**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Н. Э. БАУМАНА**

## Образовательный центр VK

# Дисциплина: Нейронные сети в ML

# Тема проекта: Машинный перевод

Выполнили студенты:
Мейерханс Алексей
Сухова Екатерина
Фадеев Роман

Москва - 2024

# Введение

Обработка естественного языка (NLP) — это область искусственного интеллекта, которая фокусируется на взаимодействии между компьютерами и людьми через естественный язык.

Машинный перевод (МП) —это применение NLP, которое занимается автоматическим переводом текстов с одного языка на другой с использованием алгоритмов и технологий.

Состояние проблемы машинного перевода остается актуальным и многогранным. Несмотря на значительные успехи в области технологий, такие как использование глубокого обучения и нейронных сетей, остаются вызовы, связанные с качеством перевода. Проблемы, такие как многозначность слов, идиоматические выражения и культурные особенности, все еще могут приводить к ошибкам в автоматическом переводе.

# Основные подходы и технологии в МП

1. Правила и системы на основе правил: набор заранее определенных правил для обработки грамматики и лексики языков.
2. Статистический машинный перевод (СМП): основывается на вероятностных моделях, которые анализируют большие корпуса параллельных текстов для выявления вероятностей перевода слов и фраз.
3. Нейронный машинный перевод (НМП): учитывает контекст всего предложения, а не отдельных слов.
4. Трансформеры: используют механизм внимания для обработки входных данных, что позволяет лучше захватывать зависимости между словами.
5. Адаптация к домену: Специфические области могут требовать адаптации моделей перевода для улучшения качества в узкоспециализированных текстах, таких как юридическая или медицинская документация.

# Анализ обучающего набора данных

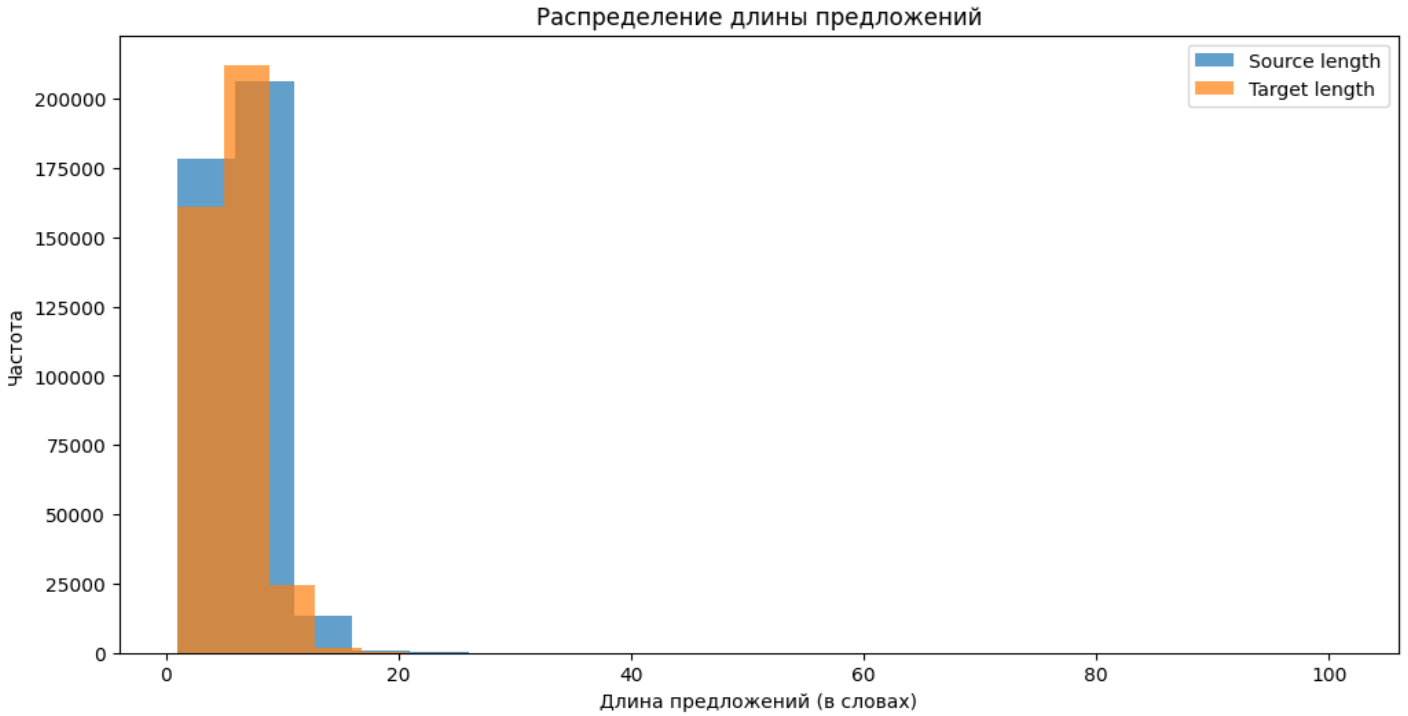
Датасет (399919 записей)

| Index | Source | Target |
|---|---|---|
| 0 | go. | марш! |
| 1 | go. | иди. |
| 2 | go. | идите. |
| 3 | hi. | здравствуйте. |
| 4 | hi. | привет! |

Количество уникальных слов (source): 31077
Количество уникальных слов (target): 87892

