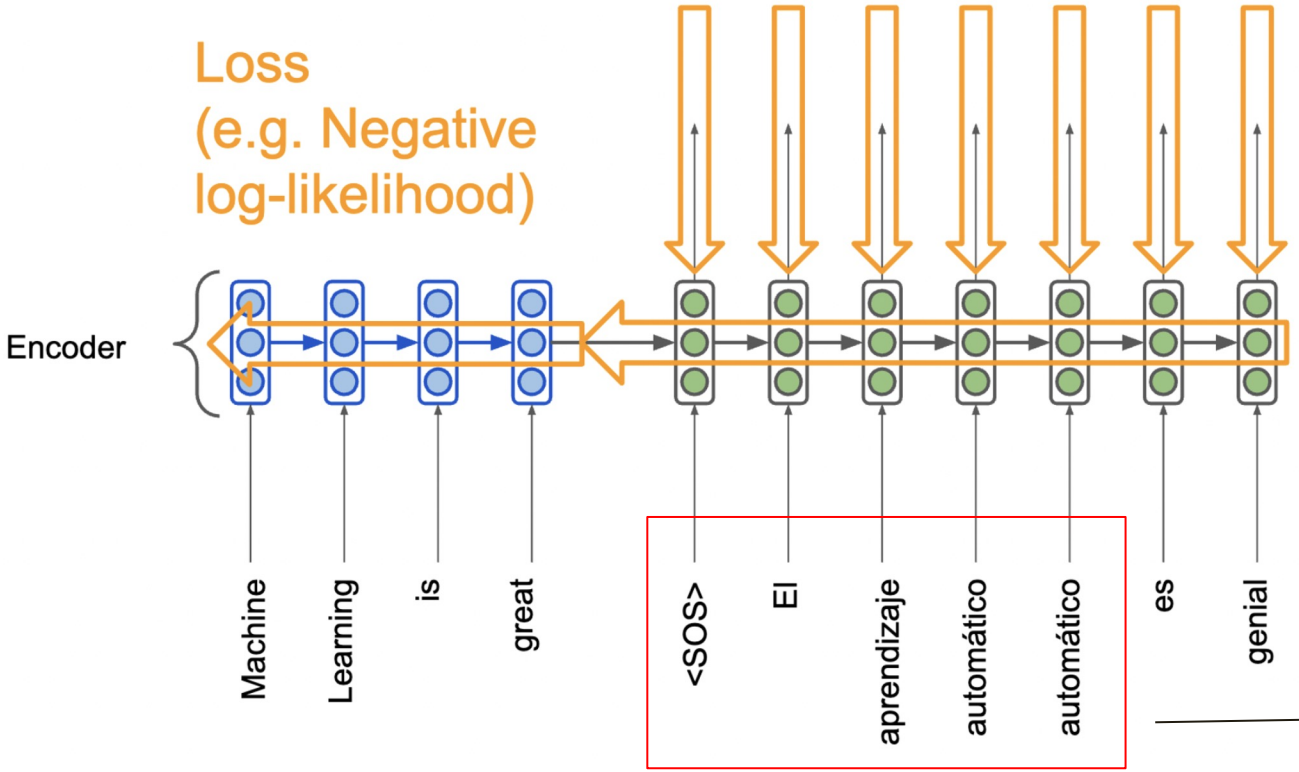
- Модель - это одна дифференцируемая функция
- Обучение end-to-end

Encoder-decoder

Sutskever et.al. (2014)

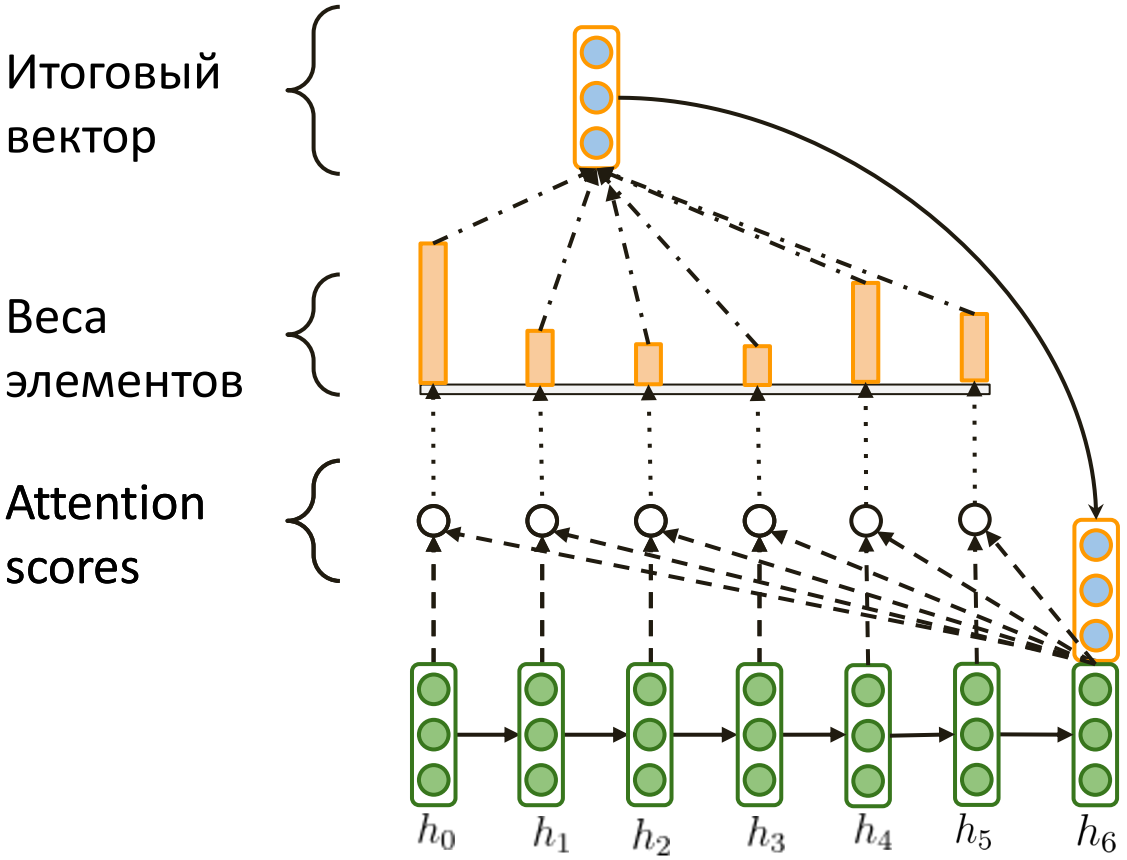
$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \qquad CE = -\sum_i^C t_i log(f(s)_i)$$