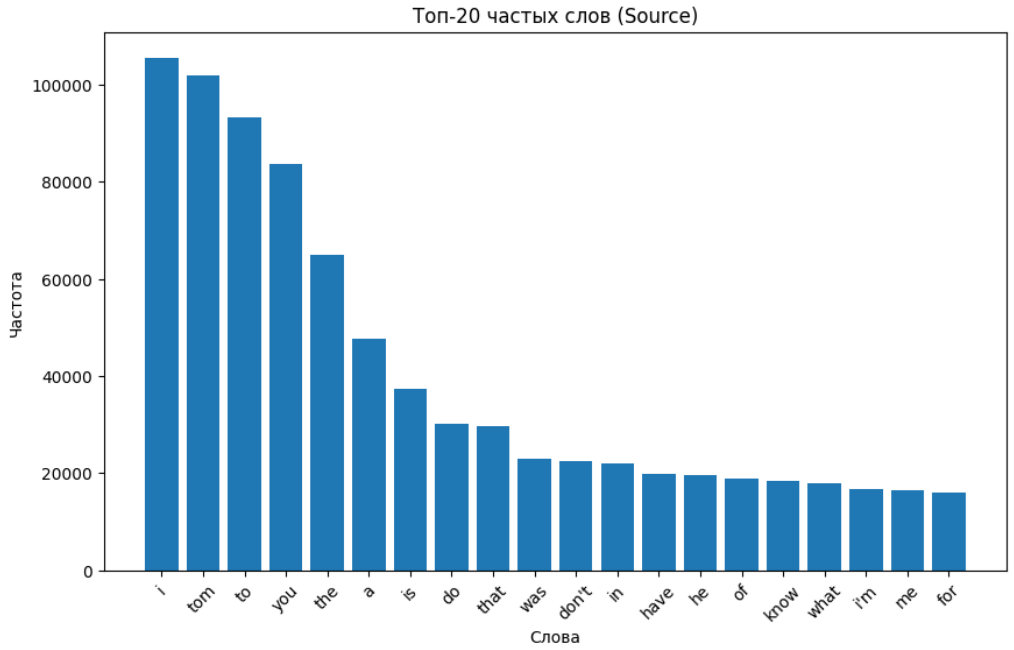
Количество общих слов: 313

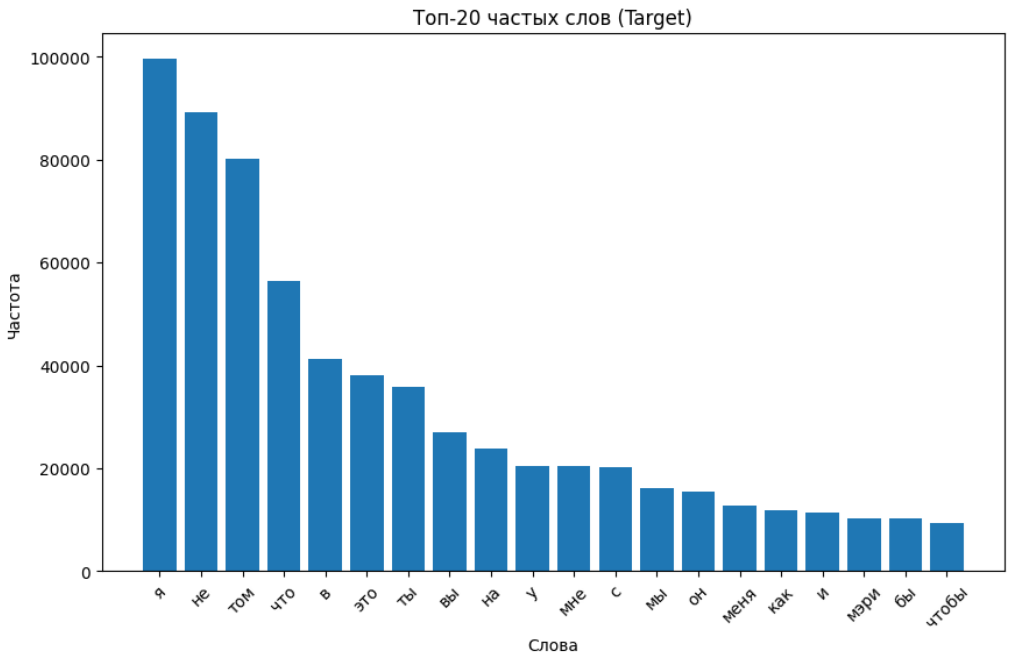
Средняя тональность (source): 0.034
Средняя тональность (target): 3.75e-07

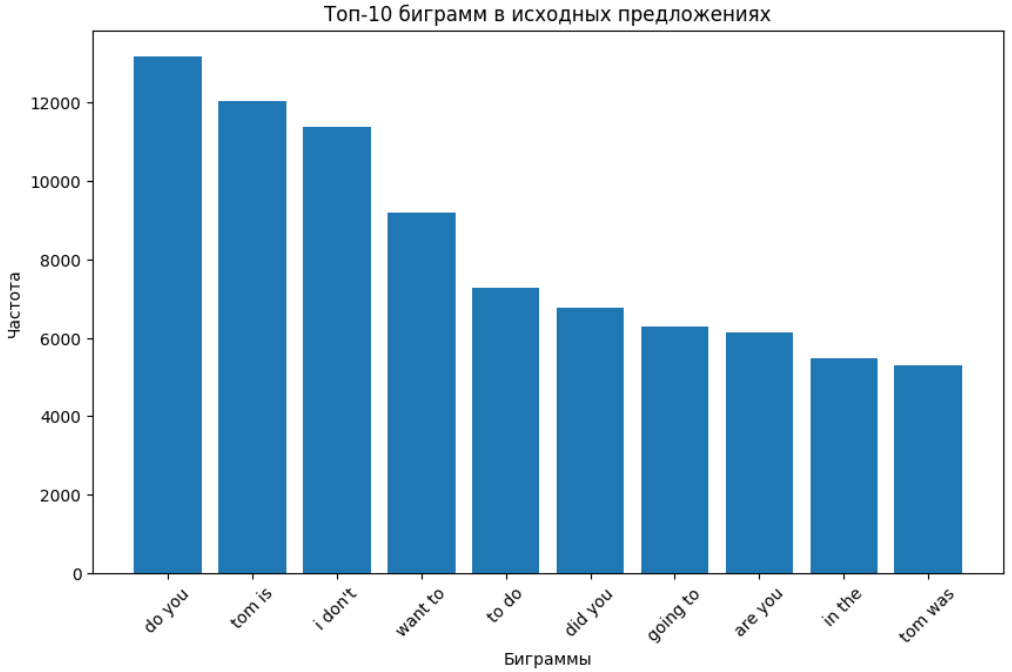


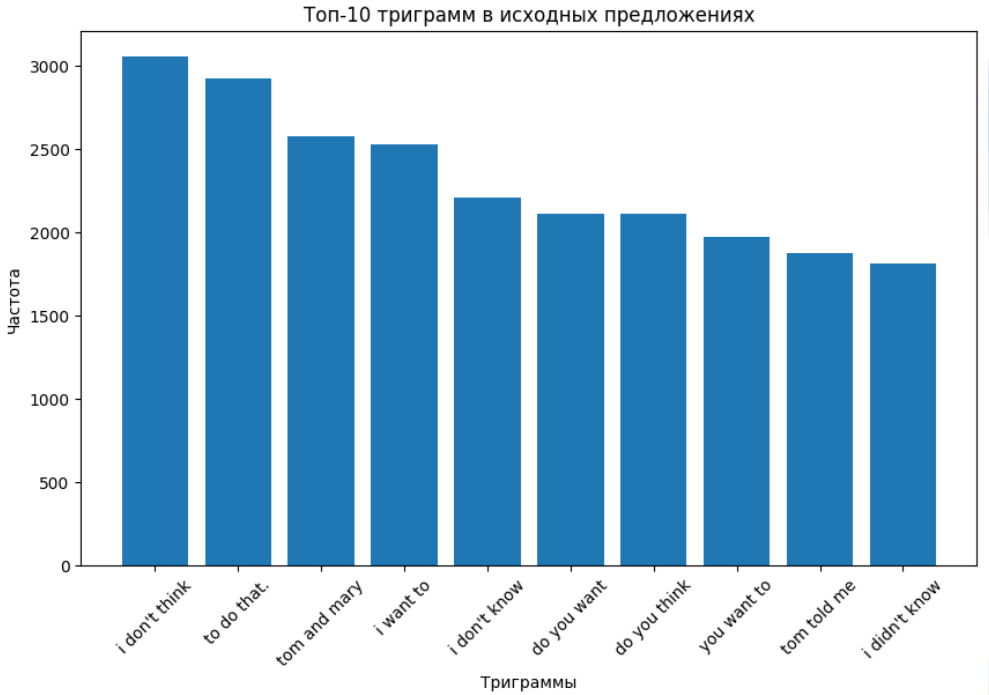

Статистика

| Статистика | Исходная длина | Таргетная длина |
|---|---|---|
| count | 399919.000000 | 399919.00000 |
| mean | 6.057924 | 5.24902 |
| std | 2.236346 | 2.07320 |
| min | 1.000000 | 1.00000 |
| 25% | 5.000000 | 4.00000 |
| 50% | 6.000000 | 5.00000 |
| 75% | 7.000000 | 6.00000 |
| max | 101.000000 | 80.00000 |


Топ-20 частых слов (Source)
Частота
Слова
0
20000
40000
60000
80000
100000
i
tom
to
you
the
a
is
do
that
was
don't
in
have
he
of
know
what
i'm
me
for


Топ-20 частых слов (Target)
Частота
Слова
0
20000
40000
60000
80000
100000
я
не
том
что
в
это
ты
вы
на
у
мне
с
мы
он
меня
как
и
мэри
бы
чтобы


Топ-10 биграмм в исходных предложениях
Частота
Биграммы
0
2000
4000
6000
8000
10000
12000
do you
tom is
i don't
want to
to do
did you
going to
are you
in the
tom was


Топ-10 триграмм в исходных предложениях
Частота
Триграммы
0
500
1000
1500
2000
2500
3000
i don't think
to do that.
tom and mary
i want to
i don't know
do you want
do you think
you want to
tom told me
i didn't know

# Архитектура LSTM

Основная проблема стандартных RNN заключается в том, что они могут забывать важную информацию из-за эффекта затухающего градиента, когда градиенты становятся слишком малыми для эффективного обучения.

То есть стандартная RNN сеть не может видеть далеко назад – при прохождение градиента обратно – происходит экспоненциальное затухание градиента, чем дальше назад – тем больше затухает.

Решение – **LSTM**.

LSTM (Long Short-Term Memory) — это тип рекуррентной нейронной сети (RNN), разработанный для преодоления проблем, связанных с обучением на длинных последовательностях.

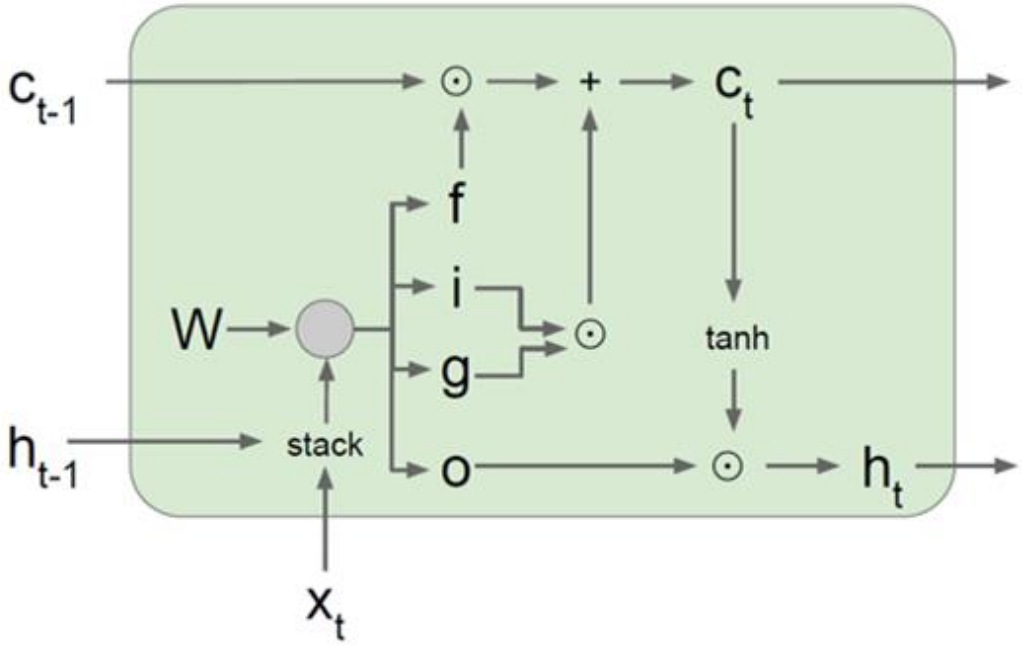
LSTM разработаны специально, чтобы избежать проблемы долговременной зависимости. Запоминание информации на долгие периоды времени – это их обычное поведение, а не что-то, чему они с трудом пытаются обучиться.

LSTM используются в различных областях, включая: • Машинный перевод • Генерация текста • Распознавание речи • Анализ тональности • Прогнозирование временных рядов

# Архитектура LSTM

$$
\begin{aligned}
\boldsymbol{c}'_t &= \tanh\left(W_{xc}\boldsymbol{x}_t + W_{hc}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{c'}\right) && \textit{candidate cell state} \\
\boldsymbol{i}_t &= \sigma\left(W_{xi}\boldsymbol{x}_t + W_{hi}\boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right) && \textit{input gate} \\
\boldsymbol{f}_t &= \sigma\left(W_{xf}\boldsymbol{x}_t + W_{hf}\boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right) && \textit{forget gate} \\
\boldsymbol{o}_t &= \sigma\left(W_{xo}\boldsymbol{x}_t + W_{ho}\boldsymbol{h}_{t-1} + \boldsymbol{b}_o\right) && \textit{output gate} \\
\boldsymbol{c}_t &= \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \boldsymbol{c}'_t, && \textit{cell state} \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t) && \textit{block output}
\end{aligned}
$$

Преимущества LSTM
- Долгосрочные зависимости
- Гибкость
- Устойчивость к затухающим градиентам




$$
\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}
$$

$$
c_t = f \odot c_{t-1} + i \odot g
$$

$$
h_t = o \odot \tanh(c_t)
$$

**f**: Forget gate, Забывать ли состояние элемента

**i**: Input gate, Писать ли состояние элемента

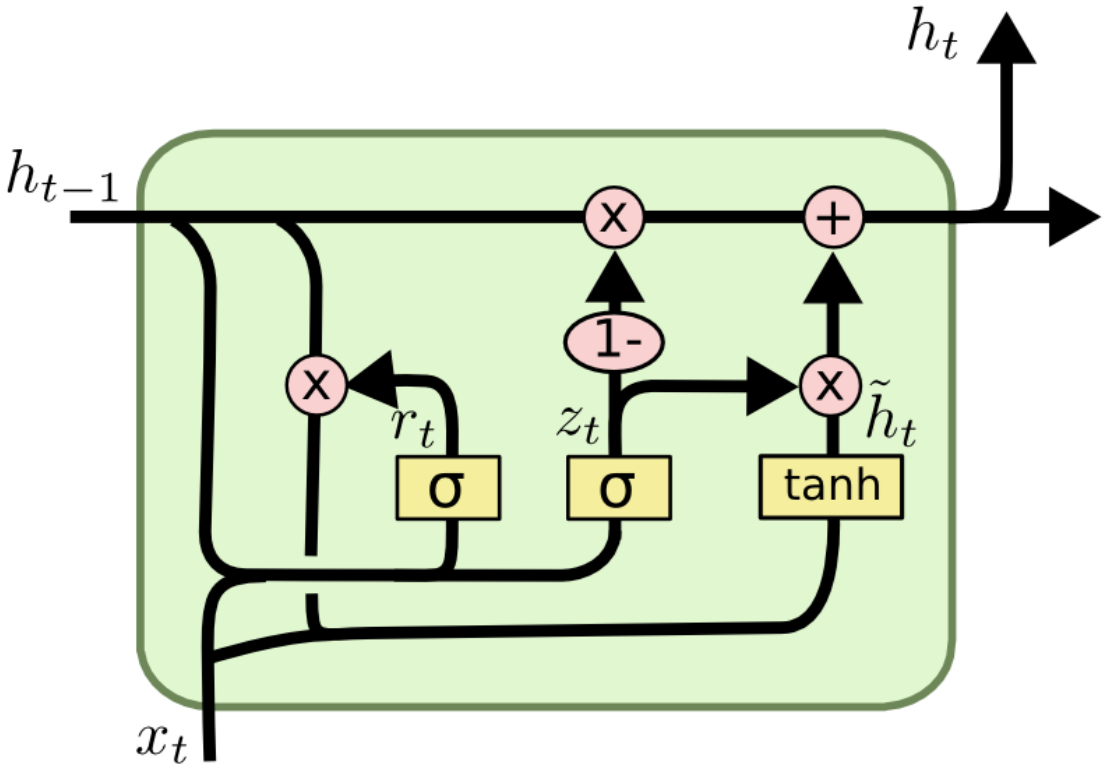
**g**: Gate gate, Сколько писать в элемент