Функция потерь - категориальная кроссэнтропия потокенная



“teacher forcing” - при обучении используем префикс эталонного перевода

Attention [Bahdanau et.al. 2014]



$$z = \sum_{i=1}^{T} \alpha_i h_i$$

$$\sum_{i=1}^{T} \alpha_i = 1$$

$$\forall i \quad \alpha_i \in [0; 1]$$

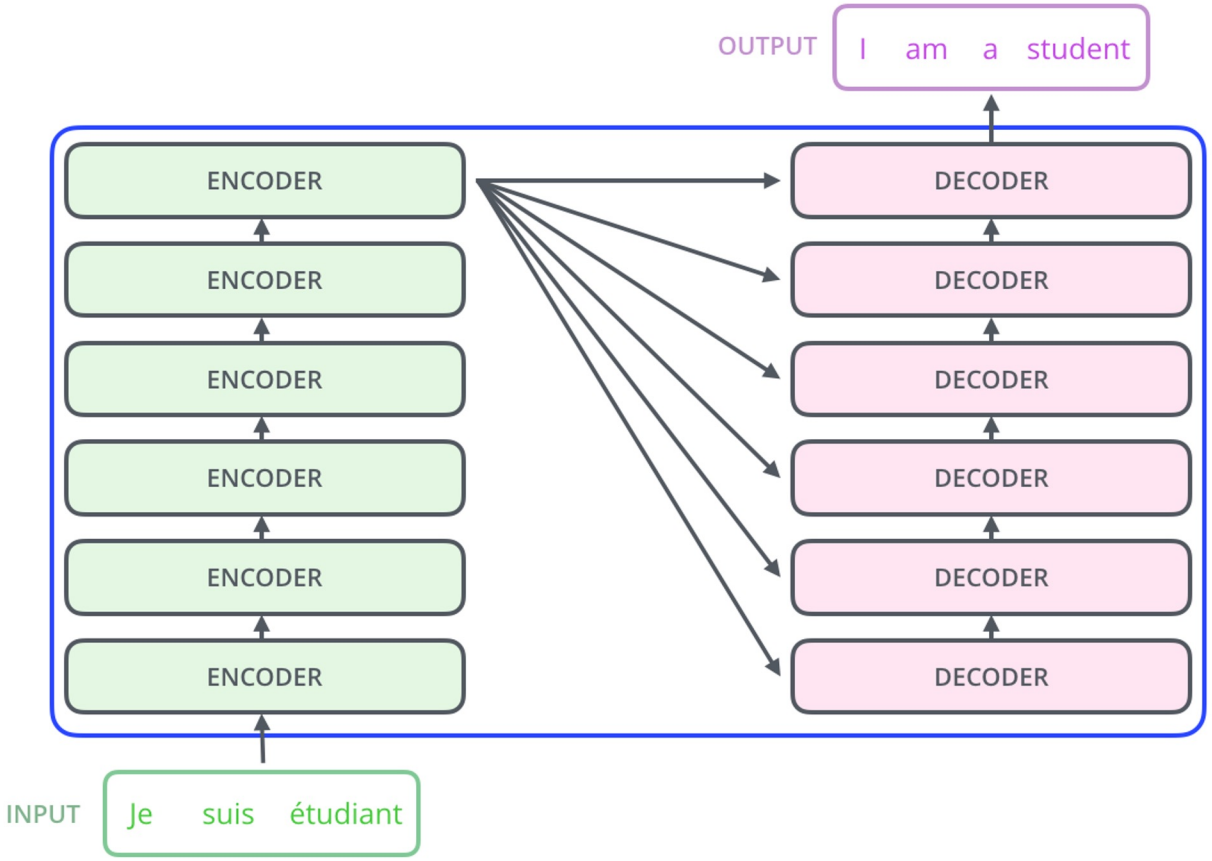
Transformer [Vasvani et.al. 2017]

Ключевая идея:

- для учета контекстных зависимостей не нужна рекуррентность
- достаточно модели попарных связей между токенами через механизм внимания

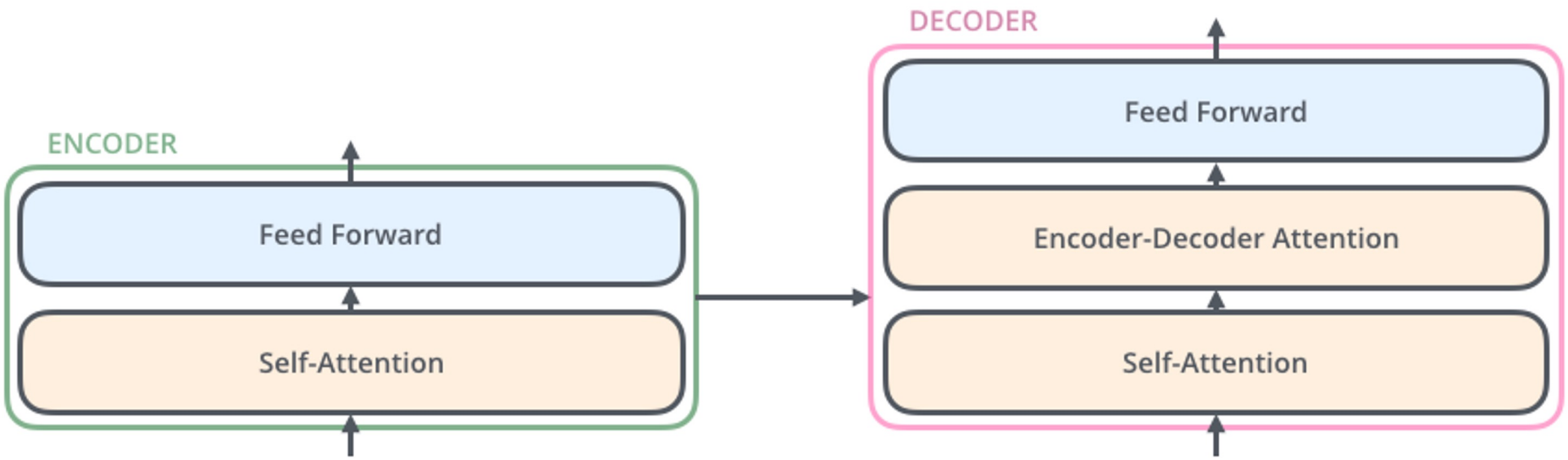
Источник изображения: http://jalammar.github.io/illustrated-transformer/

Transformer [Vasvani et.al. 2017]



Источник изображения: http://jalammar.github.io/illustrated-transformer/

Transformer [Vasvani et.al. 2017]



Источник изображения: http://jalammar.github.io/illustrated-transformer/

Обучение

Обучающая выборка - “параллельные” корпуса

Откуда брать обучающие данные?

1. Ручные переводы аннотаторами
2. Автоматический сбор данных
    a. Переводы книг / статей + выравнивание
    b. Поиск похожих по смыслу предложений и абзацев среди неструктурированных данных

Оценка качества перевода

Потокенные overlap-based метрики

- Разбиваем текст на токены
- Смотрим на долю совпадений между токенами перевода и токенами референса

Precision

$$\frac{correct}{output\text{-}length}$$

Recall

$$\frac{correct}{reference\text{-}length}$$

F-мера

$$\frac{precision \times recall}{(precision + recall)/2}$$

Расстояние Левенштейна (нормированное)

$$\text{WER} = \frac{substitutions + insertions + deletions}{reference\text{-}length}$$

BLEU

- смотрим на n-граммную точность
- “какая доля n-грамм перевода присутствует в референсе?”

$$\text{BLEU} = \min\left(1, \frac{\textit{output-length}}{\textit{reference-length}}\right) \left(\prod_{i=1}^{4} \textit{precision}_i\right)^{\frac{1}{4}}$$

Нейросетевые автометрики

- BLEURT
- COMET

Трансформерные encoder-only модели,
предсказывают меру близости к референсу

- COMET-QE

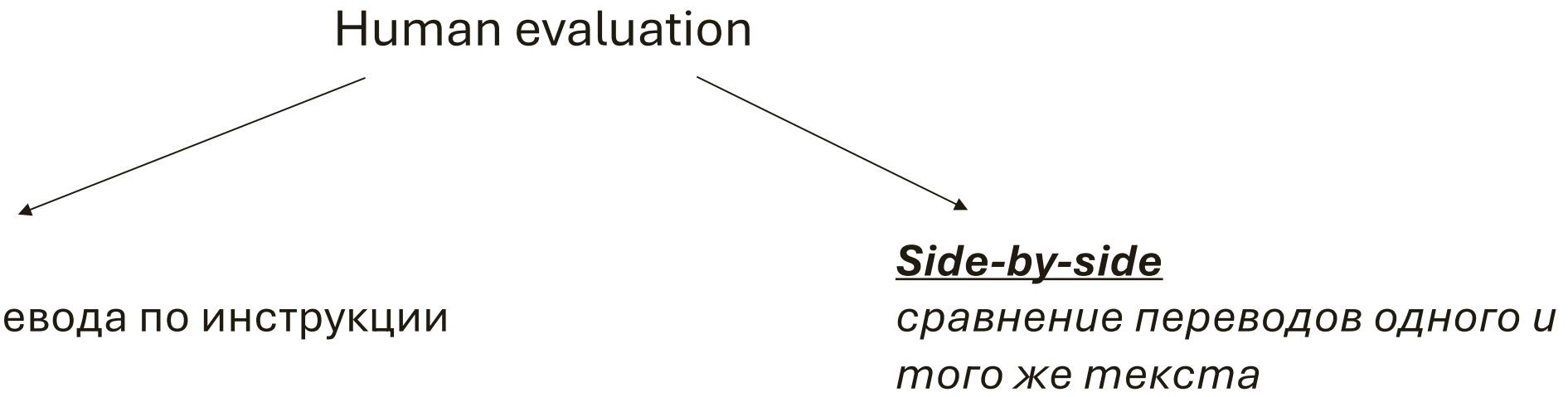
Аналогично COMET, но в режиме QE
Т.е. вычисляет “качество в абсолюте”, а не близость