**o**: Output gate, Сколько выводить из элемента

# Архитектура GRU

**GRU** – gated recurrent unit

- Совмещаем выходные и забывающий гейты – их заменяет update_gate (u)
- Совмещаем hidden_state(h) и cell_state(c) – объединяем их
- Добавляем reset_gate




$$\boldsymbol{u}_t = \sigma(W_{xu}\boldsymbol{x}_t + W_{hu}\boldsymbol{h}_{t-1} + \boldsymbol{b}_u),$$
$$\boldsymbol{r}_t = \sigma(W_{xr}\boldsymbol{x}_t + W_{hr}\boldsymbol{h}_{t-1} + \boldsymbol{b}_r),$$
$$\boldsymbol{h}'_t = \tanh(W_{xh'}\boldsymbol{x}_t + W_{hh'}(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1})),$$
$$\boldsymbol{h}_t = (1 - \boldsymbol{u}_t) \odot \boldsymbol{h}'_t + \boldsymbol{u}_t \odot \boldsymbol{h}_{t-1}.$$

# Архитектура Transformer

Архитектура Transformer была представлена в статье "Attention is All You Need" в 2017 году и произвела революцию в области обработки естественного языка (NLP) и других задачах машинного обучения. Трансформер — это архитектура нейронных сетей, которая стала стандартом для задач обработки естественного языка (NLP), таких как машинный перевод, генерация текста и др.

Основная идея трансформеров в том, что они используют механизм внимания (Attention), который позволяет эффективно обрабатывать последовательности, выделяя важные элементы контекста без необходимости пошаговой обработки, как в RNN.

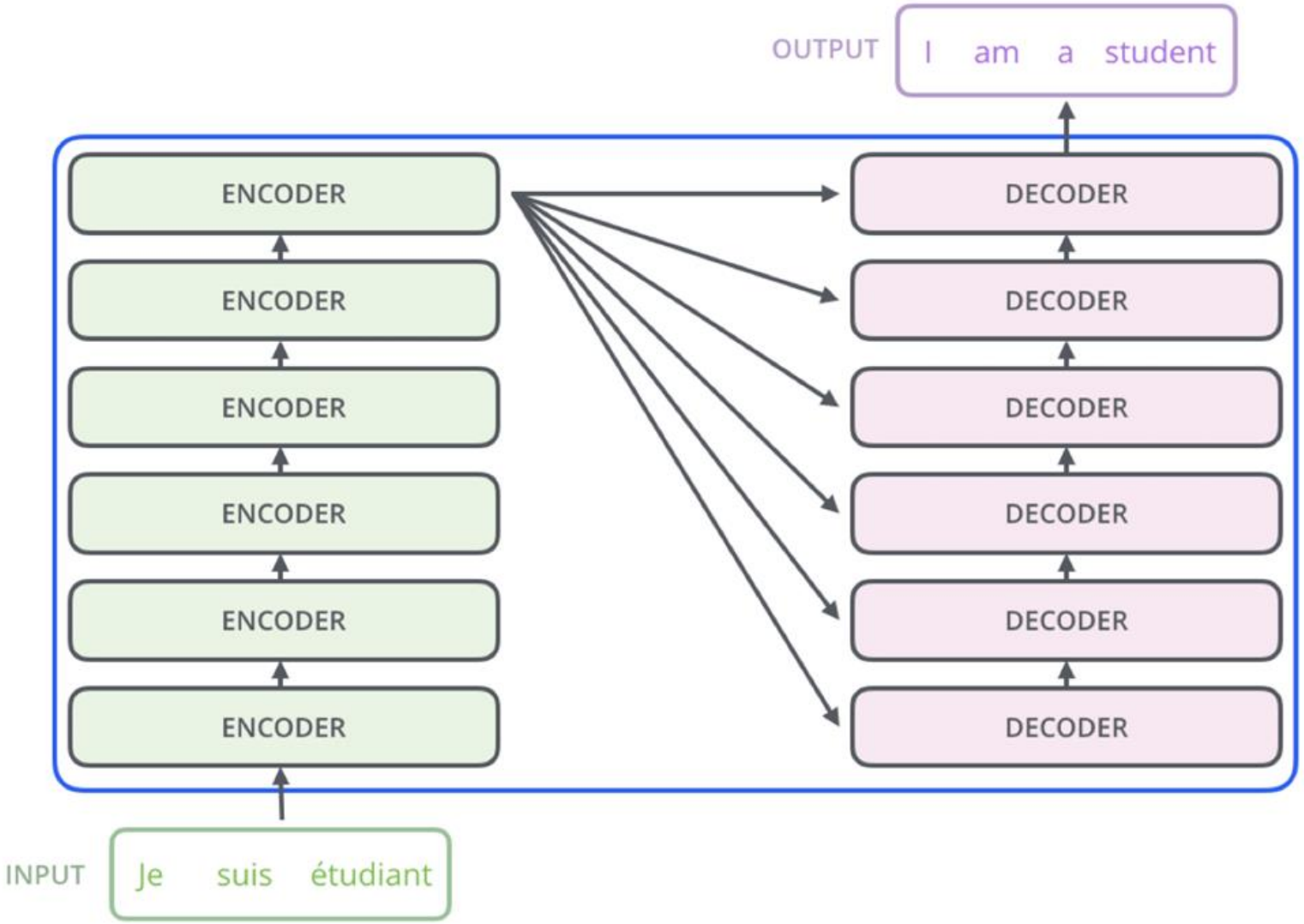
Энкодер преобразует входную последовательность в контекстные представления.

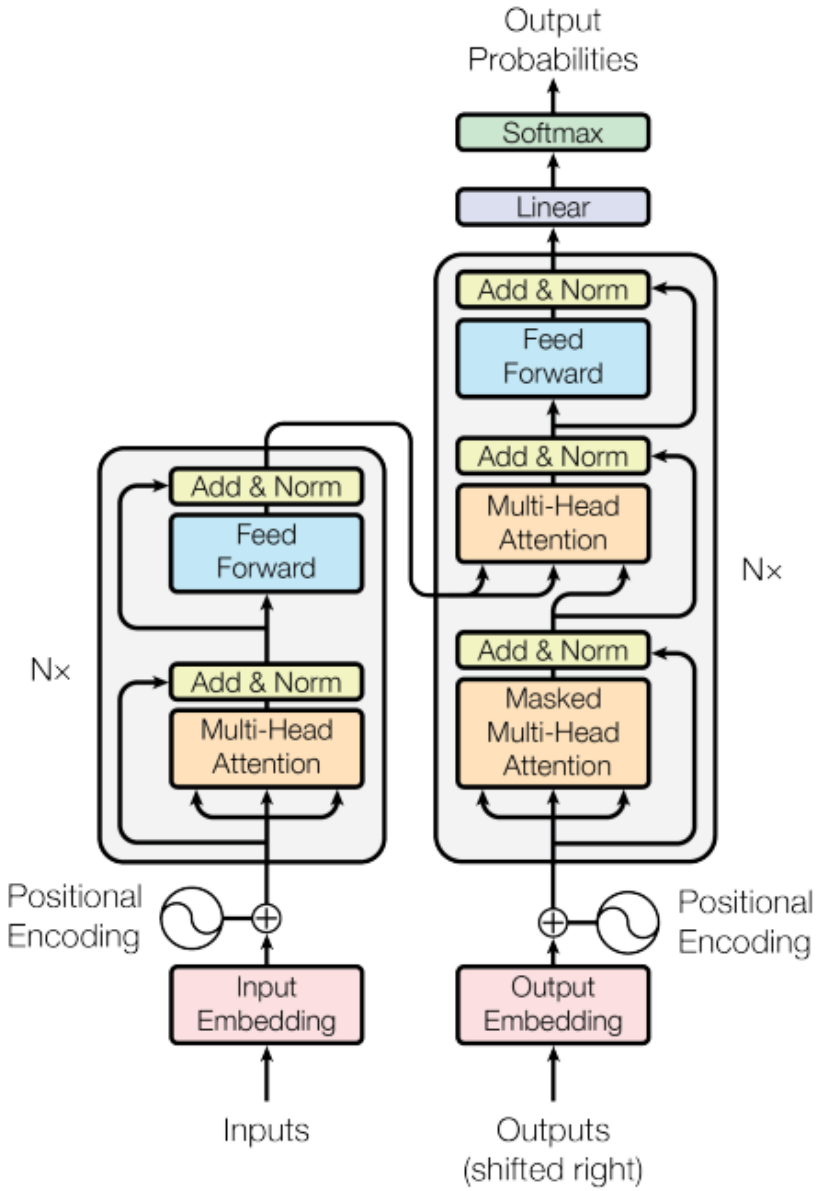
- Self-Attention (учитывает взаимосвязи между токенами входа).
- Feed-Forward Network (нелинейная обработка токенов).
- Нормализация и пропуски связей (Residual Connections).

Декодер генерирует выходную последовательность.

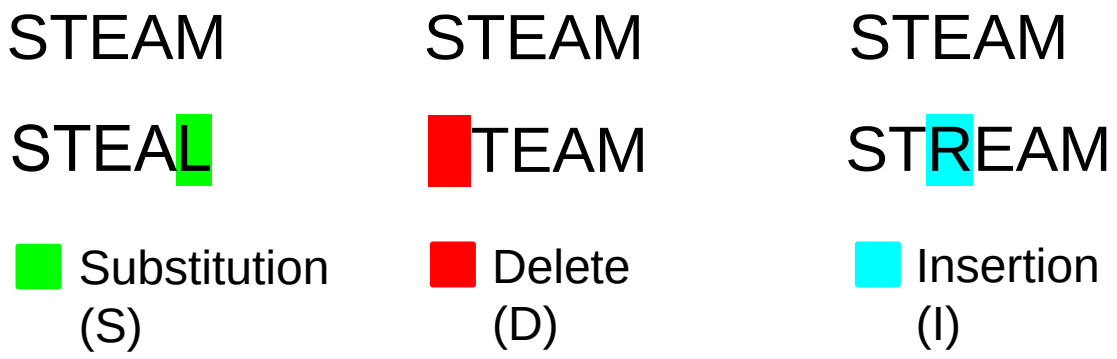
- Masked Self-Attention (учитывает только предыдущие токены).
- Кросс-внимание (взаимодействует с выходами энкодера).
- Аналогичные FFN и Residual Connections.