- GEMBA

Дообучение GPT под выделение ошибок

Human evaluation

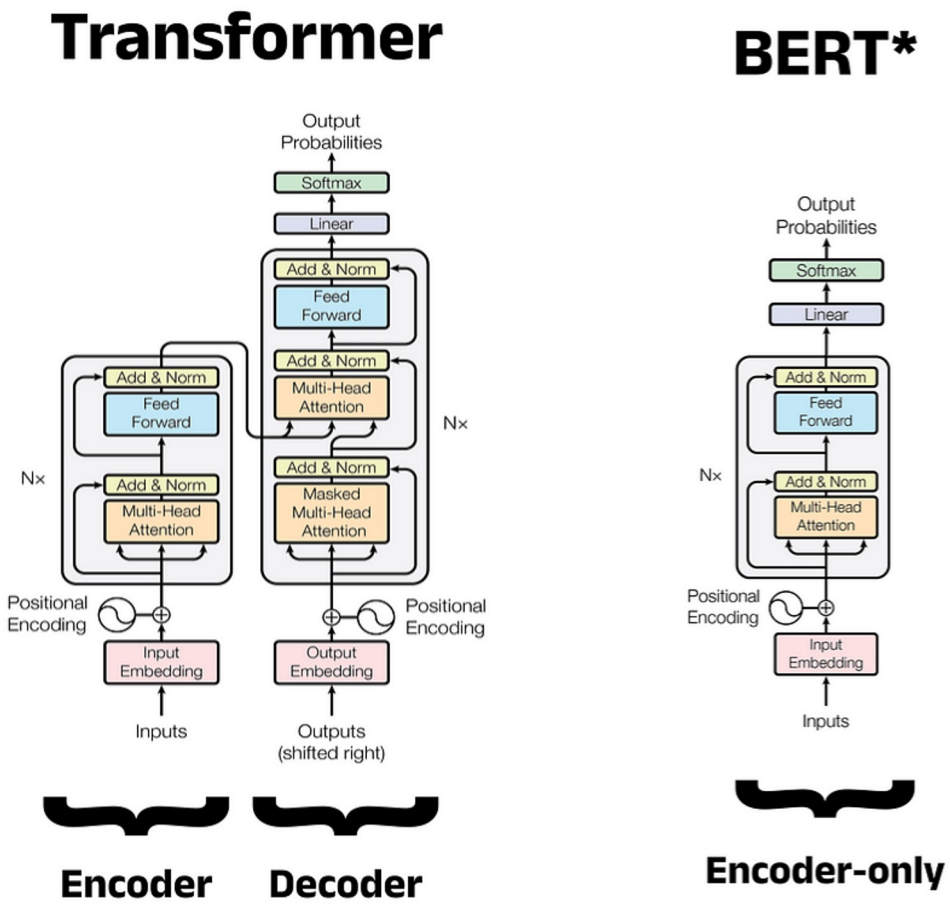
- Более точная оценка - ручная аннотация человеком



BERT – Bidirectional Encoder Representations from Transformers

BERT

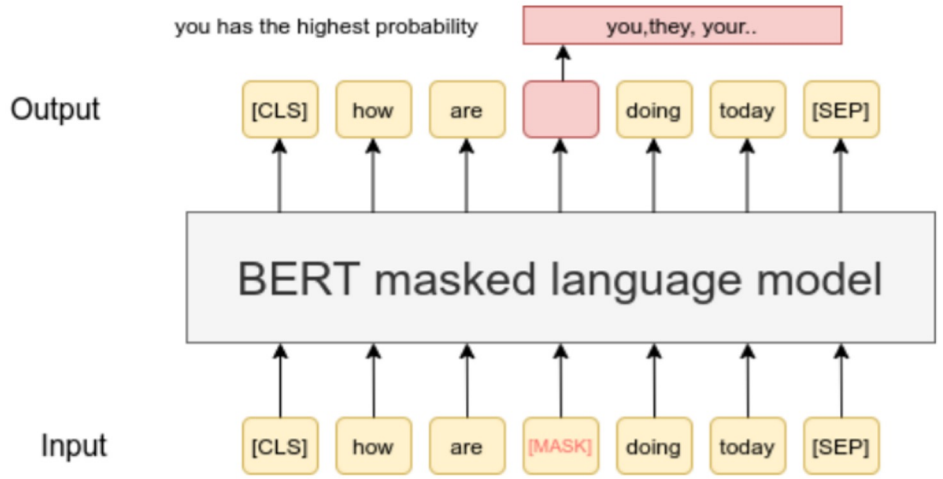
- Encoder-only
- Bidirectional Encoder Representations from Transformers (MLM task)
- Based on Transformer architecture
- Bidirectional context encoding
- Pretrained on Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) tasks