# Архитектура Transformer



# Выбор метрик




WER

$$WER = \frac{S + D + I}{N}$$

Ошибка на уровне слова

CER

$$CER = \frac{S + D + I}{N}$$

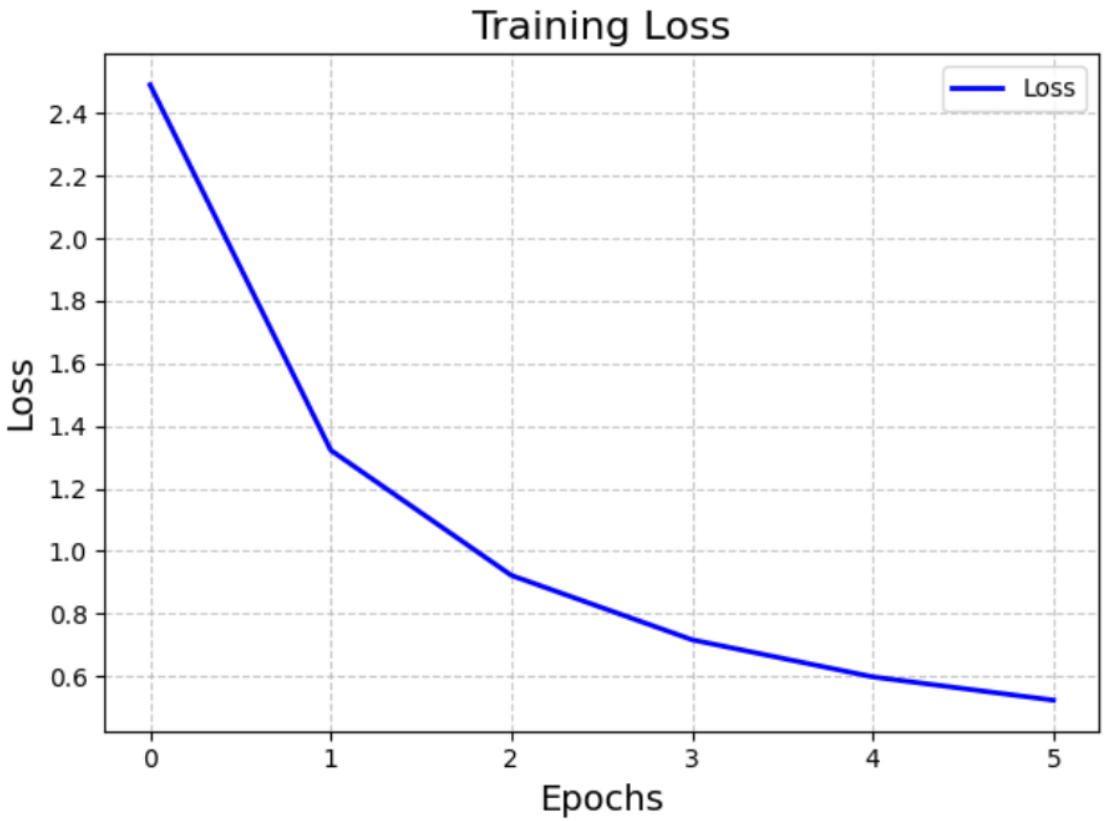
Ошибка на уровне символа

BLEU

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

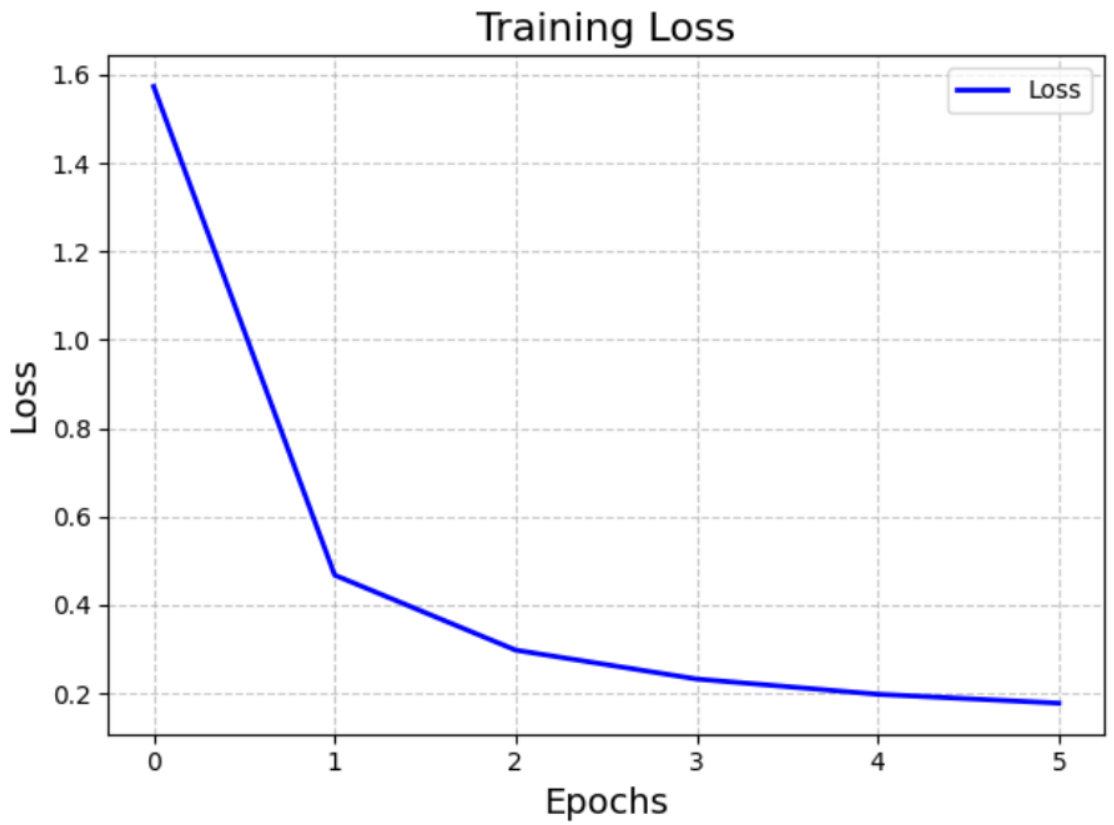
# Обучение LSTM

```
Training Epochs:  13%|          | 4/30 [10:09<53:00, 122.35s/epoch]
Epoch 5/30 - Loss: 2.4910

Training Epochs:  17%|          | 5/30 [25:32<3:08:51, 453.26s/epoch]
Validation at Epoch 5: WER = 0.8088, CER = 0.6607
Training Epochs:  30%|          | 9/30 [34:51<1:03:58, 182.80s/epoch]
Epoch 10/30 - Loss: 1.3228

Training Epochs:  33%|          | 10/30 [48:07<2:15:26, 406.33s/epoch]
Validation at Epoch 10: WER = 0.7582, CER = 0.6022
Training Epochs:  47%|          | 14/30 [57:22<48:04, 180.29s/epoch]
Epoch 15/30 - Loss: 0.9225

Training Epochs:  50%|          | 15/30 [1:10:39<1:39:57, 399.82s/epoch]
Validation at Epoch 15: WER = 0.7351, CER = 0.5743
Training Epochs:  63%|          | 19/30 [1:19:54<32:59, 179.97s/epoch]
Epoch 20/30 - Loss: 0.7167

Training Epochs:  67%|          | 20/30 [1:33:04<1:06:06, 396.65s/epoch]
Validation at Epoch 20: WER = 0.7242, CER = 0.5614
Training Epochs:  80%|          | 24/30 [1:42:18<17:56, 179.35s/epoch]
Epoch 25/30 - Loss: 0.5981

Training Epochs:  83%|          | 25/30 [1:55:29<33:00, 396.10s/epoch]
Validation at Epoch 25: WER = 0.7169, CER = 0.5531
Training Epochs:  97%|          | 29/30 [2:04:43<02:59, 179.48s/epoch]
Epoch 30/30 - Loss: 0.5235
Training Epochs: 100%|          | 30/30 [2:18:01<00:00, 276.04s/epoch]
Validation at Epoch 30: WER = 0.7124, CER = 0.5485
```

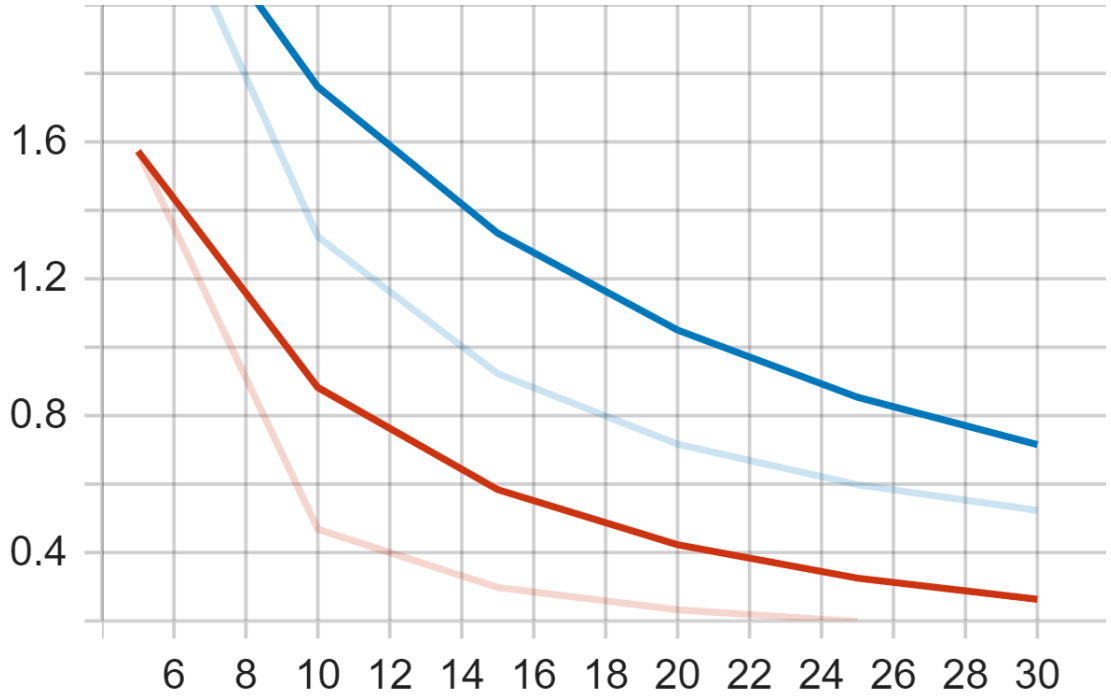


# Обучение GRU

```
Training Epochs:  13%|▉          | 4/30 [16:02<1:23:00, 191.56s/epoch]
Epoch 5/30 - Loss: 1.5730

Training Epochs:  17%|▉          | 5/30 [32:52<3:51:17, 555.10s/epoch]
Validation at Epoch 5: WER = 0.7198, CER = 0.5590
Training Epochs:  30%|▉          | 9/30 [48:39<1:32:51, 265.32s/epoch]
Epoch 10/30 - Loss: 0.4677

Training Epochs:  33%|▉          | 10/30 [1:05:26<3:04:16, 552.83s/epoch]
Validation at Epoch 10: WER = 0.7024, CER = 0.5388
Training Epochs:  47%|▉          | 14/30 [1:21:12<1:13:13, 274.58s/epoch]
Epoch 15/30 - Loss: 0.2979

Training Epochs:  50%|▉          | 15/30 [1:37:57<2:17:56, 551.80s/epoch]
Validation at Epoch 15: WER = 0.6992, CER = 0.5353
Training Epochs:  63%|▉          | 19/30 [1:53:56<51:08, 278.94s/epoch]
Epoch 20/30 - Loss: 0.2326

Training Epochs:  67%|▉          | 20/30 [2:10:47<1:32:40, 556.02s/epoch]
Validation at Epoch 20: WER = 0.6971, CER = 0.5337
Training Epochs:  80%|▉          | 24/30 [2:26:36<27:45, 277.56s/epoch]
Epoch 25/30 - Loss: 0.1982

Training Epochs:  83%|▉          | 25/30 [2:43:47<46:42, 560.54s/epoch]
Validation at Epoch 25: WER = 0.6981, CER = 0.5344
Training Epochs:  97%|▉          | 29/30 [2:59:38<04:39, 279.01s/epoch]
Epoch 30/30 - Loss: 0.1780
Training Epochs: 100%|▉          | 30/30 [3:16:32<00:00, 393.09s/epoch]
Validation at Epoch 30: WER = 0.6976, CER = 0.5331
```



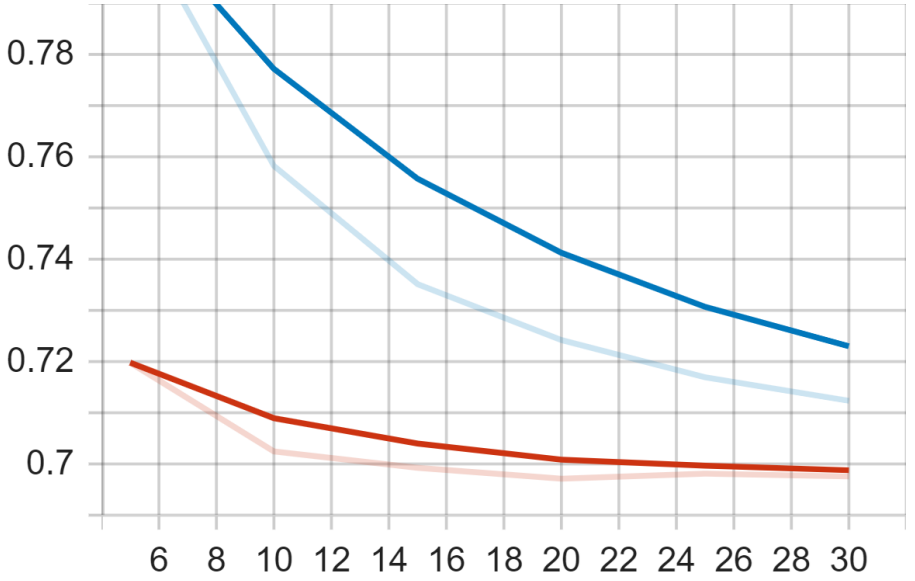
## Train Loss



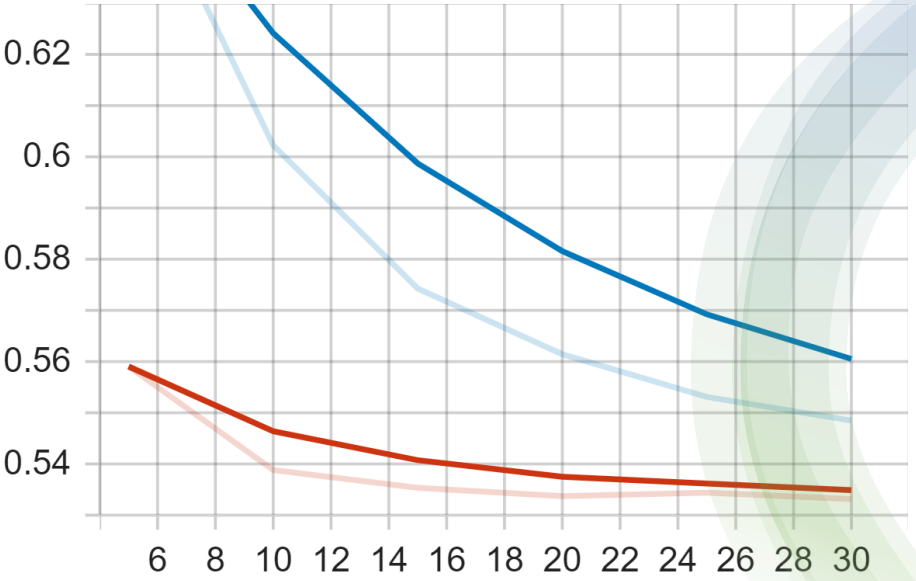

LSTM

GRU

## WER




## CER



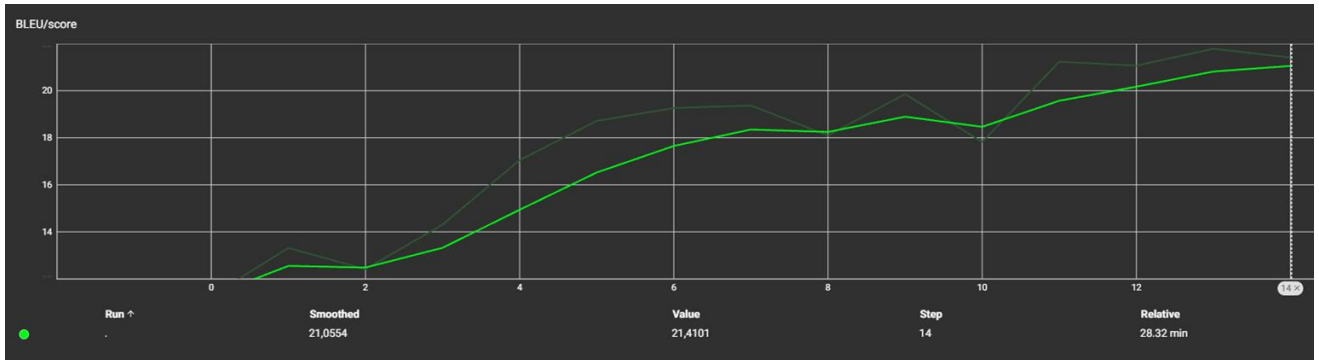
# Обучение Transformer

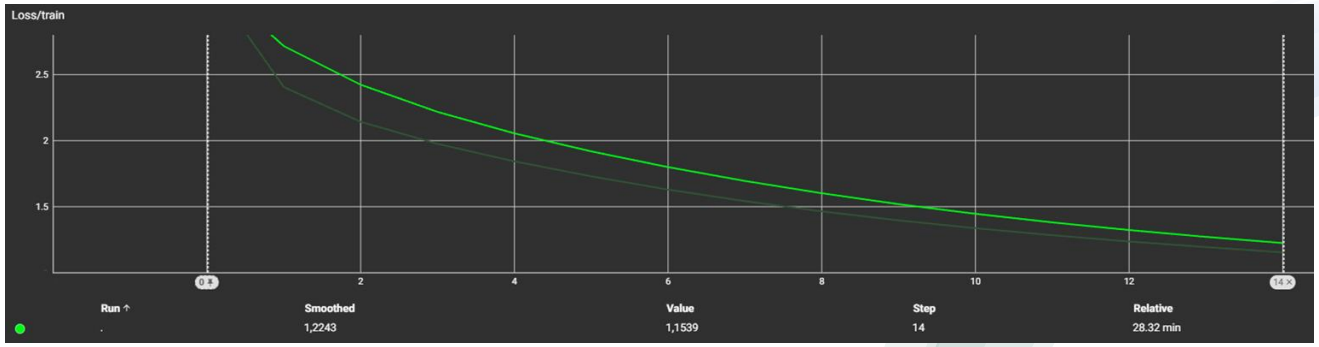
```
EPOCH - 0
  7%|▋          | 1/15 [02:03<28:46, 123.29s/it]
Current BLEU:  11.281875423719455
не думаю что я здесь уже бывал ---> не думаю что я я я
хочешь пойти в кино завтра вечером ---> ты бы бы в в
я весь день работал не покладая рук так что был очень уставшим ---> я все все что я я я
я знаю что некоторые люди ценят мою работу ---> я знаю что это на на
том пожертвовал жизнью чтобы спасти мэри ---> том его его мэри мэри мэри мэри
-----------------------------------
```