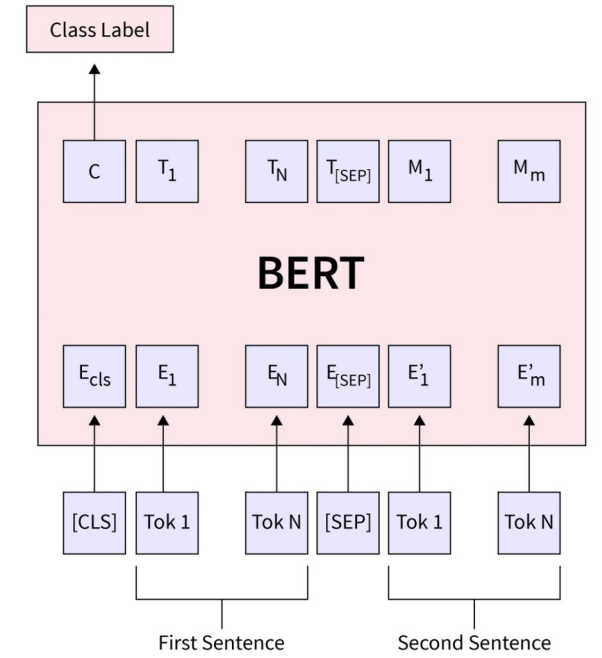


2018

BERT. Pretrain Tasks.

- Masked Language Model (MLM)
  - 15% of tokens are randomly masked
  - 80% replaced with [MASK]
  - 10% replaced with a random word
  - 10% remain unchanged
- Next Sentence Prediction (NSP)
  - Model learns to predict if sentence B logically follows sentence A





source1, source2

BERT. Architecture and Training.

Architecture:

- 12 layers (BERT-base) or 24 layers (BERT-large)
- 768 hidden neurons (base) or 1024 (large)
- 12 attention heads (base) or 16 (large)

Training Details:

- Batch size: 256 sequences
- 1,000,000 training steps
- Optimizer: Adam with
  - $\beta 1 = 0.9$
  - $\beta 2 = 0.999$
- Learning rate: 1e-4
- Dropout: 0.1 across all layers
- Activation: GELU

BERT. Corpora and Data.

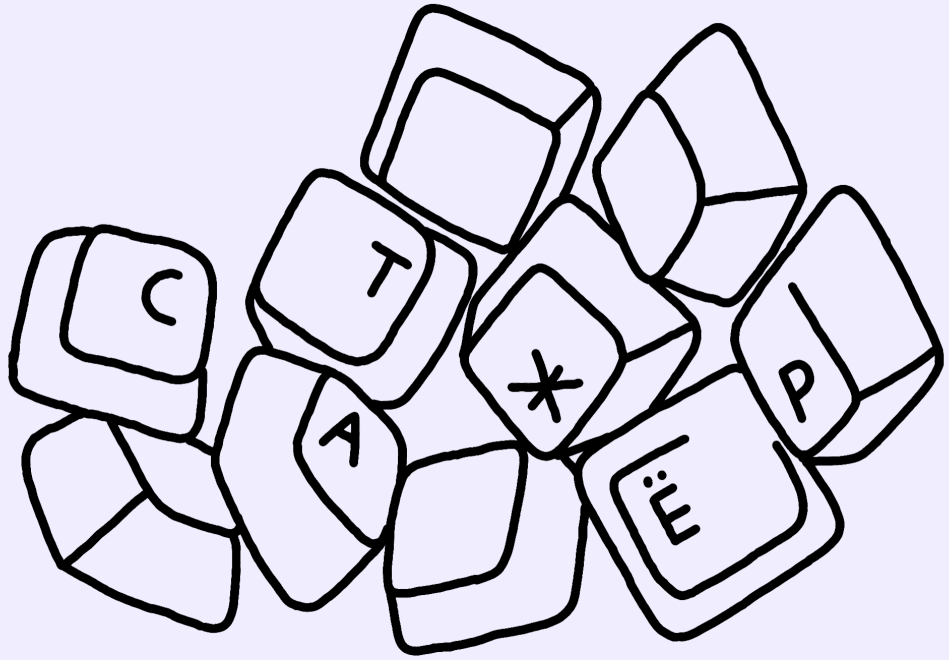
Input Data:

- WordPiece tokenization with a 30,000-token vocabulary
- Maximum sequence length: 512 tokens
- Special tokens: [CLS] at the start, [SEP] between sentences and at the end

Pretraining Corpora:

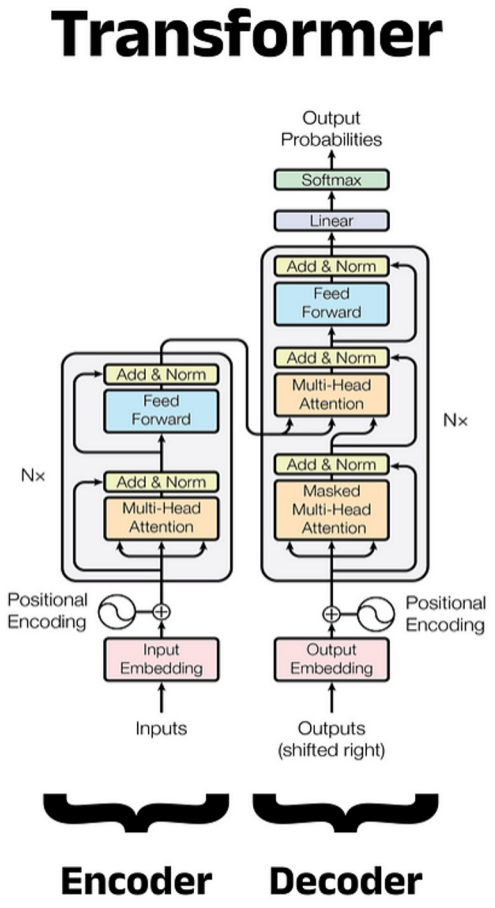
- BookCorpus (800M words)
- English Wikipedia (2.5B words)

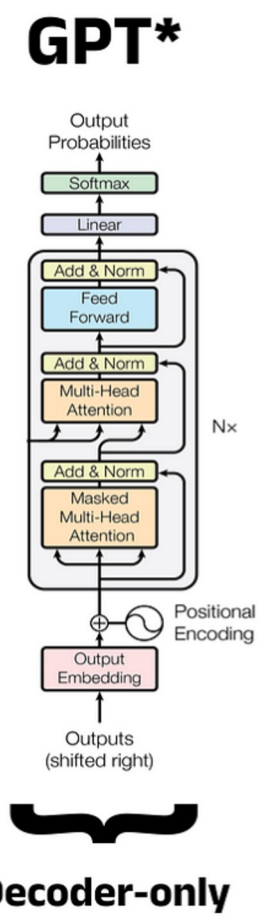
GPT – Generative Pre-trained Transformer



GPT

- Generative Pre-trained Transformer
- Based on decoder-only Transformer architecture
- Autoregressive modeling (next-token prediction)
- Trained on massive unlabelled text datasets





GPT-1. Architecture and Training

Architecture:

- Decoder-only transformer
- 12 layers
- 768-dimensional embeddings
- 12 attention heads
- Total parameters: 117M

Training Details:

- Batch size: 64 sequences
- Optimizer: Adam
- Max learning rate: 2.5e-4
- Linear warmup for the first 2000 updates
- Cosine decay down to 0
- Total updates: 100 epochs on 1B tokens (~800k updates)
- Dropout: 0.1 on attention outputs and feed-forward layers
- L2 regularization: 0.01 for non-embedded weights

GPT-1. Corpora and Data.

Input Data:

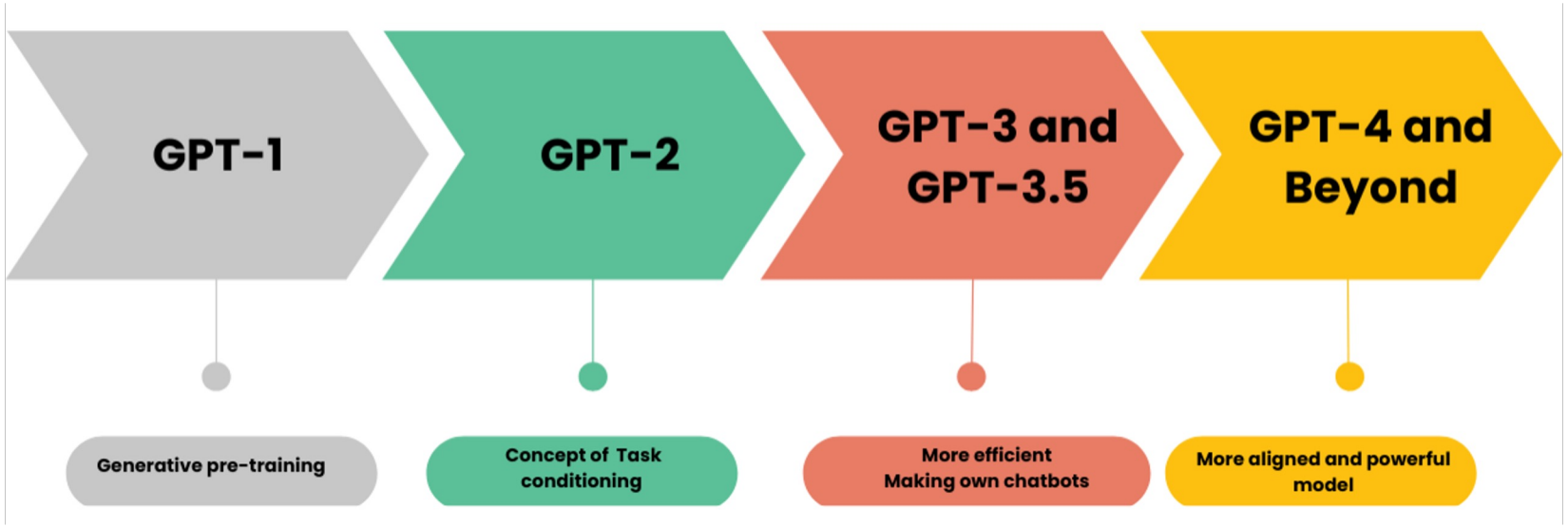
- BytePair Encoding (BPE) with a 40,000-token vocabulary
- Special tokens: [START], [END], [EXTRACT]
- Sequence length: 512 tokens

Pretraining Corpora:

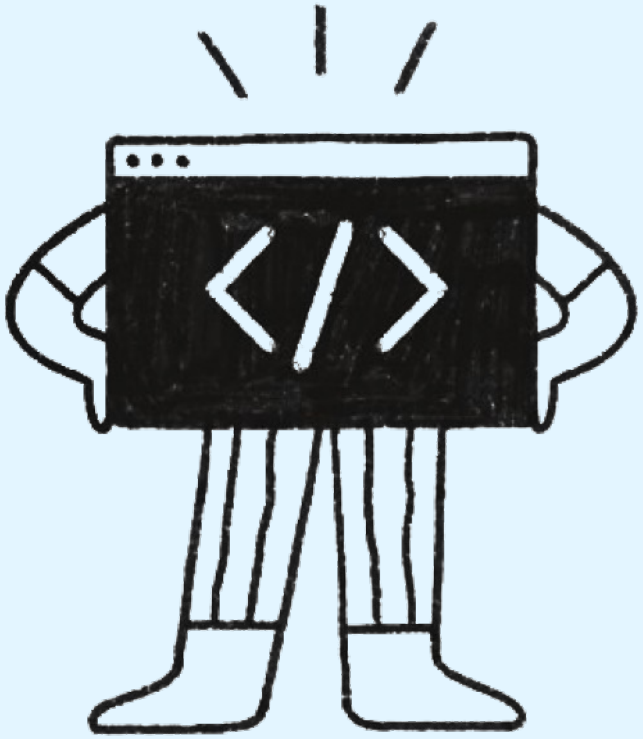
- BookCorpus (over 7,000 unpublished books)

GPT. Model Versions

- **GPT-1 [2018]:** 117M parameters
- **GPT-2 [2019]:** 1.5B parameters, improved text generation quality
- **GPT-3 [2020]:** 175B parameters, few-shot learning
- **GPT-3.5 [2022]:** GPT-3 + Instruct tuning + RLHF
- **GPT-4 [2023]:** Multimodal, improved context understanding