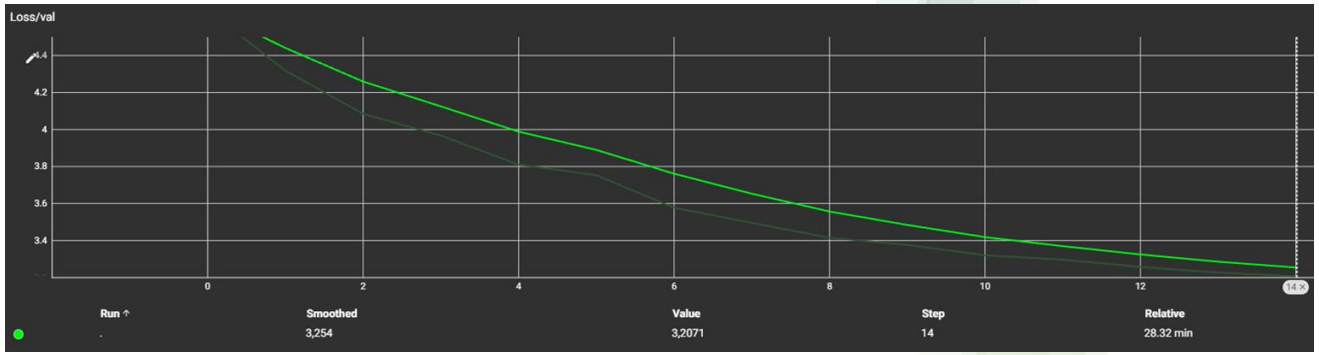
```
EPOCH - 5
 40%|████       | 6/15 [12:08<18:11, 121.30s/it]
Current BLEU:  18.717730186468245
я даже пытался написать песню о том что произошло ---> я даже пытался о о о том
никто не может ла дить с таким человеком ---> никто не может может с не
не понимаю зачем тому это делать ---> не не почему почему тому это делать
я встал раньше обычного чтобы успеть на первый поезд ---> я встал чем чем чем
помни что все мы в одной лодке ---> помню что мы все в в
-----------------------------------
```

```
EPOCH - 10
 73%|███████    | 11/15 [22:14<08:06, 121.51s/it]
Current BLEU:  17.8200199398306
хочешь мы отвезём тебя обратно домой ---> хочешь чтобы чтобы мы тебя тебя
родители должны следить за благо получи ем своих детей ---> родители должно за за за
я бы удивился если бы том этого не сделал ---> я бы удивился если бы том этого делал делал
в мире много плохих людей ---> в много много людей
люди всегда спрашивают меня почему я делаю то что я делаю ---> люди всегда спроси меня почему я я что я я
-----------------------------------
```

```
EPOCH - 14
100%|██████████| 15/15 [30:22<00:00, 121.51s/it]
Current BLEU:  21.41007293142588
том не знал плакать ему или смеяться ---> том не знал что он флиртует
тебе необязательно отвечать на все вопросы ---> ты не должен отвечать на вопросы
ты не знал что том видел как ты это делал ---> ты не знал что том видел как это это это
он осуществил свою меч ту стать художником ---> он понял свои мечта художник
том ведёт себя так будто ничего не случилось ---> том ведёт себя так так ничего не случилось
-----------------------------------
Last 14 epoch train loss:  1.153939488351097
Last 14 epoch val loss:  3.2071023713974727
Last 14 epoch val bleu:  21.41007293142588
```



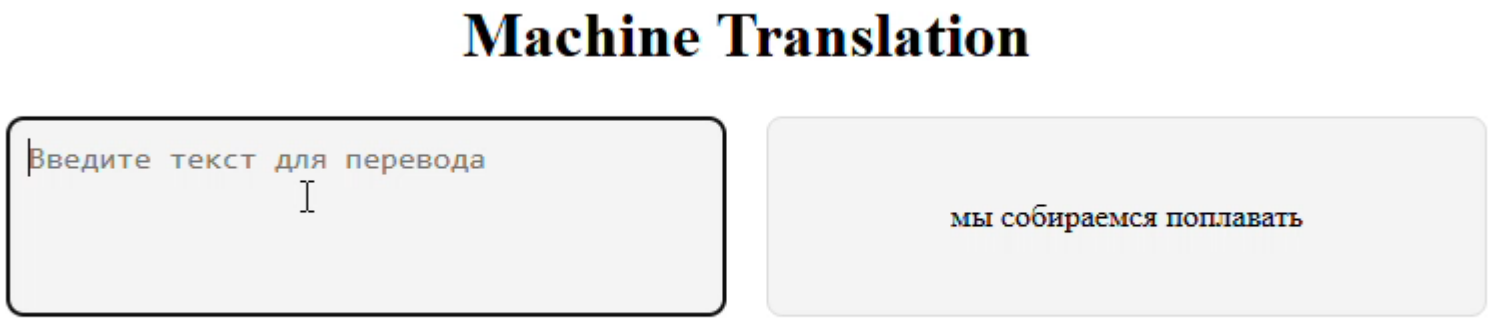
# Пример перевода моделей одного текста

| Input text | LSTM output | GRU output | Transformer output |
|---|---|---|---|
| you work hard | ты слишком много работаешь | ты много работаешь | ты много работаешь |
| tom was sick | том был болен | том болел больнои | том был болен |
| summer is here | лето здесь | летом здесь лето | лето лето |
| tom will win | том выиграет | том выиграет | том выиграет |
| i wonder where she lives | интересно где она живет | интересно где она живет в жизнь | интересно где она живёт |
| you will be there again soon | ты скоро будешь там снова | ты скоро опять там скоро | ты скоро там там будешь |
| i rarely make a mistake | я редко совершаю ошибку | я редко совершаю ошибку в ошибки | я редко делаю ошибку |
| i knew all about that | я знал об этом все | я знала об этом все знаю | я всё об этом знал |
| i plan on going there | я планирую туда поити | я планирую туда поити | я планирую туда туда |
| we are going to swim | мы собираемся поплавать | мы будем плавать | мы будем |

# Сравнение BLEU-метрики моделей

| LSTM BLEU | GRU BLEU | Transformer BLEU |
|---|---|---|
| 17,24 | 24,91 | 21,41 |

# Пример работы



# Использованные материалы:

- Лекции VK
- Книга «Николенко, Кадурин, Архангельская: Глубокое обучение. Погружение в мир нейронных сетей»
- Статьи с хабр:

https://habr.com/ru/companies/wunderfund/articles/331310/

https://habr.com/ru/articles/341240/

https://habr.com/ru/companies/mvideo/articles/780774/

- Еще полезные ссылки:

https://arxiv.org/abs/1706.03762

https://huggingface.co/docs/transformers/index

https://jalammar.github.io/illustrated-transformer/

# Спасибо за внимание!