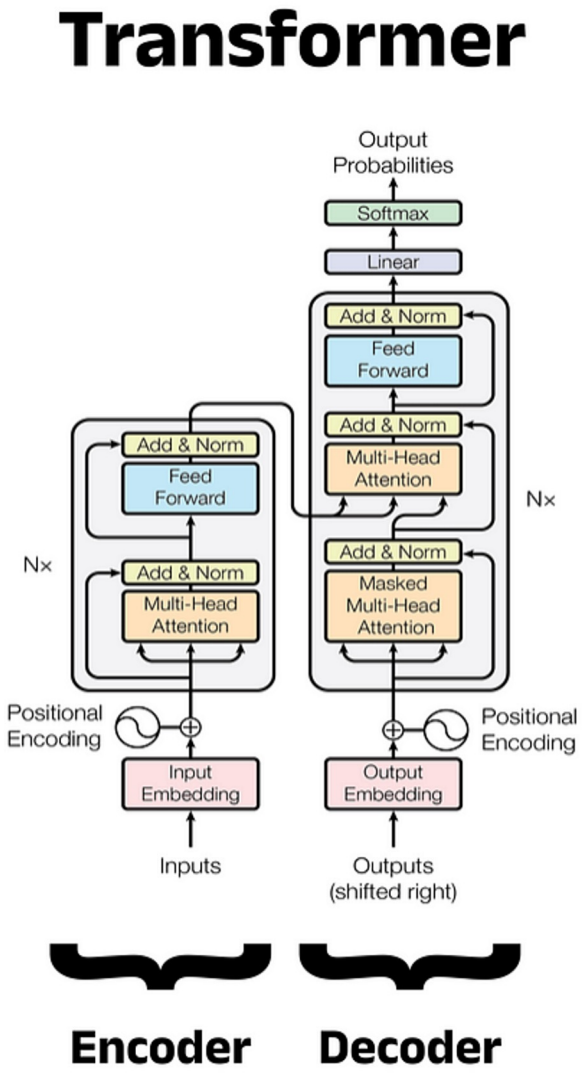


T5 – Text-to-Text Transfer Transformer



T5

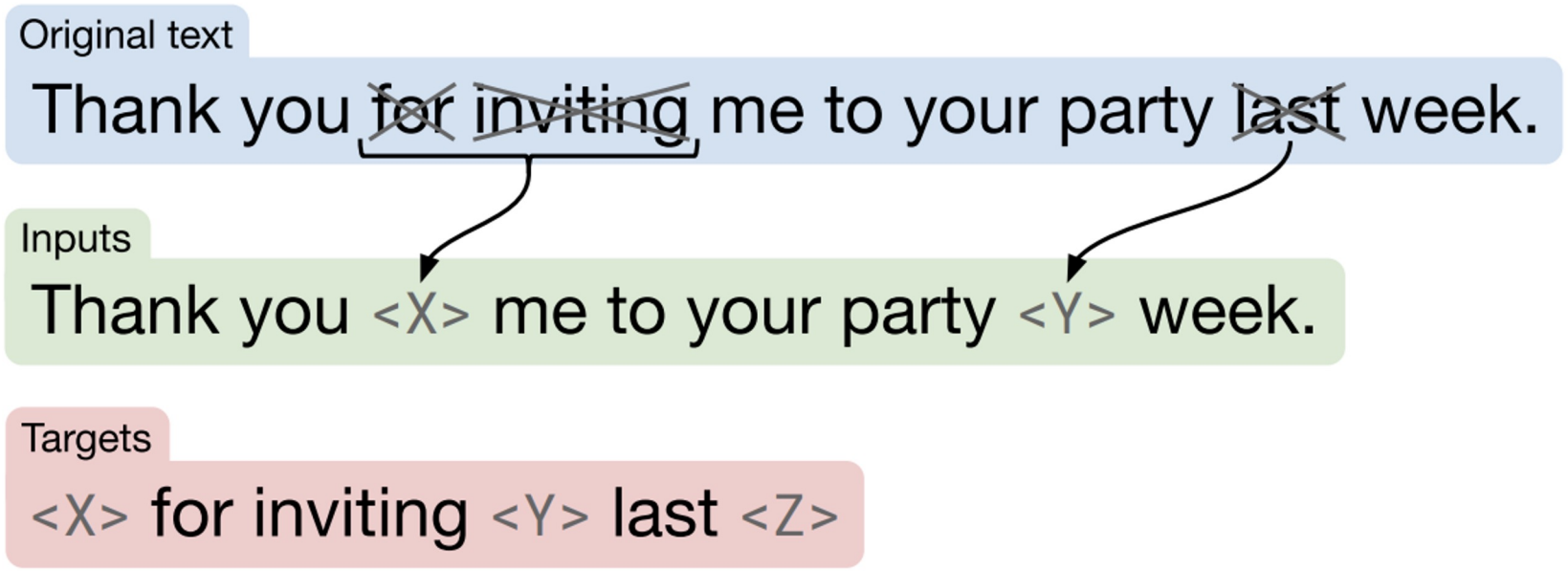
- Text-to-Text Transfer Transformer
- Unified approach: all tasks represented as text-to-text transformation
- Uses encoder-decoder transformer architecture



2019

T5. Pretrain Tasks.

- **Masked Language Modeling** with “span corruption”
- All tasks presented in a text-to-text format



source

T5. Architecture and Training

Architecture:

- Encoder-decoder transformer
- Various model sizes (Small, Base, Large, 3B, 11B)
- T5-Base:
  - 12 encoder and 12 decoder layers
  - 768-dimensional embeddings
  - 12 attention heads
- Total parameters: ~220M

Training Details:

- Batch size: 128 sequences
- Optimizer: AdaFactor
- Constant learning rate: 0.01
- Trained on 1 trillion tokens
- Dropout: 0.1
- Using prefixes to denote tasks (for example, 'translate English to German:')
- Using relative positional encoding – a method that employs a power law and sinusoidal functions for effective encoding of relative positions between tokens in a sequence, allowing for better scalability on long texts and improving the model's generalization capability.

T5. Corpora and Data.

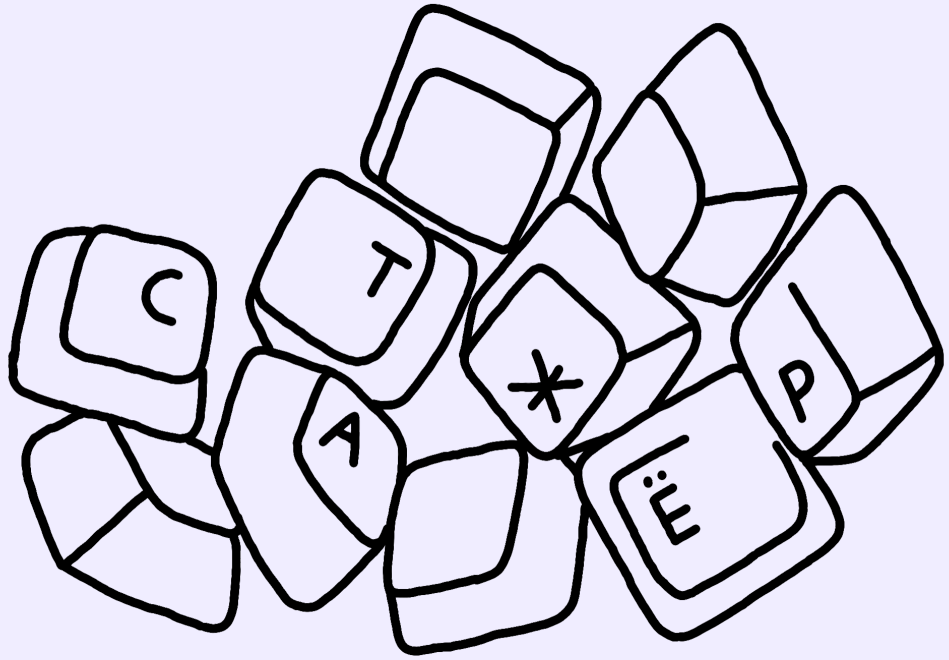
Input Data:

- SentencePiece tokenizer with a 32,000-token vocabulary
- Sequence length: 512 tokens by default, but can be increased
- All tasks are presented as text-to-text transformations
- Use of special tokens to denote the beginning and end of a sequence

Pretraining Corpora:

- C4 (Colossal Clean Crawled Corpus)
- Wikipedia
- WebText

Практика



Спасибо за внимание